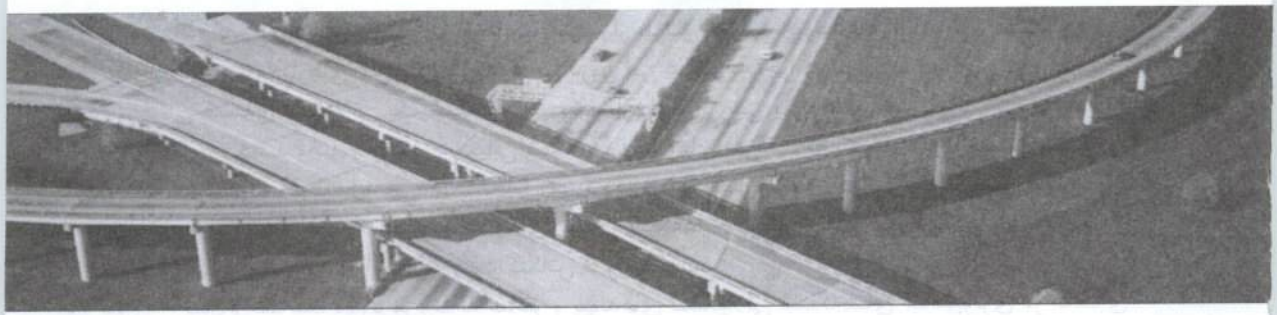


فصل ۱

شبکه‌های کامپیوتری و اینترنت

شبکه‌ی اینترنت امروزی به راستی عظیم‌ترین سیستم مهندسی شده به‌وسیله‌ی انسان است. این سیستم شگفت‌انگیز شامل صدها میلیون رایانه‌ی متصل به یکدیگر، پیوندهای ارتباطی و راهیاب‌های (سوئیچ) ارتباطی می‌گردد. صدها میلیون کاربر از طریق ارتباط‌های هم‌زمان با استفاده از تلفن‌های همراه، تجهیزاتی چون حس‌گرها، دوربین‌های ویژه وب، کنسول‌های بازی، تبلت‌ها، لوح‌های تصویری و حتی ماشین‌های لباسشویی به این شبکه‌ی جهانی متصل شده‌اند. با توجه به این ابعاد از لحاظ گستردگی کاربران و فناوری تجهیزات به‌کار گرفته شده در آن، آیا این امیدواری وجود دارد که بتوان چگونگی عملکرد این شبکه و به طور کلی شبکه‌های کامپیوتری را درک نمود؟ آیا اصول و ساختارهایی را می‌توان تعریف نمود که بنیادهای درک و تحلیل چنین سیستم عظیم و پیچیده‌ای باشند؟ و اگر چنین است، آیا این امکان وجود دارد که این اصول و بنیادها هم جالب و هم سرگرم‌کننده برای یادگیری این شبکه باشند؟ خوشبختانه، پاسخ به همه‌ی این پرسش‌ها آری است! در این اثر تمام تلاش پدیدآورندگان، فراهم‌آوری امکان درک درست از چگونگی عملکرد این عرصه‌ی پویای علمی از شبکه‌های کامپیوتری از طریق ارائه‌ی اصول و زمینه‌های علمی مورد نیاز جهت درک بهینه‌ی نه تنها شبکه‌های کامپیوتری امروز، بلکه فردا می‌باشد.

در این فصل چشم‌انداز کلی از شبکه‌های کامپیوتری و اینترنت ارائه می‌شود. هدف، رسم تصویری گسترده از چارچوبی است که به دنبال می‌آید، زیرا اعتقاد ما بر آن است که برای دیدن جنگل، از درون انبوه درختان آن باید عبور کرد. موارد متعددی از ابتدایی‌ترین تا پیچیده‌ترین موضوعات در ارتباط با شبکه‌های کامپیوتری در این فصل مورد بررسی قرار گرفته‌اند، بدون آنکه تصویر کلی و اصلی که هدف می‌باشد، فراموش گردد. تصویر کلی که در نظر است در این فصل از شبکه ترسیم شود به شرح ذیل سازماندهی می‌شوند.

پس از معرفی برخی واژگان و اصول، تجهیزات بنیادی سخت‌افزاری و نرم‌افزاری لازم برای ایجاد

شبکه‌های کامپیوتری بررسی می‌گردد. این مهم از طریق معرفی پیوندهای ارتباطی، سیستم‌های انتهایی و نرم‌افزارهای کاربردی شبکه که در درون این عناصر به اجرا در می‌آیند، به انجام می‌رسد. پس از آن هسته‌ی تشکیلاتی یک شبکه‌ی کامپیوتری از طریق بررسی پیوندها و راهیاب‌هایی که داده‌ها را انتقال می‌دهند و نیز چگونگی دستیابی شبکه‌ها و رسانه‌های فیزیکی که سیستم‌های انتهایی شبکه را به هسته‌ی شبکه ارتباط می‌دهند، تحلیل می‌گردد. در این فصل موضوع دیگری که بررسی می‌شود، تفهیم این نکته است که اینترنت شبکه‌ای است از شبکه‌ها. چگونگی ارتباط این شبکه‌ها با یکدیگر از موضوعات مهم دیگری است که مورد کنکاش قرار می‌گیرد.

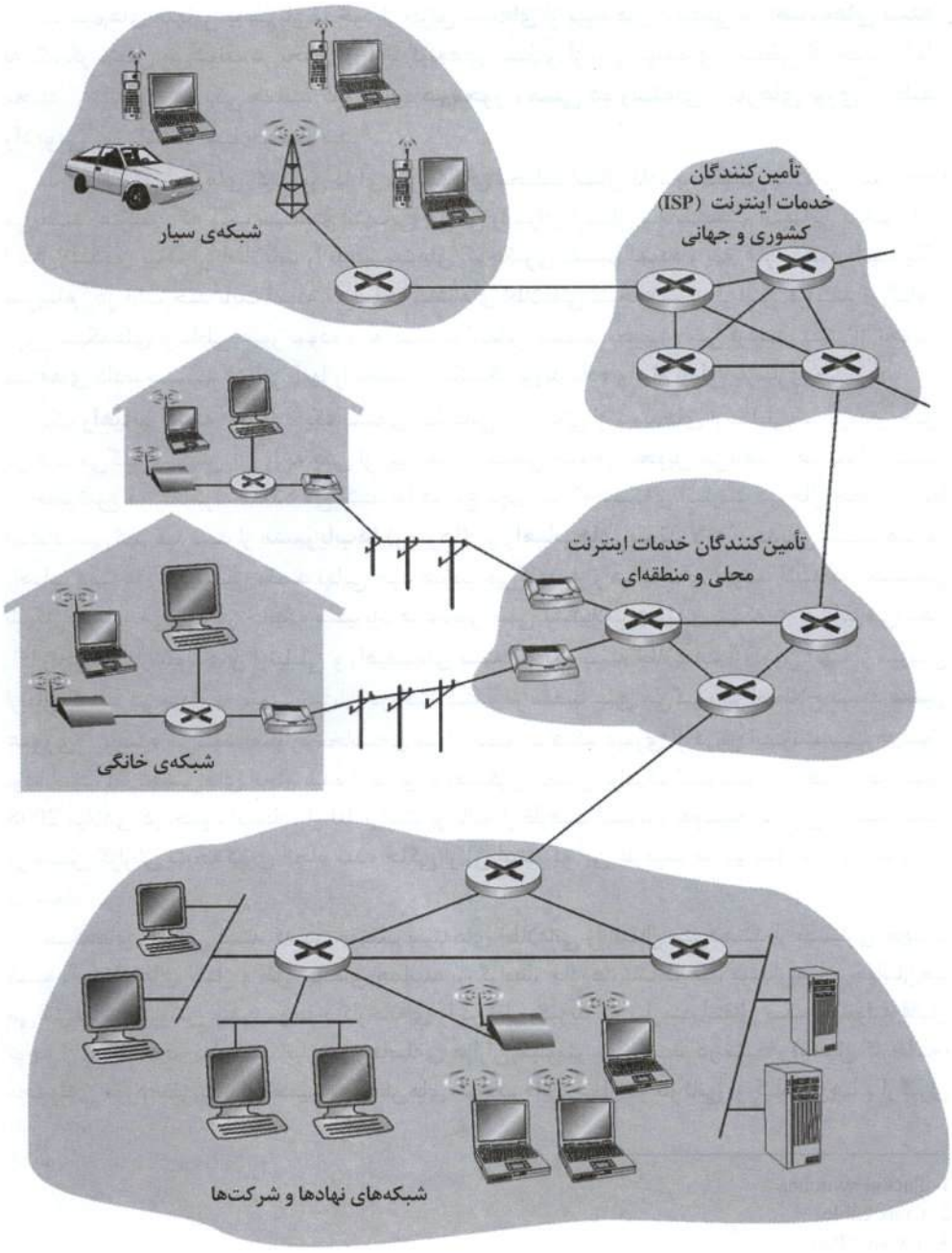
پس از شناخت موارد مربوط به عناصر و پیوندهای تشکیلاتی شبکه‌های کامپیوتری در ادامه‌ی مطالب فصل یک، موضوعات مربوط به تأخیر، تضعیف و بازدهی و **نهفتگی**[1] در یک شبکه‌ی کامپیوتری بررسی می‌گردد. همراه با آن الگوهایی برای نهفتگی انتها به انتها و تأخیر که مواردی چون تأخیرهای انتقال، انتشار وصف را در بر می‌گیرند، بررسی شده است. در بخش انتهایی فصل، اصول کلیدی معماری شبکه‌های کامپیوتری شامل پروتکل‌ها، لایه‌بندی‌ها و الگوهای خدمات شبکه معرفی می‌شود. نقاط ضعف شبکه‌های کامپیوتری در برابر انواع تهدیدها و بررسی چگونگی تأثیر این عوامل مخرب بر روی شبکه‌ها و نیز راه کارهای ایمن‌سازی بیشتر آنها از جمله مواردی است که تحلیل می‌شوند. پایان‌بخش مطالب این فصل تاریخچه‌ای از روند پیشرفت تهدیدها و امنیت شبکه‌های کامپیوتری است.

۱-۱ اینترنت چیست؟

این پرسش را می‌توان به دو صورت پاسخ داد. نخست با توصیف عناصر سخت‌افزاری و نرم‌افزاری که شبکه‌ی اینترنت را می‌سازند و دوم با تعریف اینترنت به‌عنوان یک شبکه‌ی زیر ساخت که خدماتی را برای برنامه‌های کاربردی به‌صورت توزیع شده فراهم می‌آورد. ابتدا از نقطه‌نظر توصیف عناصر سخت‌افزاری و نرم‌افزاری به کمک شکل ۱-۱ سعی در ارائه‌ی تعریف مشخصی از شبکه‌ی اینترنت در نظر می‌باشد. اینترنت از این رویکرد، شبکه‌ای است که صدها میلیون وسیله‌ی پردازشگر را در سراسر گیتی به یکدیگر متصل می‌نماید.

تا همین اواخر این تجهیزات پردازشگر عبارت بودند از: کامپیوترهای شخصی، ماشین‌های لینوکس و سرویس‌دهنده‌ها (سرورها) که وظیفه‌ی ذخیره و انتقال اطلاعاتی همانند صفحه‌های وب، پیام‌های الکترونیکی و نظایر آن را برعهده داشتند. اما به‌گونه‌ای فزاینده این موارد در کنار خود سیستم‌های دیگری نظیر رسانه‌ی تصویری (TV)، کامپیوترهای همراه (لپ‌تاپ)، کنسول‌های بازی، گوشی‌های همراه، دوربین‌های دیجیتال وب، خودروها، تجهیزات حس‌گر محیط زیست، فریم‌های تصویری و سایر سیستم‌های الکتریکی و امنیتی خانگی را جهت اتصال به اینترنت مشاهده کردند. بدیهی است واژگان **شبکه‌ی کامپیوتری** به نظر منسوخ می‌رسد، زیرا بسیاری از این عناصر متصل به اینترنت حالت غیر رسمی را به خود گرفته‌اند و بنابراین می‌توان عنوان نمود که اینترنت بازار مکاره‌ای است از همه‌ی این تجهیزات، که به آنها **میزبان**[2] یا **سیستم‌های انتهایی**[3] گفته می‌شود. تا اواخر ماه ژوئیه‌ی سال 2009 براساس گزارش [ISC 2009] نزدیک به 600 میلیون سیستم انتهایی به شبکه‌ی اینترنت متصل بودند، این تعداد بدون در نظر گرفتن تلفن‌های همراه، کامپیوترهای همراه و سایر ابزاری است که به‌صورت متناوب به شبکه‌ی اینترنت متصل می‌شوند.

1- Throughput
2- Host
3- End systems



شکل ۱-۱ ♦ تعریف شبکه‌ی اینترنت

سیستم‌های انتهایی یا میزبان‌ها خود از طریق شبکه‌ای از پیوندهای ارتباطی و **راهیاب‌های بسته**[1] به یکدیگر متصل می‌شوند. در بخش ۲-۱ با گونه‌های بسیاری از این پیوندهای ارتباطی که خود از انواع مختلف رسانه‌های فیزیکی همانند کابل‌های: **هم‌محور**[2]، **مسی دو رشته‌ای**[3]، **تارهای نوری**[4] و **طیف رادیویی**[5] می‌باشند، آشنا خواهیم شد.

بدیهی است پیوندهای گوناگون دارای سرعت‌های مختلف انتقال داده بر مقیاس بیت بر ثانیه (bps) می‌باشند. هنگامی که یک سیستم انتهایی پیامی را برای ارسال به سیستم انتهایی دیگر دارد، ارسال‌کننده‌ی پیام، آن اطلاعات را به قسمت‌های کوچکتری تقسیم نموده و به هر یک از آنها یک **سرپیام**[6] در قالب چند بایت اضافه می‌نماید. بسته‌های اطلاعاتی نتیجه شده از این فرآیند آن‌گاه از درون شبکه‌های ارتباطی عبور نموده و به سیستم انتهایی مقصد تحویل می‌گردند. پس از تحویل بسته‌های داده، سیستم انتهایی آنها را مجدداً به یکدیگر پیوند داده و پیام اصلی بازسازی می‌گردد.

یک **راهیاب بسته**[7] همواره یک بسته‌ی اطلاعاتی را از یکی از پیوندهای ارتباطی در ورودی خود دریافت می‌کند و سپس آن را به یکی از پیوندهای ارتباطی خروجی تحویل می‌دهد. راهیاب‌های بسته از لحاظ تنوع در عملکرد، متعدد می‌باشند اما دو نوع مهم آنها که شبکه‌ی اینترنت در حال حاضر از آنها استفاده می‌کند عبارتند از **مسیریاب‌ها (روترها)**[8] و **راهیاب‌های پیوند لایه**[9]. بدیهی است هر دو راهیاب بسته‌ها را به سمت مقصد نهایی خود هدایت می‌کنند. از راهیاب‌های پیوند لایه در دستیابی شبکه استفاده می‌شود در حالیکه مسیریاب‌ها عناصر اصلی تشکیلات هسته‌ی شبکه به حساب می‌آیند.

زنجیره‌ای از پیوندهای ارتباطی و راهیاب‌های بسته که یک بسته اطلاعات از درون آنها از میزبان ارسال کننده در مبداء به سمت میزبان دریافت کننده در مقصد عبور می‌کند، در اصطلاح شبکه **مسیر عبوری**[10] نامیده می‌شود. همواره محاسبه‌ی میزان دقیق ترافیک عبوری از درون اینترنت بسیار دشوار بوده است، اما تخمین‌های انجام شده از سوی پژوهشگران تحلیل ترافیک اینترنت، این رقم را در سال 2008 میلادی در حدود استفاده از 10 ترابیت بر ثانیه از ظرفیت اینترنت به‌وسیله کاربران دانسته است. در همین گزارش نتیجه‌گیری انجام شده حاکی از آن است که این ظرفیت هر دو سال یک بار دو برابر می‌شود.

شبکه‌های راهیابی بسته که در حقیقت بسته‌های اطلاعاتی را انتقال می‌دهند از بسیاری جهات شبیه به شبکه‌های حمل و نقل جاده‌ای همانند، بزرگراه‌ها، جاده‌ها، تقاطع‌ها، میدان‌ها و خیابان‌ها می‌باشند. برای روشن شدن موضوع کارخانه‌ای را در نظر بگیرید که نیازمند انتقال یک محموله قابل توجه از لحاظ حجم به مقصد انباری در فاصله‌ی هزاران کیلومتر می‌باشد. در مبداء (محل کارخانه) محموله‌ی مورد نظر پس از تقسیم به بخش‌های کوچک قابل حمل در ناوگانی از کامیون‌ها بارگیری

1- Packet switches
2- Coax cable
3- Twisted Pairs
4- Fibre optics
5- Radio spectrum
6- Header Bytes
7- Packet switches
8- Routers
9- Link Layer Switches
10-Path Through

می‌شود. هر یک از کامیون‌ها به گونه‌ای مستقل از درون شبکه‌ی حمل و نقل جاده‌ای شامل بزرگراه‌ها، جاده‌ها، تقاطع‌ها عبور کرده و سرانجام در محل انبار هر یک محموله‌ی خویش را تخلیه می‌کند (تحویل می‌دهد). بدیهی است در محل انبار این محموله‌ها با بقیه محموله‌های دریافت شده از آن کارخانه تجمیع شده و به‌صورت همان محموله‌ی جامع اولیه‌ی مبداء سازماندهی می‌شوند. بنابراین بسته‌ها را می‌توان همانند کامیون‌ها، پیوندهای ارتباطی را شبیه بزرگراه‌ها و جاده‌ها، راهیاب‌های بسته را نظیر تقاطع‌ها و سرانجام سیستم‌های انتهایی یا میزبان‌ها را شبیه ساختمان‌ها تصور نمود. درست همانگونه که یک کامیون مسیر عبور خود را از درون شبکه حمل و نقل جاده‌ای انتخاب می‌کند، یک بسته نیز مسیر عبور خود را از درون یک شبکه‌ی کامپیوتری گزینش می‌نماید.

سیستم‌های انتهایی، اینترنت را از طریق **فراهم‌آورندگان خدمات اینترنت**[1] دستیابی می‌کنند. این تأمین‌کنندگان خدمات اینترنت می‌توانند به انواع شاخه‌ها گروه‌بندی شوند، از جمله‌ی آنها: تأمین‌کنندگان خانگی همانند شبکه‌های کابلی محلی و یا شرکت‌های مخابراتی، فراهم آورندگان تجاری، آموزشی نظیر دانشگاه‌ها و گروه‌های دیگر می‌باشند که دستیابی را از طریق **شبکه‌های بی‌سیم**[2] در فرودگاه‌ها، هتل‌ها، رستوران‌ها و کافی شاپ‌ها و سرانجام دیگر مکان‌های عمومی امکان‌پذیر می‌سازند. توجه به این نکته ضروری است که هر یک از این تأمین‌کنندگان، خود یک شبکه‌ای از راهیاب‌های بسته و پیوندهای ارتباطی را تشکیل می‌دهند.

تأمین‌کنندگان خدمات اینترنت خود انواع مختلفی از دستیابی‌ها را برای سیستم‌های انتهایی فراهم می‌آورند، از جمله آنها عبارتند از: دستیابی از طریق مودم‌های 56 کیلوبیت بر ثانیه به شکل شماره‌گیری مستقیم، که منسوخ شده‌اند. **دستیابی باند گسترده**[3] و پر سرعت مسکونی نظیر **خطوط مشترکین دیجیتال**[4] یا شبکه‌های کابلی، شبکه‌های محلی سریع و دستیابی‌های بی‌سیم. علاوه بر همه‌ی موارد بالا این شبکه‌های تأمین دستیابی، امکان ارتباط سامانه‌های وب و شبکه‌های گسترده‌ی اطلاع‌رسانی اجتماعی را به گونه‌ای مستقیم به اینترنت نیز فراهم می‌آورند.

در یک نگاه اینترنت عبارت است از اتصال همه‌ی سیستم‌های انتهایی به یکدیگر، بنابراین ISPها که خود عوامل دستیابی به اینترنت را فراهم می‌آورند بایستی دارای ارتباط بین خویش نیز باشند.

این رده‌های پائین از ISPها از طریق لایه‌های بالاتر خود در سطح کشوری و یا بین‌المللی همانند شبکه‌های زیر ساخت مخابراتی کشورها به یکدیگر متصل می‌شوند. معمولاً یک رده‌ی بالاتر از ISPها شامل مسیریاب‌های بسیار سریع می‌باشند که از طریق شبکه‌ی زیر ساخت که خود از پیوندهای نوری پر شتاب تشکیل شده، به یکدیگر متصل می‌شوند. هر شبکه‌ی ISP در هر رده‌ای که قرار گرفته باشد، دارای سه ویژگی از لحاظ عملکرد است، که عبارتند از (۱) مدیریت مستقل، (۲) اجرا و پیاده‌سازی پروتکل‌های IP و (۳) استفاده از نوعی قواعد ویژه‌ی آدرس‌دهی و نامگذاری[5]. در بخش ۱-۳ ارتباط بین ISPها به‌صورت مفصل بررسی می‌گردد.

سیستم‌های انتهایی، راهیاب‌های بسته و دیگر تجهیزات شبکه‌ی اینترنت از طریق اجرای پروتکل‌ها،

1- Internet Service Providers(ISPs)
2- Wireless Networks (Wi - Fi)
3- Broadband access
4- Digital subscriber Lines (DSL)
5- Naming

ارسال و دریافت اطلاعات را بین شبکه‌ی اینترنت کنترل می‌کنند. **پروتکل‌های کنترل انتقال**[1] و IP از جمله‌ی مهمترین پروتکل‌های شبکه‌ی اینترنت می‌باشند. پروتکل IP قالب‌بندی بسته‌های اطلاعات ارسالی و دریافتی را در بین مسیریاب‌ها و سیستم‌های انتهایی مشخص می‌کند. مجموعه‌ی پروتکل‌های اصلی اینترنت، اصطلاحاً پشته‌ی پروتکلی TCP/IP نامیده شده‌اند. با توجه به اهمیت پروتکل‌ها در اینترنت، توافق جامع بر روی آنکه هر یک از این پروتکل‌ها چه وظایفی را بایستی برعهده گیرند حائز اهمیت است. در این حالت است که استانداردها نقش مهمی را ایفا می‌نمایند. استانداردهای اینترنت غالباً از طریق **کارگروه مهندسی اینترنت**[2] طراحی، تدوین و تصویب می‌گردد. مجموعه اسناد و مدارک این کارگروه (IETF)، اصطلاحاً **درخواست برای پیشنهاد**[3] نامیده می‌شود. این درخواست‌ها حالت کلی و عمومی دارد و تمرکز آنها بیشتر بر روی موضوعات فنی و جزئیات کاربردی شبکه‌ی اینترنت است.

آنها پروتکل‌هایی از قبیل TCP، IP و HTTP را برای وب و SMTP را برای پست الکترونیکی تعریف می‌کنند. در حال حاضر بیش از 5000 سند RFC وجود دارد. دیگر نهادهای مرتبط با استاندارد نمودن تجهیزات شبکه‌های کامپیوتری، کار گروه‌های **سازمان مهندسین برق و الکترونیک**[4] می‌باشد، که بیشتر به موضوعات مربوط به پیوندهای شبکه می‌پردازد. به‌عنوان نمونه کارگروه استانداردهای IEEE 802 LAN/MAN از جمله کار گروه‌هایی است که استانداردهای مربوط به **شبکه‌های اترنت**[5] و Wi-Fi را بر عهده دارد.

۱-۲ تعریف خدمات

مواردی که در بالا بررسی شد بسیاری از عناصر تشکیل دهنده‌ی شبکه‌ی اینترنت را مشخص نمی‌کند. اما اینترنت را می‌توان از زاویه‌ی کاملاً متفاوت دیگری هم تحلیل نمود و آن عبارت است از یک زیر ساخت که خدمات در اختیار کاربردها قرار می‌دهد. این کاربردها شامل: پست الکترونیکی، وب گردی، مبادله پیام بی‌درنگ، **انتقال صدا از طریق IP**[6]، ارسال و دریافت صدا و تصویر ویدئویی، بازی‌های چند کاربری همزمان، اشتراک فایل نظیر به نظیر (P2P)، رسانه‌ی تصویری (TV) از طریق اینترنت، دسترسی راه دور از طریق وارد و یا دریافت نمودن اطلاعات و بسیاری از کاربردهای دیگر می‌باشد. این کاربردها را معمولاً توزیع شده می‌نامند. این به مفهوم آن است که چندین سیستم انتهایی به مبادله‌ی اطلاعات بین خود اقدام می‌کنند. از همه مهم‌تر اجرای کاربردها بر روی سیستم‌های انتهایی است، آنها بر روی راهیاب‌های بسته در هسته‌ی اینترنت اجرا نمی‌شوند. اگرچه این راهیاب‌ها امکانات مبادله را بین سیستم‌های انتهایی فراهم می‌آورند امّا اهمیتی به یک کاربرد که در مبداء یا مقصد اطلاعات است نمی‌دهند. برای روشن شدن بیشتر مفهوم زیر ساخت که خدمات را برای کاربردها فراهم می‌آورد، توضیحات بیشتری در ذیل می‌آید. فرض کنید یک طرح جدید و جالبی را به‌عنوان یک کاربرد اینترنتی توزیع شده دارید که این ایده برای جامعه بسیار مفید و در عین حال سبب شهرت و درآمد هنگفت برای شما خواهد شد. چگونه شما این طرح را به یک کاربرد واقعی در اینترنت تبدیل می‌کنید؟

1- Transmission Control Protocol (TCP)
2- Internet Engineering Task Force (IETF)
3- Requests for comments (RFC)
4- Institute of Electrical and Electronics Engineers (IEEE)
5- Ethernet
6- Voice Over Internet Protocol (VOIP)

از آنجائیکه کاربردها بر روی سیستم‌های انتهایی پیاده‌سازی می‌شوند، شما نیازمند نوشتن یک قطعه‌ی نرم‌افزاری هستید که بر روی سیستم‌های انتهایی (میزبان‌ها) قابل اجرا باشد. به‌عنوان مثال ممکن است این قطعه برنامه را به زبان جاوا، C و یا پیتون[1] بنویسید. از این رو که هدف، یک کاربرد قابل توزیع اینترنتی است، قطعه نرم‌افزارهایی که بر روی میزبان‌های دیگر در حال اجرا هستند، نیازمند ارسال اطلاعات به یکدیگر می‌باشند. در اینجا یک موضوع اساسی مطرح می‌شود و آن شناسایی یک روش جایگزین جهت توصیف اینترنت به‌عنوان یک پایگاه برای نرم‌افزارهای کاربردی است. چگونه یک قطعه‌ی کاربردی در حال اجرا بر روی یک سیستم انتهایی، شبکه‌ی اینترنت را، به تحویل اطلاعات به یک قطعه نرم‌افزار دیگری که بر روی یک میزبان دیگر در حال اجرا می‌باشد، مجبور می‌کند؟

سیستم‌های انتهایی متصل به شبکه‌ی اینترنت یک واسط برنامه‌نویسی کاربردی (API) را فراهم می‌آورند که مشخص می‌کند، چگونه یک قطعه‌ی نرم‌افزاری در حال اجرا بر روی یک میزبان از زیر ساخت اینترنت تقاضای تحویل دادن اطلاعاتی را به یک قطعه‌ی نرم‌افزاری مشخص در حال اجرا بر روی میزبان دیگر می‌کند. واسط‌های اینترنت API مجموعه‌ای از قواعد می‌باشند که نرم‌افزار ارسال کننده بایستی از آنها پیروی کند تا اینترنت بتواند اطلاعات را به قطعه نرم‌افزار مقصد تحویل نماید. در فصل دوم کتاب موارد مربوط به API با جزئیات بیشتری مطالعه می‌شود. اکنون یک مثال شبیه‌سازی شده از این مهم آورده می‌شود. فرض کنید دو نفر A و B قصد مبادله یک نامه را از طریق شبکه‌ی پست به یکدیگر دارند. بدیهی است A نمی‌تواند صرفاً مطالب نامه خود را بر روی یک قطعه کاغذ بنویسد و آن را از پنجره به بیرون پرتاب نماید. از طرفی شبکه پست A را مجبور می‌کند که نامه‌ی خود را در درون یک پاکت قرار داده، نام کامل B را بر روی آن به همراه آدرس و کدپستی کامل در مرکز و یا گوشه‌ای از پاکت بنویسد، در پاکت را چسب زده و به همراه باطل نمودن تمبر لازم، آن را در صندوق پستی بیندازد. بنابراین شبکه‌ی خدمات پست برای خدمات پستی خود یک API تعریف نموده است که شخص A بایستی آنها را رعایت نموده تا شبکه خدمات پست نامه را به B تحویل نماید.

در حالت مشابه شبکه‌ی اینترنت از طریق API، نرم‌افزار ارسال کننده‌ی اطلاعات را موظف به رعایت قواعدی می‌کند که از طریق آنها اطلاعات ارسالی به قطعه نرم‌افزار کاربردی دریافت کننده، تحویل گردد. بدیهی است همانگونه که شبکه‌ی خدمات پستی انواع سرویس‌های پستی را فراهم می‌آورد، شبکه‌ی اینترنت نیز خدمات چندگانه‌ای را برای نرم‌افزارهای کاربردی امکان‌پذیر می‌سازد. بنابراین هنگامی که یک قطعه کاربردی طراحی می‌گردد، طراح آن نرم‌افزار بایستی یکی از خدمات ارائه شده بوسیله‌ی اینترنت را برای آن کاربرد گزینش نماید.

این تعریف دوم اینترنت که فراهم‌آوری یک زیر ساخت برای خدمات به کاربردهای توزیع شده می‌باشد، یکی از مهمترین موضوعات می‌باشد. با روند رو به رشد فناوری قطعات تشکیل دهنده‌ی اینترنت نیاز به نرم‌افزارهای کاربردی جدید، گسترش بیشتری یافته است. بنابراین به یاد داشتن این نکته که اینترنت یک شبکه‌ی زیر ساخت می‌باشد که کاربردها در آن مرتباً در حال تغییر و نوآوری می‌باشند، بسیار حیاتی است. تا اینجا دو تعریف از شبکه‌ی اینترنت ارائه گردید. یکی از این توصیف‌ها در ارتباط با عناصر سخت‌افزاری بود و دیگری تعریف به‌عنوان شبکه‌ی زیر ساخت که خدمات برای نرم‌افزارهای کاربردی توزیع شده فراهم می‌آورد. شاید هنوز این ابهام وجود داشته باشد که اینترنت،

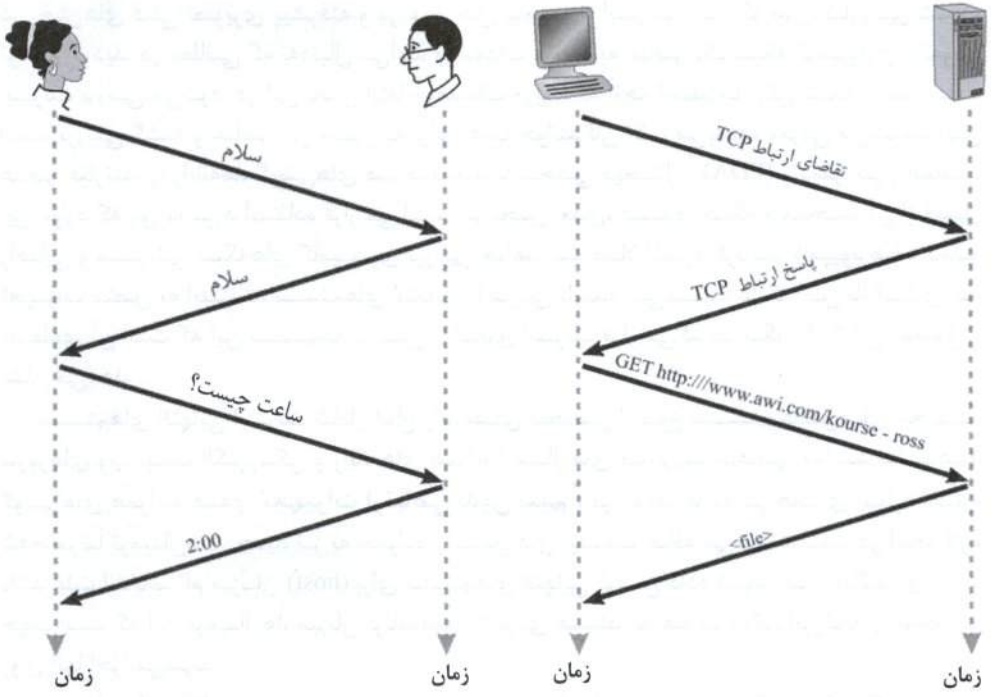
1- Python

راهیابی بسته، TCP/IP و یک API چیست؟ مسیریاب‌ها چه ابزاری هستند؟ انواع پیوندهای ارتباطی در حال حاضر در اینترنت چیست؟ یک نرم‌افزار کاربردی توزیع شده چیست؟ چگونه یک وسیله همانند ماشین لباسشویی و یا حسگر هواشناسی به شبکه‌ی اینترنت متصل می‌شود؟ اگر کمی هیجان زده شده اید، جای نگرانی نیست. هدف این کتاب معرفی هر دو موضوع عناصر سخت‌افزاری و نرم‌افزاری و اصولی که این عناصر را کنترل و مدیریت می‌کنند، می‌باشد. در فصل‌های بعدی به همه‌ی این پرسش‌ها پاسخ داده شده است.

۱-۳ پروتکل چیست ؟

اکنون که دیدگاهی کلی از چگونگی عملکرد شبکه‌اینترنت بدست آوردیم، لازم است به بررسی یکی از واژگان کلیدی آن، پروتکل بپردازیم. به عبارت دیگر پاسخ به این دو سؤال که پروتکل چیست ؟ و پروتکل چه کاربردی دارد ؟ شاید ساده‌ترین روش برای درک مفهوم یک پروتکل شبکه، شبیه‌سازی آن با یک رفتار انسانی باشد. زیرا افراد تقریباً در همه‌ی موارد به نوعی پروتکل‌هایی را اجرا می‌کنند. حالتی را در نظر بگیرید که شخصی می‌خواهد در مورد زمان از فرد دیگری پرسش نماید. شکل ۱-۲ نمونه‌ای از این رفتار پروتکلی را نشان می‌دهد. پروتکل انسانی یا به عبارت دیگر رفتار انسانی این‌گونه ایجاب می‌کند که فرد قبل از مطرح نمودن سؤال اصلی خود به نوعی به مبادله‌ی اطلاعاتی بپردازد که بیانگر رفتار مؤدبانه‌ی انسانی باشد. این رعایت چگونگی معاشرت به نوعی آغازگر ارتباط با طرف مقابل خواهد شد. به عبارت دیگر گفتن "سلام" از سوی پرسشگر پاسخ مناسب خود را از سوی مخاطب به دنبال دارد. دریافت پاسخ مناسب از سوی مخاطب به این مفهوم است که وی آمادگی پاسخ و مبادله‌ی اطلاعات را یافته است. هر پاسخ غیر مناسب دیگری از سوی مخاطب به معنی عدم آمادگی وی برای ادامه مبادله‌ی اطلاعات می‌باشد. همانگونه که این منظور در شکل ۱-۲ نشان داده شده است. بدیهی است در پروتکل انسانی پیام‌های دریافتی صادر می‌شود و یا آنکه پاسخی داده نمی‌شود. واضح است همه‌ی موارد و رفتاری که انسان‌ها در این رابطه از خود بُروز می‌دهند نقش اساسی را در پروتکل انسانی بازی می‌کند. اگر افراد دارای پروتکل‌های مختلف باشند که نهایتاً اِعمال آنها سبب عدم ارتباط بین آنها شود، آن‌گاه هیچ نوع مبادله‌ای صورت نخواهد پذیرفت. این واقعیت در مورد شبکه‌های کامپیوتری نیز درست است. دو کاربرد در حال ارتباط بایستی از یک پروتکل مشترک و مورد توافق استفاده نمایند تا وظیفه‌ای که بر عهده‌ی آنها گذارده شده است به خوبی اجرا گردد. مثال دیگر یک جلسه‌ی پرسش و پاسخ است که در یک کلاس درس بین استاد و دانشجویان رخ می‌دهد. یک پروتکل شبکه بسیار مشابه مثال هایی است که در بالا عنوان شدند، تنها تفاوت شبکه در آن است که مبادله‌ی پیام‌ها از طریق برنامه‌های کاربردی، وابسته به عملکردی است که عناصر سخت‌افزاری و نرم‌افزاری همانند کامپیوترها، گوشی تلفن، مسیریاب‌ها و سایرین از خود نشان می‌دهند. بنابراین می‌توان نتیجه گرفت که همه‌ی فعالیت‌های انجام شده در اینترنت مربوط به دو یا بیش از دو موجودیت مرتبط با یکدیگر است که از طریق پروتکل‌ها کنترل و مدیریت می‌شوند. به‌عنوان مثال: پروتکل‌های سخت‌افزاری پیاده‌سازی شده بر روی کارت‌های شبکه، هنگامی که دو کامپیوتر از طریق این کارت‌ها به‌صورت فیزیکی به یکدیگر متصل شده‌اند، جریان انتقال بیت‌ها را بر روی رسانه‌ی انتقال که ممکن است یک رشته سیم باشد، بین آن دو کارت شبکه کنترل می‌کند. پروتکل‌های کنترل تراکم در سیستم‌های انتهایی سرعت انتقال بسته‌ها را بین فرستنده و گیرنده کنترل می‌کند و یا پروتکل‌های پیاده‌سازی شده در مسیریاب‌ها به جستجو و یافتن مسیر از مبداء تا به مقصد اقدام می‌کنند. در حقیقت پروتکل‌ها در همه‌ی بخش‌های

اینترنت در حال اجرا می‌باشند و در نتیجه بخش وسیعی از این کتاب به بررسی آنها اختصاص یافته است. به‌عنوان نمونه یک پروتکل شبکه که بارها عملکرد آن را تجربه نموده‌اید، مربوط به هنگامی است که درخواستی به یک وب سرور ارسال شده است. این درخواست معمولاً در قالب وارد نمودن آدرس URL[1] یک صفحه وب در پنجره‌ی مرورگر می‌باشد. نیمه سمت راست شکل ۱-۲ این مهم را نشان می‌دهد. نخست کامپیوتر ارسال کننده‌ی آدرس، یک پیام در خواست ارتباط به سمت وب سرور ارسال می‌کند و سپس منتظر می‌ماند تا پاسخ دریافت کند.



شکل ۱-۲ ♦ نمونه‌ی یک پروتکل انسانی و یک پروتکل شبکه‌های کامپیوتری

پس از یک گذر زمانی وب سرور سرانجام درخواست را دریافت و پاسخ درخواست ارتباط را در قالب یک پیام بازگشت می‌دهد. کامپیوتر متقاضی با آگاهی از مناسب بودن شرایط ارتباط، نام صفحه‌ی وب مورد نظر را که می‌خواهد از وب سرور واکشی نماید در قالب یک پیام GET به سمت وب سرور گسیل می‌دارد. سرانجام وب سرور با بازگردانی صفحه‌ی وب مورد نظر در قالب یک پوشه (فایل) به کامپیوتر متقاضی عملیات مورد نظر را به گونه‌ی مطلوب به انجام می‌رساند. با توجه به دو نمونه‌ی عنوان شده در بالا، مبادله‌ی پیام‌ها و عملیات انجام شده، هنگامی که این پیام‌ها ارسال و دریافت می‌شوند در قالب یک تعریف کلی از یک پروتکل به‌صورت زیر قابل بیان می‌باشد:

«یک پروتکل به تعریف قالب و ترتیب مبادله‌ی پیام‌ها بین دو یا بیش از دو موجودیت ارتباطی می‌پردازد. علاوه بر آن عملیات لازم برای انتقال و یا دریافت یک پیام را نیز مشخص می‌کند.»

1- Uniform Resource Locator (URL)

شبکه‌ی اینترنت و در حالت کلی شبکه‌های کامپیوتری به‌صورت گسترده از پروتکل‌های مختلف جهت انجام عملیات ارتباطی گوناگون استفاده می‌کنند. برخی از پروتکل‌ها ساده به نظر می‌رسند و برخی دیگر دشوار. مهارت در شبکه‌های کامپیوتری به مفهوم آگاهی از عملکرد پروتکل‌ها از لحاظ پاسخ به پرسش‌هایی نظیر چه، چرا و چگونه می‌باشد.

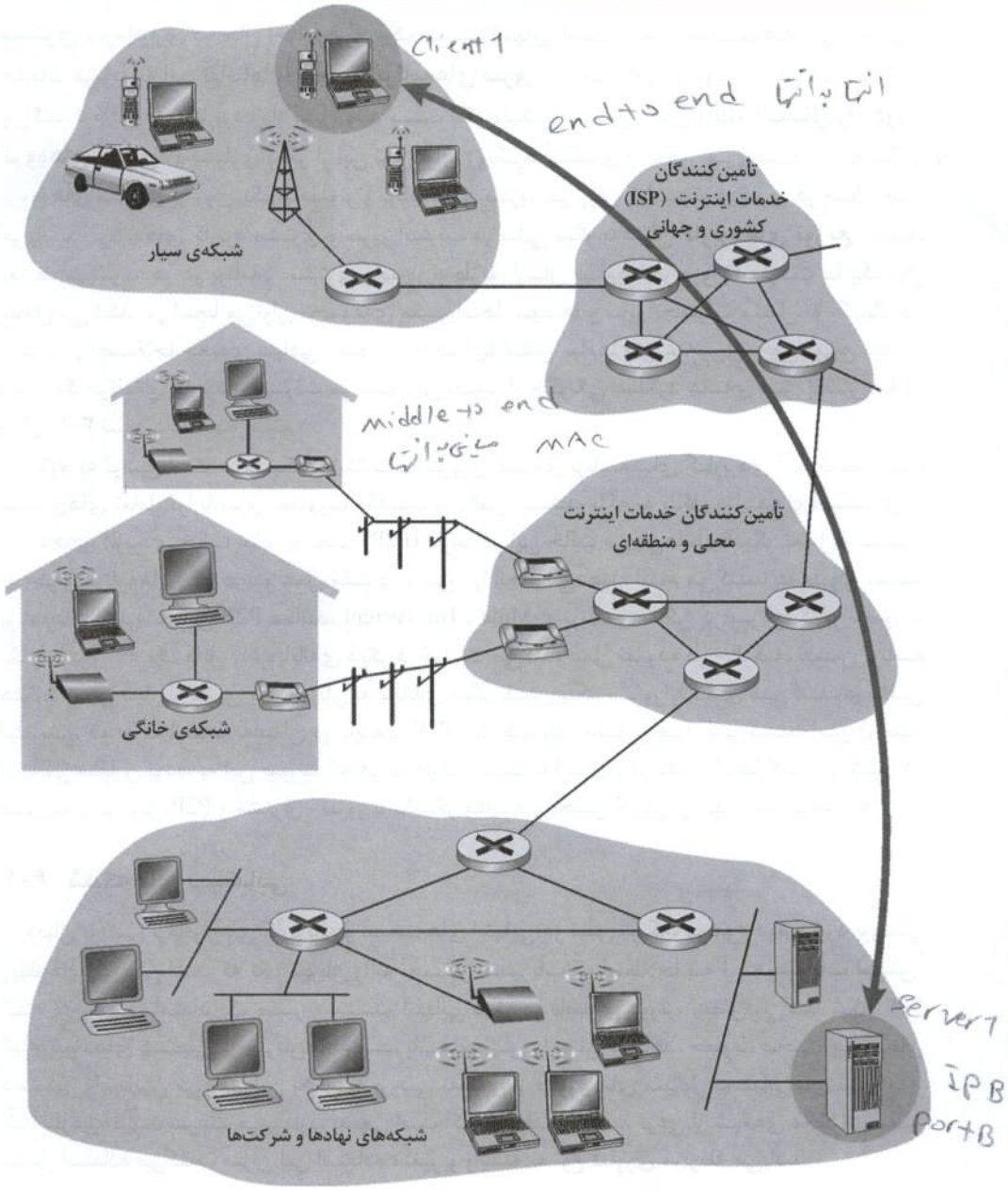
۱-۴ نبش شبکه[1]

در بخش‌های قبلی تصویری پیشرفته و در عین حال پیچیده از اینترنت و پروتکل‌های کاربردی شبکه ترسیم گردید. در مطالبی که به‌دنبال می‌آید، موضوعات مربوط به عناصر یک شبکه کامپیوتری و به‌ویژه اینترنت بررسی می‌شود. در این بخش ابتدا موضوعات مربوط به آنچه اصطلاحاً نبش شبکه نامیده شده است، بررسی گشته و عناصر این بخش که برای اغلب خوانندگان آشنا می‌باشد، معرفی می‌گردد. این عناصر عبارتند از: رایانه‌ها، گوشی‌های همراه، برنامه‌ریز شخصی دیجیتال[2] (PDA) و دیگر ابزار همانند این موارد که روزانه مورد استفاده قرار می‌گیرند. در بخش بعدی، هسته‌ی شبکه و مباحث آن از قبیل راهیابی و مسیریابی شبکه‌های کامپیوتری بررسی خواهد شد. قبلاً اشاره گردید کامپیوترها و دیگر تجهیزات متصل به اینترنت، سیستم‌های انتهایی یا میزبان نامیده می‌شوند. علت این نامگذاری به واسطه‌ی آن است که این سیستم‌ها در نبش یا انتهای اینترنت قرار می‌گیرند. شکل ۱-۳ این موضوع را نشان می‌دهد.

سیستم‌های انتهایی اینترنت شامل انواع رایانه‌های شخصی از نوع سیستم عامل‌های مختلف، سرورهای وب، پست الکترونیکی و رایانه‌های همراه، ترمینال‌های مدیریت شخصی (مانند تبلت‌ها)، گوشی‌های همراه و همه‌ی **تجهیزات ارتباطی بدون سیم**[3]، می‌گردد. علاوه بر همه‌ی موارد اشاره شده، مرتباً ترمینال‌های جدید نیز به خانواده‌ی میزبان‌های اینترنت اضافه می‌گردند. شاید در اینجا لازم باشد علت انتخاب نام میزبان (host) برای سیستم‌های انتهایی توضیح داده شود. این نامگذاری از آن جهت است که این ترمینال‌ها، میزبان برنامه‌های کاربردی هستند. به عبارت دیگر، این‌گونه برنامه‌ها بر روی آنها اجرا می‌شوند.

در سراسر این کتاب از هر دو عبارت میزبان و سیستم انتهایی به‌صورت جایگزین یکدیگر استفاده می‌شود. بدیهی است میزبان‌ها خود گاهی اوقات به دو نوع: **مشتری**[4] و **سرویس‌دهنده**[5] تقسیم می‌شوند. به شکل غیر رسمی مشتری‌ها شامل ترمینال‌هایی همانند کامپیوترهای شخصی و سیار، گوشی‌های همراه، پایانه‌های مدیریت شخصی و غیره می‌باشند، در حالیکه سرورها ماشین‌های قدرتمندتر از لحاظ پردازش برای ذخیره و توزیع صفحه‌های وب، پخش تصاویر صوتی و ویدیویی، انتقال پست الکترونیکی و نظایر آن می‌باشند.

1- The Network Edge
2- Personal Digital Assistant
3- Wireless Communication Devices
4- Client
5- Server



شکل ۱-۳ ♦ تعامل بین سیستم‌های انتهایی

۱-۵ برنامه‌های مشتری و سرور

در چارچوب نرم‌افزاری شبکه، تعریف دیگری از یک مشتری و یک سرور ارائه می‌شود. تعریفی که در گفتمان سراسر این کتاب حاکم خواهد بود. این تعریف به این گونه است که **برنامه‌ی**

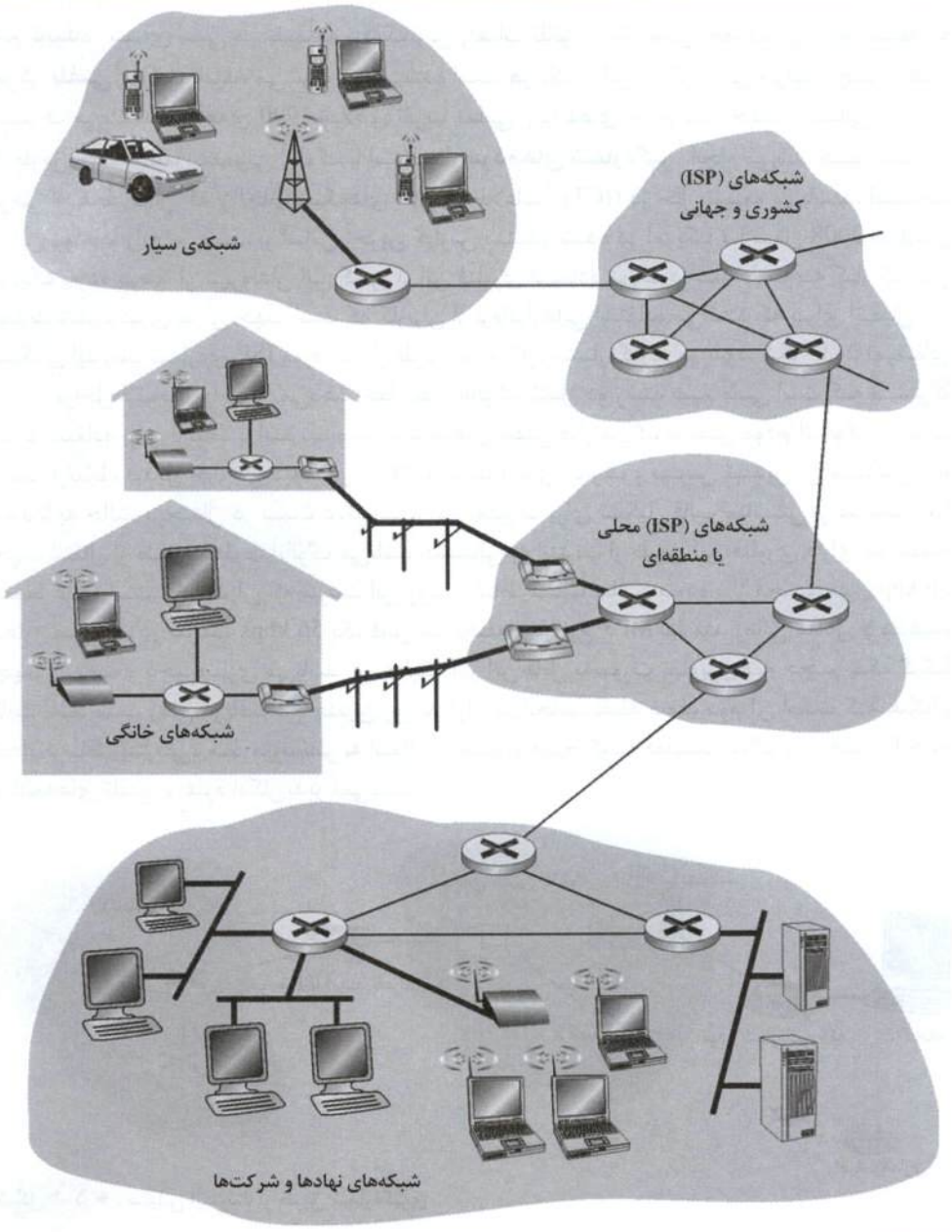
مشتری[1]، نرم‌افزاری در حال اجرا بر روی یک سیستم انتهایی است که درخواست‌هایی را ارسال و خدمات مربوط به این تقاضاها را از طریق **برنامه‌های سرور**[2] در حال اجرا بر روی میزبان دیگر دریافت می‌کند. نرم‌افزارهای کاربردی از قبیل وب، پست الکترونیکی، انتقال فایل (ftp)، **انتقال راه دور**[3]، **گروه‌های خبری**[4] و بسیاری دیگر از این نوع دارای رویکرد مشتری و سرور می‌باشند. از آنجائیکه برنامه‌های مشتری بر روی یک کامپیوتر و برنامه‌های سرور بر روی کامپیوتر دیگری در حال اجرا می‌باشند، برنامه‌های کاربرد مشتری و سرور اینترنت در بیانی دیگر به‌عنوان **کاربردهای توزیع**[5] شده، تعریف می‌شوند. هر دو برنامه‌ی مشتری و سرور از طریق ارسال پیام‌ها در درون اینترنت با یکدیگر تعامل می‌کنند. در اینجا می‌توان مجموعه‌ی مسیریاب‌ها، پیوندها و سایر تجهیزات متصل به یکدیگر در اینترنت را اصطلاحاً جعبه‌ی سیاهی نامید که هدف آنها انتقال پیام‌ها بین عناصر ارتباطی توزیع شده از سوی یک برنامه‌ی کاربردی اینترنت می‌باشد. این تصویر از چگونگی عملکرد همه‌ی این بخش‌ها در شکل ۱-۳ نشان داده شده است.

لازم به توضیح است در شبکه‌ی اینترنت امروزی همه‌ی برنامه‌های کاربردی اینترنت، مانند مشتری‌های تعامل‌گرا با سرور به‌صورت یکدست و خالص نیستند، بلکه به شکل فزاینده ای، بسیاری از برنامه‌های کاربردی حالت نظیر به نظیر (P2P) دارند. در این حالت میزبان‌ها با یکدیگر تعامل و نسبت به اجرای برنامه‌هایی که هر دو عمل مشتری و سرور را انجام می‌دهند، اقدام می‌کنند. به‌عنوان نمونه برنامه‌های اشتراک فایل P2P همانند، e-Mule , Bit Torrent، برنامه‌ی ساکن در میزبان کاربر به‌صورت یک مشتری که یک فایل را از پایانه‌ی دیگر درخواست می‌کند، عمل نموده در حالیکه، همان برنامه هنگام ارسال فایل از طریق یک میزبان به پایانه‌ی دیگر نقش یک سرور را بازی می‌کند. در تلفن اینترنتی، دو طرف ارتباط به‌عنوان دو پایانه‌ی P2P با یکدیگر تعامل برقرار می‌کنند. این فرآیند ارتباطی متقارن بوده، به این صورت که هر دو طرف نسبت به ارسال و دریافت داده‌ها اقدام می‌کنند. در فصل دوم دو روش P2P و مشتری- سرور با یکدیگر مقایسه و تحلیل کاملی از آنها ارائه خواهد شد.

۱-۶ شبکه‌های دستیابی

به دنبال بررسی نرم‌افزارهای کاربردی و سیستم‌های انتهایی در نبش شبکه، موضوع دوم مورد بررسی پیوندهای فیزیکی است که یک میزبان را به نخستین مسیریاب که اصطلاحاً به آن **مسیریاب نبش**[6] گفته می‌شود در امتداد یک مسیر و سیستم انتهایی دیگر در فاصله‌ی دورتر، متصل می‌کند. شکل ۱-۴ انواع پیوندهای دستیابی از میزبان‌ها به مسیریاب‌های نبش را نشان می‌دهد. خطوط ضخیم، پیوندهای دستیابی را نمایش می‌دهد. این بخش به بررسی تعدادی از فناوری‌های متداول دستیابی‌های شبکه، از کُند سرعت تا تُند سرعت می‌پردازد. بدیهی است اکثر این فناوری‌ها به نوعی از شبکه‌ی مخابرات تلفن سنتی استفاده می‌کنند. میزان این استفاده متغیر و وابسته به نوع فناوری مربوطه می‌باشد.

1- Client Programs
2- Server Programs
3- Remote Login
4- Newsgroup
5- Distributed applications
6- Edge router

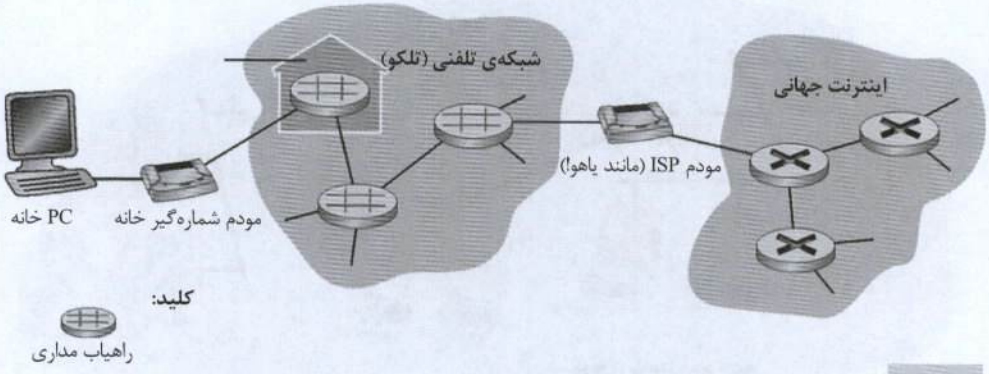


شکل ۱-۴ ♦ شبکه‌های دستیابی

این شبکه‌های تلفنی محلی را از این به بعد در این کتاب شبکه‌های محلی **تلکو**[1] (telco) می‌نامیم. لازم به یادآوری است که شبکه‌های محلی تِلکو در کشورهای مختلف، گوناگون بوده، اما از لحاظ استاندارد و پروتکل‌های ارتباطی همانند یکدیگرند. هر مشترک این شبکه از طریق **دو رشته سیم به**

1- Local telco (telecommunications networks)

هم تابیده[1] شده‌ی مِسی مستقیماً به نزدیک‌ترین راهیاب تلکو در یک مکان جغرافیایی که اصطلاحاً **مرکز تلفنی**[2] (CO) نامیده می‌شود، متصل شده است. هر یک از این مراکز دارای ظرفیت چندین هزار مشترک می‌باشند.در دهه‌ی 1990 میلادی تقریباً تمامی ارتباط‌های انجام شده جهت دستیابی اینترنت از طریق خطوط تلفن معمولی آنالوگ با استفاده از **مودم‌های شماره‌گیر**[3] انجام می‌شد. هنوز هم در برخی از نقاط جهان که از لحاظ شبکه‌های فناوری اطلاعات[4] (ICT) در حال توسعه می‌باشند، استفاده از این مودم‌ها رایج می‌باشد. بر اساس آخرین گزارش منتشر شده در آمریکا تا سال 2008 میلادی نزدیک به ده درصد از شهروندان این کشور از این فناوری استفاده می‌کرده‌اند. علت به کار گرفتن عبارت شماره‌گیری به آن جهت است که کاربران از نرم‌افزارهایی استفاده می‌کنند که برای اتصال به شبکه‌ی اینترنت شماره‌ی ISP مورد نظر از طریق مودم کاربر،شماره‌گیری می‌شود. شکل ۱-۵ نمونه‌ای از این مراحل ارتباطی را نشان می‌دهد. خط تلفن آنالوگ شامل دو رشته سیم مسی است که مشترکاً مورد استفاده برای ارتباط با اینترنت و نیز مکالمه‌های تلفنی قرار می‌گیرد. نقش مودم آنالوگ در یک سمت ارتباط، تبدیل خروجی دیجیتال از PC به سیگنال‌های آنالوگ و سپس تبدیل آن سیگنال‌ها، مجدداً به حالت دیجیتال در سمت دیگر است. این تغییرات برای تشکیل قالب مناسبی از سیگنال‌ها جهت انتقال از طریق خطوط آنالوگ می‌باشد. دستیابی به اینترنت از طریق این فناوری دارای دو نقطه ضعف عمده است. نخست آن که سرعت این روش ارتباط نسبتاً پائین بوده و از محدوده‌ی 56 kbps تجاوز نمی‌کند. در سرعت 56 kbps یک فایل سه دقیقه‌ای از نوع MP3 نیازمند زمانی معادل 8 دقیقه، جهت دریافت و ذخیره‌سازی می‌باشد. در حالیکه اگر این فایل به‌صورت تصویری به حجم یک گیگا بایت باشد مدت زمان دریافت آن چندین روز به درازا می‌انجامد. نقطه ضعف دوم آن است که هنگام انجام ارتباط اینترنتی، خط مورد نظر به اشغال درآمده و هیچ گونه فعالیت دیگری از قبیل انجام مکالمه‌های تلفنی و غیره امکان‌پذیر نمی‌باشد.



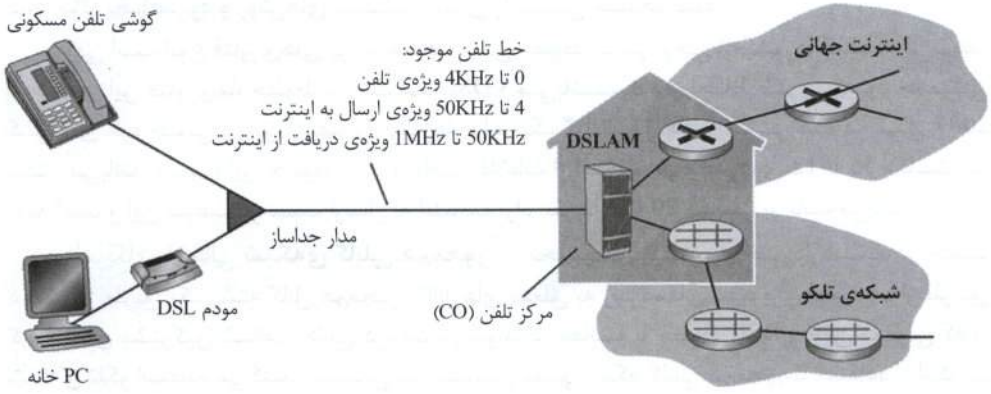
شکل ۱-۵ ♦ دستیابی اینترنت از طریق شماره‌گیری

در حال حاضر دو نوع از متداول‌ترین دستیابی به اینترنت از طریق آنچه که **پهنای باند وسیع**[5] نام

1- Twisted Pairs (TP)
2- Central Office
3- Dial up Modems
4- Information and communication technology
5- Broad band access

گرفته است صورت می‌پذیرد. این دو عبارتند از **خطوط مشترکین دیجیتال**[1] (DSL) و **خطوط کابلی**[2]. در بسیاری از کشورهای توسعه یافته بیش از 50 درصد از شهروندان مسکونی دارای دستیابی باند وسیع هستند. در این میان کشورهای کره جنوبی، ایسلند، هلند، دانمارک و سوئیس از بقیه‌ی کشورها از لحاظ این نوع دستیابی جلوتر می‌باشند. در این کشورها این رقم به بیش از 74 درصد تا سال 2008 رسیده است. در ایالات متحده بنا بر پژوهش انجام شده، سهم هر دو فناوری یکسان است. خارج از دو کشور آمریکا و کانادا، سهم فناوری DSL در اروپا از همه بالاتر و به بیش از 90 درصد می‌رسد. همان گونه که در شکل ۱-۶ نشان داده شده است، یک مشترک DSL از طریق خط تلفن دیجیتال دو رشته‌ای به یک بخش ویژه با عنوان **مالتی‌پلکس دستیابی مشترک دیجیتال**[3] (DSLAM) در مرکز تلفن دیجیتال شبکه‌ی تلکو واقع در یک CO متصل گردیده است. این خطوط معمولاً از لحاظ پهنای باند به سه گروه ذیل تقسیم می‌شوند:

- کانال پر سرعت دریافت از سوی اینترنت (down stream) در فاصله‌ی باند 50 کیلو هرتز تا 1 مگا هرتز
- کانال با سرعت متوسط ارسال به طرف اینترنت (up stream) در فاصله‌ی باند 4 تا 50 کیلو هرتز
- کانال معمولی دو طرفه‌ی تلفنی در فاصله‌ی باند صفر تا 4 کیلو هرتز.



شکل ۱-۶ ♦ دستیابی اینترنت DSL

این رویکرد یک پیوند DSL را به‌صورت سه پیوند مجزا از یکدیگر نشان می‌دهد. در این پیوند یک مکالمه‌ی تلفنی و یک ارتباط اینترنت به‌صورت مشترک از پیوند DSL هم‌زمان استفاده می‌کنند. این فناوری مالتی‌پلکس تقسیم فرکانس نامیده می‌شود. در ادامه‌ی مطالب همین فصل بررسی بیشتری بر روی آن انجام خواهد شد. در محل کاربر، سیگنال‌های وارد شده به محل مشترک از طریق یک مدار جدا کننده به دو بخش سیگنال داده‌ها و تلفنی تقسیم شده و سیگنال داده به سمت DSL گسیل می‌گردد. در نقطه‌ی تلکو و در CO بخش DSLAM سیگنال‌های داده و تلفنی را مجزا نموده و سپس

1- Digital Subscriber Lines
2- Cable Networks
3- Digital Subscriber Line Access Multiplexer (DSLAM)

سیگنال‌های مربوط به داده‌ها را به طرف اینترنت گسیل می‌نماید. صدها کاربر خانگی و مشابه آن می‌توانند به‌صورت هم‌زمان به یک DSLAM متصل شوند.

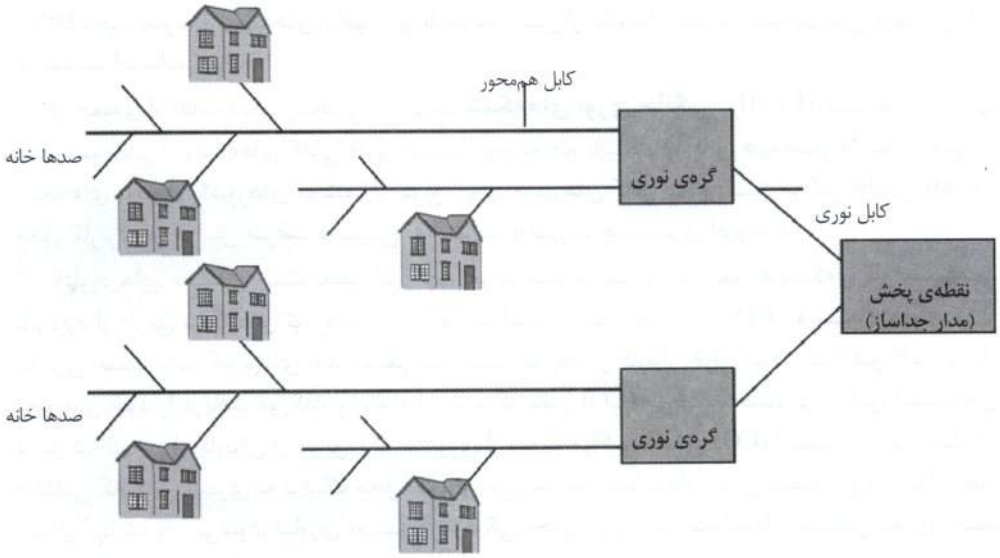
خطوط DSL دارای دو ویژگی اساسی نسبت به خطوط آنالوگ شماره گیر است، این دو عبارتند از: ارسال و دریافت اطلاعات با سرعت بسیار بالاتر. نرخ دریافت اطلاعات در خطوط DSL می‌تواند از سمت CO به طرف کاربر در محدوده‌ی یک تا دو مگا بیت بر ثانیه باشد. این سرعت در جهت متقابل یعنی کاربر به CO می‌تواند بین 128 Kbps تا 1 Mbps تنظیم گردد. از آنجایی که معمولاً در خطوط DSL دو سرعت به اینترنت و دریافت از آن متفاوت می‌باشند، دستیابی به این خطوط را اصطلاحاً دستیابی **غیر متقارن**[1] یا ADSL نامگذاری کرده‌اند. نقطه‌ی قوت دوم این خطوط، هم‌زمانی کاربرد داده و مکالمه‌ی تلفنی آنها می‌باشد. در این خطوط کاربر نیاز به شماره‌گیری ISP ندارد و همواره به‌صورت دایمی به مدار DSLAM متعلق به ISP مورد نظر متصل است. لازم به توضیح است، سرعت ارسال به و یا دریافت از اینترنت برای هر کاربر تابعی است از فاصله‌ی بین نقطه‌ی دستیابی کاربر و CO و نیز دو عامل، ضخامت قطر دو رشته کابل ارتباطی و اختلال‌های الکتریکی. طراحی خطوط DSL برای فاصله‌های کوتاه مابین CO و نقطه‌ی کاربری می‌باشد، نتیجه‌ی آن افزایش چشمگیر سرعت در مقایسه با دستیابی شماره‌گیری است. فناوری DSL برای این افزایش نرخ انتقال از الگوریتم‌های پیشرفته‌ی پردازش سیگنال و خطایابی استفاده می‌کند. این الگوریتم‌ها سبب تاخیر چشمگیر بسته‌ها نیز می‌شود. در صورتی‌که فاصله‌ی بین کاربر و CO مابین 7.5 تا 15 کیلومتر باشد، فناوری پردازش سیگنال DSL دیگر مؤثر نخواهد بود و روش‌های دستیابی دیگری را بایستی جستجو نمود.

بدیهی است انواع فناوری‌های پر سرعت دیگر این خطوط نیز در برخی از کشورها استفاده می‌شوند. از جمله‌ی این فناوری‌ها، خطوط پر سرعت VDSL می‌باشد، که از لحاظ ضریب نفوذ جمعیتی کشورهای کره جنوبی و ژاپن از بقیه‌ی کشورها، بنا بر یک گزارش پژوهشی منتشر شده در سال 2009 بیشتر می‌باشد. سرعت این خطوط برای دریافت اطلاعات از اینترنت در محدوده‌ی 12 تا 55 مگابیت بر ثانیه است و این سرعت در سمت ارسال به اینترنت برابر تقریباً 2 تا 20 مگابیت بر ثانیه می‌باشد.

در آمریکای شمالی **شبکه‌ی کابلی هم‌محور**[2] از محبوبیت بیشتری برخوردار است. در حالت عمومی از طریق یک رشته کابل هم‌محور، کانال‌های متعلق به رسانه‌های تصویری ارسال و از طریق گیرنده‌های مشترکین شبکه‌ی کابلی دریافت می‌شوند. در مقایسه با خطوط DSL و تلفن معمولی که از شبکه‌ی تلکو استفاده می‌کنند، دستیابی به اینترنت از طریق شبکه کابلی هم‌محور با استفاده از امکانات موجود رسانه‌های تصویری انجام می‌شود. به عبارت دیگر یک مشترک رسانه تصویری، ارتباط اینترنت خود را نیز از همان طریق میسر می‌سازد. همانگونه که از شکل ۱-۷ ملاحظه می‌شود از طریق یک زیر ساخت **ارتباط تجهیزات نوری**[3] به نقطه‌ی انتهایی، شبکه‌ی کابلی هم‌محور خدمات مورد نظر تصویری و یا اینترنتی را در اختیار هر یک از کاربران قرار می‌دهد.

هر مجموعه‌ی شبکه کابلی، یک محله نامیده می‌شود و هر محله معمولاً بین 500 تا 5000 کاربر را می‌تواند پوشش دهد. از آنجائیکه از هر دو راهکار ارتباطی نوری و کابلی هم‌محور در این شبکه استفاده می‌شود، اصطلاحاً آن را **شبکه‌ی ترکیبی نوری- هم‌محور**[4] **(HFC)** نامیده‌اند.

1- Asymmetric DSL
2- Coaxial Cable Networks (CCN)
3- Fibre optics
4- Hybrid fiber Coax



شکل ۱-۷ ♦ شبکه‌ی ترکیبی نوری- هم‌محور

برای ایجاد ارتباط اینترنت از طریق شبکه‌ی HFC نیاز به یک مودم ویژه می‌باشد که مودم کابلی نامیده شده است. از لحاظ عملکرد این مودم شبیه به همتای خود در خطوط DSL می‌باشد، به این‌گونه که شبکه‌ی HFC را به دو جریان دریافت از اینترنت و ارسال به اینترنت تقسیم می‌کند. چگونگی دستیابی از طریق این شبکه همانند DSL به‌صورت غیر متقارن و در دو سرعت مختلف انجام می‌شود.

یکی از مهمترین ویژگی‌های HFC استفاده مشترک از خصوصیت پخش در رسانه‌ی آن می‌باشد. هر بسته‌ی ارسالی از طریق **نقطه‌ی پخش**[1] HFC با عبور از هر یک از شاخه پیوندها به هر کاربر اینترنت رسیده، و هر بسته‌ی گسیل شده از سوی کاربر از طریق هر یک از این شاخه پیوندها، سرانجام به نقطه‌ی پخش تحویل می‌شود. به همین دلیل اگر چندین کاربر به‌صورت همزمان در حال دریافت یک فایل تصویری، بر روی پیوند دریافت باشند، نرخ واقعی هر یک از آنها به مراتب کمتر از سرعت انباشته‌ی همان پیوند است. از طرف دیگر اگر تعداد کاربران فعال اندک باشد و همه‌ی آنها در حال وبگردی باشند، آنگاه سرعت هر یک از کاربران در حالت دریافت، برابر سرعت واقعی پیوند است، زیرا کاربران به‌ندرت به شکل همزمان تقاضای دریافت یک صفحه‌ی وب را از پیوند و نهایتاً اینترنت دارند. از آنجائیکه علاوه بر پیوند دریافت، مسیر ارسال نیز بصورت مشترک مورد استفاده‌ی کاربران قرار می‌گیرد، بنابراین برای جلوگیری از برخورد تقاضاها و نیز مدیریت ارسال آنها، نیاز به پروتکل‌های دستیابی چندگانه به شکل توزیع‌گرا می‌باشد. بدیهی است طرفداران خطوط DSL همواره به این نکته اشاره می‌کنند که DSL یک ارتباط نقطه به نقطه را بین کاربر و ISP فراهم می‌آورد و بنابراین تمامی ظرفیت انتقال یک پیوند DSL به کاربر و ISP اختصاص می‌یابد و اشتراکی در میان نیست. از طرف دیگر طرفداران شبکه‌های کابلی HFC بر این نکته پای می‌فشارند که سرعت شبکه‌های HFC به مراتب بالاتر از خطوط DSL است. در مناطق دور افتاده که امکان استفاده از خطوط DSL و یا

1- Head end

HFC وجود ندارد، از پیوندهای ماهواره‌ای با سرعت بیش از یک مگا بیت بر ثانیه می‌توان جهت ارتباط با اینترنت استفاده نمود.

از جمله‌ی امکانات دیگر ارتباطی با اینترنت **شبکه‌های نوری خانگی**[1] (FTTH) است که سرعتی به مراتب بالاتر از رسانه‌های کابلی فلزی همانند زوج به هم تابیده و یا کابل هم‌محور دارند. برخی از شبکه‌های تلکو در کشورهای مختلف از طریق ایجاد مسیرهای کابل نوری بین مراکز تلفنی (CO) و محل کاربران بر افزایش سرعت دستیابی به اینترنت به‌صورت چشمگیری افزوده‌اند.

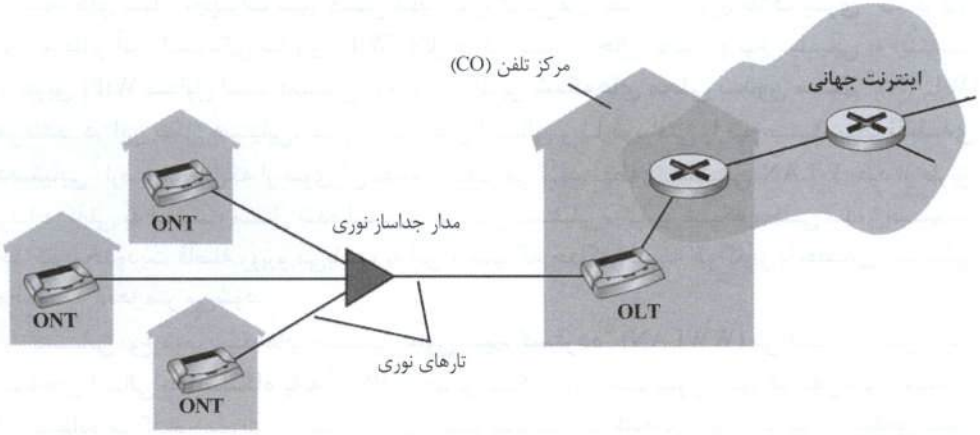
فناوری‌های مختلفی جهت تحقق این مهم تعریف شده‌اند. ساده‌ترین آنها که شبکه‌ی **تار مستقیم**[2] نام دارد از طریق شبکه‌های توزیع نوری که عبارت است از عبور یک تار از CO به سمت هر یک از کاربران، سبب شده که پهنای باند وسیعی در اختیار هر یک از کاربران قرار گیرد. زیرا هر کاربر، تار انحصاری خود را دریافت می‌کند و امتداد آن تا مرکز تلفن (CO) می‌باشد. فناوری دیگر، استفاده‌ی مشترک انبوهی از کاربران از طریق یک تار نوری از سمت مراکز تلفنی (CO) است. در این فناوری هنگامی که کابل عبوری به نزدیک محل کاربران می‌رسد به انشعاب‌های نوری مستقل و ویژه برای هر یک از آنها تبدیل می‌شود. فناوری تقسیم در نزدیکی محل کاربران به انشعاب‌های مستقل نوری، خود به دو شاخه تقسیم می‌گردد.

این دو شاخه عبارتند از: (۱) **شبکه‌های نوری فعال**[3] (AONS) و (۲) **شبکه‌های نوری غیرفعال**[4] (PONS). شبکه‌های فعال نوری از فناوری که اصطلاحاً اترنت راهیابی شده نامیده می‌شود، استفاده می‌کنند. در فصل ۵ موارد مربوط به این فناوری، به گونه‌ی گسترده‌تری تحلیل شده است. عناصر اصلی و چارچوب کلی شبکه‌های PON در شکل ۱-۸ نشان داده شده و همانگونه که در شکل ملاحظه می‌شود، این شبکه‌ها در حقیقت FTTH هستند که از معماری توزیعی PON استفاده نموده‌اند. هر کاربر دارای یک **نقطه‌ی انتهایی شبکه نوری**[5] (ONT) می‌باشد که خود از طریق تار نوری اختصاصی به یک ONT محلی متصل می‌شود. در سمت دیگر این تار **مدار جداساز نوری**[6] قرار گرفته است. وظیفه‌ی این مدار ترکیب و یا تفکیک حداکثر تعداد 100 کاربر بر روی یک تار نوری مشترک می‌باشد. این تار مشترک در سوی دیگر به نقطه‌ی انتهایی خط نوری (OLT) در یک مرکز تلفن تِلکو متصل شده است. مدار OLT تبدیل سیگنال‌های داده از الکتریکی به نوری و برعکس و نیز اتصال به شبکه‌ی اینترنت از طریق مسیریاب تلکو را انجام می‌دهد.

کاربران با استفاده از یک مسیریاب بی‌سیم به ONT نصب شده در محل خود و از طریق آن به مسیریاب تلکو و شبکه‌اینترنت دسترسی پیدا می‌کنند. در معماری PON همه‌ی بسته‌های ارسال شده از OLT در مدار جداساز بازیابی می‌شوند. استفاده از شبکه‌ی FTTH سبب می‌شود که سرعت دستیابی به چندین گیگا بیت در ثانیه برسد. بدیهی است پهنای اختصاص یافته به هر کاربر متفاوت و بستگی به نوع توافق بین ISP و کاربر دارد. اغلب کاربران FTTH، در حال حاضر سرعت دریافت از اینترنت را بین 10 تا 20 مگا بیت بر ثانیه دارند در حالیکه سرعت ارسال به سمت اینترنت از سوی

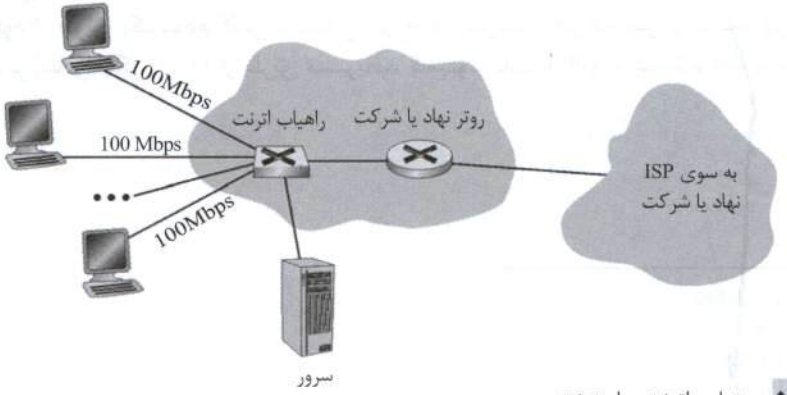
1- Fiber to the home
2- Direct Fiber
3- Active Optical Networks
4- Passive Optical Networks
5- Optical Network Terminator (ONT)
6- Optical Splitter (OS)

همین کاربران بین 2 تا 10 مگا بیت بر ثانیه است. علاوه بر دستیابی به اینترنت، شبکه‌های نوری امکان انتقال و پخش تصاویر مربوط به رسانه‌های تصویری و خدمات تلفنی را نیز میسر می‌سازند.



شکل ۱-۸ ♦ دستیابی به اینترنت از طریق **FTTH**

معمولاً در اماکن اجتماعی از قبیل دانشگاه‌ها و یا شرکت‌های بزرگ تجاری از یک شبکه‌ی محلی جهت برقراری ارتباط سیستم‌های انتهایی با مسیریاب نبش شبکه استفاده می‌شود. اگر چه فناوری‌های مختلفی جهت ایجاد این شبکه‌های محلی تعریف شده است، اما در بین این فناوری‌ها اترنت یکی از رویکردهای بسیار متداول جهت دستیابی امکانات اینگونه اماکن با شبکه‌ی اینترنت می‌باشد. شکل ۱-۹ ساختار کلی و چگونگی دستیابی یک شبکه‌ی اترنت را به شبکه جهانی اینترنت نشان می‌دهد. کاربران اترنت با استفاده از زوج کابل مسی بهم تابیده شده (TP) به یک راهیاب اترنت متصل می‌شوند و سپس از طریق ارتباط راهیاب با مسیریاب این شبکه سرانجام به ISP مورد نظر و اینترنت دسترسی پیدا می‌کنند. سرعت دستیابی کاربران اترنت معمولاً به 100 مگا بیت بر ثانیه می‌رسد، در حالیکه سرعت سرورهای همین شبکه ممکن است بین یک تا 10 گیگا بیت بر ثانیه باشد.



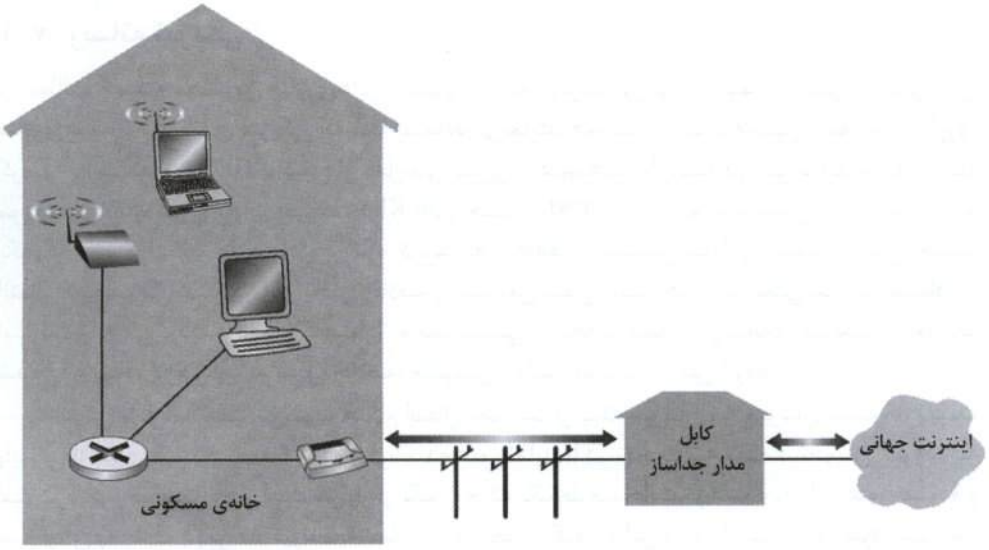
شکل ۱-۹♦ دستیابی اترنت و اینترنت

در حال حاضر رویکرد دستیابی به اینترنت از طریق **شبکه‌های بدون سیم**[1] یا WiFi به گونه‌ای چشمگیر رو به گسترش می‌باشد. با ورود تجهیزات مختلف قابل حمل به‌وسیله‌ی کاربران همانند کامپیوترهای سیار، تجهیزات سیار دستی نظیر انواع گوشی‌های همراه، آیفون، **بلاک بری**[2] یا گوگل فون و نظایر آنها گستردگی فناوری WiFi قابل درک است. در حال حاضر دو نوع دستیابی به اینترنت از طریق WiFi متداول است. دستیابی نوع اول از طریق **شبکه‌های محلی بدون سیم**[3] (WLAN) می‌باشد. در این شکل دستیابی، کاربران بسته‌های ارسالی و یا دریافتی را به سمت یک **نقطه‌ی دستیابی**[4] ارسال و یا آنکه از سوی آن نقطه دریافت می‌کنند. نقطه‌ی دستیابی WLAN خود از طریق ارتباط کابلی به اینترنت متصل شده است. در این نوع دستیابی، کاربر شبکه محلی بدون سیم با حداکثر محدودیت فاصله روبرو می‌باشد، به این ترتیب که حداکثر فاصله هر کاربر با نقطه‌ی دستیابی محدود به ده‌ها متر می‌شود.

دستیابی نوع دوم، شبکه‌های دستیابی **بدون سیم گسترده**[5] (WWLAN) می‌باشند. در این روش بسته‌های ارسالی به **ایستگاه پایه**[6] (BS) از طریق شبکه زیر ساخت بدون سیم که تلفن‌های همراه از آن استفاده می‌کنند، صورت می‌پذیرد. بدیهی است مدیریت ایستگاه‌های پایه از طریق شبکه‌ی سیار تلفنی انجام می‌گیرد و فاصله‌ی کاربران از این ایستگاه‌ها در چنین شرایطی می‌تواند به ده‌ها کیلومتر نسبت به نوع اول افزایش یابد. شبکه‌های دستیابی به اینترنت WLAN بر اساس فناوری IEEE802-11 پایه‌ریزی شده‌اند که همان WiFi می‌باشد. فضای تحت پوشش WiFi از کوچکترین مجموعه‌ها تا سطح شهرها، کشورها و حتی جهان نیز گسترده شده است. این شبکه از طریق فراهم‌آوری نرخ اشتراک انتقال، حداکثر 54 مگا بیت بر ثانیه تقریباً امکان دستیابی همه‌ی شهروندان جهان را با یکدیگر فراهم آورده است.

در فصل ۶ مجموعه‌ی پروتکل IEEE802-11 به‌صورت گسترده‌تر بررسی شده است. در حال حاضر بسیاری از کاربران مسکونی با ترکیب دستیابی پهنای باند گسترده خود، همانند شبکه‌های کابلی هم‌محور و یا خطوط DSL، با فناوری نسبتاً ارزان WLAN، از شبکه‌های خانگی قدرتمند جهت دستیابی به اینترنت بهره گرفته‌اند. شکل ۱-۱۰ نمونه‌ای از شبکه‌های خانگی را نشان می‌دهد. در این شکل، شبکه شامل چندین کامپیوتر با **امکانات گردشگری**[7] (جابه جایی) و یک کامپیوتر با اتصال کابلی است. یک نقطه‌ی دستیابی بدون سیم با این کامپیوترها ارتباط برقرار می‌کند. سرانجام خود از طریق یک مودم کابلی دستیابی سریع به اینترنت را فراهم می‌آورد. بدیهی است در این شبکه، بر اساس شکل ۱-۱۰ از طریق مسیریاب، کامپیوتر ثابت (PC) به ایستگاه پایه متصل شده است.

1- Wireless Fidelity
2- Blackberry
3- Wireless LAN
4- Access Point
5- Wide Area Wireless LAN
6- Base Station
7- Roaming



شکل ۱-۱۰ ♦ پیاده‌سازی شبکه بدون سیم خانگی

هنگامی که دستیابی اینترنت از طریق WLAN صورت می‌پذیرد، فاصله تا نقطه‌ی دستیابی بایستی در محدوده چند صد متر رعایت گردد. این محدودیت، دسترسی را در اماکن کوچک و محدود امکان‌پذیر می‌نماید، اما در نقاط دیگری از شهر و یا به هنگام آمد و شد بین نقاط مختلف در یک شهر یا کشور WLAN نمی‌تواند شرایط دستیابی به اینترنت را فراهم آورد. برای دستیابی به اینترنت در محدوده‌های بزرگتر نیاز به استفاده از شبکه‌های زیر ساخت سیار و به عبارت دیگر **سلولی**[1] می‌باشد که ارتباط و دستیابی را فراتر از ده‌ها کیلومتر میسر می‌سازند. در حال حاضر شرکت‌های مخابراتی در گوشه و کنار جهان سرمایه‌گذاری وسیعی را بر روی شبکه‌های بدون سیم **نسل سوم**[2] (3G) انجام داده‌اند. این شبکه‌ها قادرند، دستیابی بدون سیم، را از طریق فناوری راهیابی بسته با سرعت فراتر از 1Mbps در اختیار کاربران قرار دهند. در حال حاضر میلیون‌ها کاربر از این شبکه‌ها جهت ارسال و دریافت اطلاعاتی از قبیل پست الکترونیکی، وبگردی و دستیابی به انواع مختلف اطلاعات استفاده می‌کنند.

بدیهی است همواره با روند تغییرات گسترده در فناوری‌های انجام شده، می‌توان انتظار داشت رویکردهای تحول‌گرای خیره کننده نیز پیشنهاد گردد. یکی از این تحولات، پروتکل فناوری WiMAX[3] یا IEEE802.16 می‌باشد که در مقابل پروتکل IEEE802-11 قرار گرفته است. این فناوری از آن جهت تحول‌گرا به‌شمار می‌آید که در مقایسه با WiFi مستقل از شبکه‌های ارتباطی سیار یا سلولی عمل می‌نماید و سرعت دسترسی به اینترنت را نیز بین 5 تا 10 مگا بیت بر ثانیه در مقابل یک مگا بیت بر ثانیه 3G ارتقاء می‌دهد. فصل ۶ به بررسی مفصّل فناوری‌های WiFi، WiMAX و 3G می‌پردازد.

1- Cellular Networks
2- Third Generation Wireless Networks
3- Worldwide Interoperability for Microwave Access

۱-۷ رسانه فیزیکی

در مطالب گذشته مهمترین فناوری‌های دستیابی شبکه اینترنت بررسی گردید. در طی بررسی این فناوری‌ها به نوع رسانه‌ی فیزیکی که آنها استفاده می‌نمایند هم اشاره شد. به‌عنوان نمونه، یادآوری گردید که شبکه‌های HFC ترکیبی از کابل‌های نوری و هم‌محور را استفاده می‌کنند و یا آن که مودم‌های شماره‌گیری دارای سرعت 56 Kbps و خطوط DSL از دو رشته سیم مسی تابیده شده به یکدیگر بهره گرفته‌اند. علاوه بر این اشاره گردید که شبکه‌های دستیابی سیار از طیف رادیویی جهت انتقال استفاده می‌کنند. در این بخش خلاصه‌ای پیرامون انواع رسانه‌های فیزیکی مورد استفاده‌ی اینترنت ارائه می‌گردد. برای آن که بتوان تعریف درستی از آنچه به‌عنوان «رسانه‌ی فیزیکی» معروف شده ارائه نمود، لازم است به شکل خلاصه، چگونگی حیات یک بیت بررسی گردد.

با اندکی تامل ملاحظه می‌شود، هنگام انتقال یک بیت از یک میزبان به میزبان دیگر، آن بیت بایستی از میان یک مجموعه گیرنده و فرستنده عبور نماید تا در نهایت به مقصد تحویل گردد. به عبارت دیگر یک بیت ناچار است هر بار از یک زوج که یک طرف، حالت فرستنده را دارد عبور نموده و پس از آن به طرف دیگر که گیرنده به‌شمار می‌آید تحویل گردد و این روند تا انتها دنبال شود. بنابراین هنگام عبور بیت از هر زوج، حرکت آن به‌صورت انتشار امواج الکترو مغناطیسی و یا امواج نوری از درون یک رسانه‌ی فیزیکی است.

رسانه‌ی فیزیکی می‌تواند شامل انواع قالب‌ها و سازه‌ها باشد و بنابراین برای هر زوج فرستنده و گیرنده در طول مسیر لازم نیست از لحاظ نوع یکسان باشد. نمونه هایی از رسانه‌ی فیزیکی عبارتند از: زوج رشته مسی بهم تابیده شده (TP)، کابل هم‌محور، تارهای نوری چند شعاع نوری، **طیف رادیویی سطحی**[1] و **طیف رادیویی ماهواره‌ای**[2]. از لحاظ کلی رسانه‌ی فیزیکی در قالب دو گونه‌ی **هدایت شده** و رسانه‌ی **هدایت نشده** تقسیم می‌شود. در رسانه‌ی هدایت شده امواج مسیر هدایت شده‌ای را در راستای یک رسانه‌ی هموار نظیر: تار نوری، زوج بهم تابیده شده و یا کابل مسی هم‌محور طی می‌نمایند. در حالیکه با رسانه‌ی هدایت نشده همین امواج در جو کره زمین و یا فضای خارج از آن ادامه‌ی مسیر می‌دهند. نمونه‌هایی از این نوع، شبکه محلی بدون سیم و یا کانال‌های دیجیتال ماهواره‌ای است.

بدون شک لازم است قبل از پرداختن به مشخصات هر یک از انواع این رسانه‌ها، تحلیل کلی از هزینه‌های مرتبط با هر یک از دو نوع رسانه به‌عمل آید. به طور کلی هزینه‌ی واقعی پیوندهای فیزیکی مثل، کابل مسی و یا نوری در اغلب موارد نسبت به سایر هزینه‌های شبکه کمتر است. این هزینه به‌ویژه در ارتباط با پرداخت کار نیروی انسانی دارای اهمیت بیشتری می‌باشد. به همین منظور عاقلانه است هنگام انجام عملیات ساختمانی به‌ویژه هنگام ایجاد یک بنای جدید، انواع مختلف رسانه‌های فیزیکی از قبیل TP، هم‌محور و یا نوری نصب گردد، تا در آینده نیازی به نصب آنها نباشد، حتی اگر هنگام ساخت بنای جدید تنها استفاده از یکی از آنها ممکن باشد، زیرا همواره این احتمال وجود دارد در آینده به نوع دیگر رسانه نیز نیاز باشد.

در میان رسانه‌های فیزیکی کابلی شاید یکی از کم‌هزینه‌ترین آنها را بتوان زوج بهم تابیده شده یا همان TP دانست. مدت زمانی بیش از یکصد سال است که در شبکه‌های تلفنی از آن استفاده می‌شود.

1- Terresterial radio spectrum
2- Satellite radio spectrum

آنچه مسلّم است بیش از 99% ارتباط‌های کابلی، از گوشی‌های تلفن گرفته تا مراکز تلفنی محلی از این نوع رسانه جهت ارتباط بین خود استفاده می‌کنند.

هر زوج TP شامل دو رشته سیم عایق شده، هر یک به ضخامت یک میلی‌متر و در قالب یک مجموعه‌ی چند زوجی در درون یک روکش از جنس PVC سازماندهی شده‌اند. علت تابیده شدن این زوج‌ها به یکدیگر جهت کاهش تداخل الکتریکی از سوی زوج‌های مشابه مجاور می‌باشد. هر زوج TP تشکیل یک پیوند ارتباطی را می‌دهد. معمولاً در شبکه‌های کامپیوتری محلی (LAN) از **زوج بهم تابیده شده‌ی بدون پوشش**[1] (UTP) استفاده می‌شود. سرعت انتقال شبکه‌هایی که از این نوع رسانه جهت ارتباط عناصر خود استفاده می‌کنند در حال حاضر در محدوده 10 Mbps تا 1 Gbps است. بدیهی است سرعت انتقال داده، وابسته به ضخامت رشته سیم استفاده شده و نیز فاصله‌ی بین فرستنده و گیرنده دارد.

یادآوری این نکته ضروری است که هنگام ظهور فناوری تارهای نوری در دهه‌ی 1980 میلادی، بسیاری بر این باور بودند که رسانه‌ی TP به علت سرعت پائین انتقال خود در مقایسه با رسانه‌ی جدید، بزودی از صحنه ناپدید خواهد شد و تارهای نوری جایگزین آن می‌شود، درحالیکه چنین نشد. فناوری نوین TP همانند **UTP گروه 5**[2] (Cat5UTP) قابلیت دستیابی به نرخ انتقال 1 Gbps در فاصله یکصد متر را دارد. در حال حاضر رسانه‌ی TP به‌عنوان رسانه‌ی تأثیر گذار در تشکیل شبکه‌های محلی (LAN) با سرعت بسیار بالا می‌باشد. قبلاً اشاره گردید رسانه‌ی TP جهت دستیابی اینترنت خانگی به‌صورت گسترده مورد استفاده قرار گرفته است. هر دو فناوری مودم 56 Kbps و نیز خطوط DSL از این رسانه جهت انتقال پیام خود استفاده می‌کنند و در اغلب موارد کاربران اینترنت خانگی می‌توانند با بهره‌گیری از خطوط DSL که از این نوع رسانه استفاده می‌کند به سرعت دستیابی قابل ملاحظه‌ای معادل 6 Mbps را هنگامی که در فاصله‌ی نه چندان دوری از ISP قرار گرفته‌اند، تجربه نمایند.

کابل هم‌محور به‌عنوان یک رسانه‌ی ارتباطی شبکه همانند TP شامل دو رسانای مسی است، اما این دو رسانا به جای آنکه موازی یکدیگر باشند، هم‌محور یکدیگرند. با این ساختار و استفاده از عایق‌بندی ویژه و نیز پوشش در برابر اختلال‌های/امواج الکتریکی و مغناطیسی، این رسانه قادر به تأمین سرعت انتقال بالا می‌باشد.

از این رسانه بیشتر در شبکه‌های کابلی تصویری که قبلاً به آن اشاره گردید، استفاده می‌شود. دستیابی به اینترنت با سرعت 1Mbps و یا بیشتر از طریق این شبکه‌ها برای کاربران خانگی امکان‌پذیر می‌باشد. با استفاده از این رسانه در شبکه‌های کابلی و دستیابی اینترنت، بخش فرستنده‌ی شبکه با انتقال سیگنال دیجیتال حامل اطلاعات، به یک پهنای باند ویژه و سپس ارسال سیگنال آنالوگ نتیجه شده در این پهنای باند، از طریق فرستنده به یک یا بیش از یک گیرنده، عملاً دستیابی را بین عناصر دو سوی شبکه امکان‌پذیر می‌سازد.

معمولاً از کابل هم‌محور به‌عنوان رسانه مشترک هدایت شده استفاده می‌شود. به عبارت دیگر تعدادی سیستم انتهایی (میزبان) دارای ارتباط مستقیم با کابل هم‌محور گردیده و در نتیجه هر یک از میزبان‌ها قادر به دریافت آنچه دیگر میزبان‌ها ارسال نموده‌اند، می‌شوند.

در مقایسه با دو رسانه‌ی فلزی که با استفاده از امواج الکتریکی به ارسال داده می‌پرداختند، یک تار

1- Unshielded Twisted Pairs
2- Category 5 UTP

نوری این مهم را از طریق امواج نوری انجام می‌دهد. یک تار نوری رسانه‌ای بسیار نازک و انعطاف‌پذیر است که هر یک بیت از داده را در قالب یک پالس نوری انتقال می‌دهد. یک تار نوری نازک به تنهایی قادر است سرعت بسیار بالایی را در حد ده‌ها و یا صدها گیگا بیت در ثانیه امکان‌پذیر سازد. این رسانه در برابر تداخل الکترومغناطیسی مقاوم بوده و از لحاظ تضعیف سیگنال بسیار پائین و در طول کابلی 100 کیلومتر قرار دارد. سرقت اطلاعات از طریق این رسانه بسیار دشوار و تقریباً غیر ممکن است. به لحاظ این ویژگی‌های چشمگیر، رسانه‌ی نوری برای انتقال داده‌های شبکه در فواصل دراز و پیوندهای ارتباطی بین قاره‌ای کاربرد گسترده‌ای را امروز به خود اختصاص داده است.

بسیاری از رسانه‌های ارتباطی راه دور شبکه‌های مخابراتی در حال حاضر از نوع تارهای نوری می‌باشند. علاوه بر همه‌ی این موارد تارهای نوری در حال حاضر شبکه زیر ساخت اینترنت را تشکیل می‌دهند. به علت هزینه‌ی بالای تجهیزات این رسانه از قبیل فرستنده‌ها، گیرنده‌ها و سایر لوازم، استفاده از این رسانه در شبکه‌های محدود از لحاظ وسعت و فاصله‌ی فیزیکی همانند شبکه‌های محلی LAN و یا کاربرد آن برای کاربران خانگی، مقرون به صرفه نمی‌باشد.

سرعت پیوندهای استاندارد **حامل نوری**[1] (OC) دارای محدوده‌ی مابین 51.8Mbps تا 39.8Gbps می‌باشند. این مشخصات اصطلاحاً OC-n نامیده می‌شوند (لازم به یادآوری است OC-n مشخصات آمریکای شمالی و معادل اروپایی آن STC-n[2] می‌باشد). در این بیان سرعت هر پیوند برابر n × 51.8 Mbps است. استانداردهای استفاده شده در حال حاضر شامل OC-1، OC-3، OC-12، OC-24، OC-48، OC-96، OC-192 و OC-768 می‌گردند. بدیهی است معادل همین ضریب‌ها برای استاندارد اروپایی آن نیز به‌صورت STC-n وجود دارد.

در مقایسه با رسانه‌های هدایت شده، کانال‌های رادیویی (فضای آزاد) به‌عنوان رسانه‌ی هدایت نشده، سیگنال‌های حامل اطلاعات را در طیف امواج الکترومغناطیس حمل می‌نماید. این کانال‌ها رسانه‌ی جذابی برای دست اندرکاران شبکه به‌شمار می‌آیند زیرا نیازمند نصب ارتباط‌های فیزیکی همانند کابل نمی‌باشند، می‌توانند از دیوارها و موانع نظیر آن نفوذ نمایند. امکان ارتباط برای کاربران سیار از طریق این رسانه امکان‌پذیر است و از همه مهمتر سیگنال داده را می‌تواند تا فواصل دوردست انتقال دهد. یک کانال رادیویی به گونه‌ی چشمگیری وابستگی به محیط انتشار سیگنال و نیز فاصله‌ی پیمایش سیگنال دارد. شرایط محیطی بر میزان تضعیف مسیر تأثیرگذار است و نیز کاهش قدرت سیگنال، عبور آن را از یک مسیر و موانع ممکن در اطراف و یا در بین آن مسیر مشخص می‌کند. کاهش قدرت سیگنال به هنگام عبور اصطلاحاً **افت سایه**[3] نامیده می‌شود. علاوه بر افت سایه، عوامل تضعیف گر دیگری نیز وجود دارند که عبارتند از: **افت مسیر چندگانه**[4] که در اثر انعکاس سیگنال در برخورد با موانع روی می‌دهد و تداخل امواج در اثر حضور سایر عوامل انتقال و نیز سیگنال‌های الکترومغناطیسی دیگر. به‌طور کلی این رسانه طیف رادیویی سطحی نامیده می‌شود. رسانه‌ی رادیویی سطحی خود به دو گروه عمده‌ی دیگر تقسیم می‌شود، این گروه‌ها عبارتند از: دسته‌ی اول که دامنه‌ی عملکرد آنها محلی است و به عبارت دیگر از ده یا چند صد متر تجاوز نمی‌نماید.

گروه دوم آنهایی هستند که دامنه‌ی عملکردشان بسیار وسیع است و کیلومترها وسعت را پوشش

1- Optical Carrier
2- Standard Transmission Code
3- Shadow Fading
4- Multipath Fading

می‌دهد. شبکه‌های محلی رادیویی از فناوری گروه اول و شبکه‌های تلفنی سیار از فناوری گروه دوم استفاده می‌کنند.

کانال‌های رادیویی ماهواره‌ای بخش دیگری از رسانه‌ی هدایت نشده را تشکیل می‌دهند. یک پیوند ارتباط ماهواره‌ای شامل دو یا بیش از دو فرستنده و یا گیرنده‌ی زمینی است که اصطلاحاً به آنها **ایستگاه‌های زمینی**[1] گفته می‌شود. یک ماهواره معمولاً انتقال سیگنال را از طریق یک پهنای باند دریافت و پس از بازیابی مجدد با استفاده از **مدار تکرار کننده**[2] خود آن را جهت انتقال مجدد به سوی دیگر، بر روی پهنای باند دیگری قرار می‌دهد.

از دو نوع ماهواره جهت کانال‌های رادیویی ماهواره‌ای استفاده می‌شود: ماهواره‌های ثابت از لحاظ مکان[3] (GOS) و ماهواره‌های سیار در ارتفاع پائین[4] (LEO).

ماهواره‌های ثابت در بالای یک نقطه از زمین به شکل ثابت باقی می‌مانند. این حالت ثابت از لحاظ مکان از طریق قرار گرفتن ماهواره در یک مدار در فاصله‌ی 36000 کیلومتر از سطح زمین امکان‌پذیر است. این فاصله‌ی طولانی از ایستگاه زمینی، ارسال سیگنال از ایستگاه زمینی به ماهواره و سپس بازگشت آن به نقطه‌ای دیگر از زمین را با یک تاخیر انتشار قابل توجه معادل 280 میلی ثانیه رو به رو می‌سازد. با وجود این تأخیر انتشار، پیوندهای ماهواره‌ای قادرند سرعت معادل صدها مگابیت بر ثانیه را اختیار نمایند. از این پیوندها در محدوده‌هایی استفاده می‌شود که امکان دسترسی به اینترنت از طریق رسانه‌های کابلی میسر نمی‌باشد. ماهواره‌های LEO در فاصله‌ی نزدیکتری از زمین در مدار قرار می‌گیرند و از لحاظ مکانی در یک نقطه نیز ثابت نمی‌باشند. چگونگی گردش آنها در اطراف کره زمین همانند گردش کره‌ی ماه به دور زمین می‌باشد. آنها می‌توانند علاوه بر ایستگاه‌های زمینی با یکدیگر نیز ارتباط برقرار نمایند. برای آنکه بتوان پوشش دائمی در یک محدوده از کره‌ی زمین را از طریق این ماهواره‌ها امکان‌پذیر نمود، بایستی تعداد قابل توجهی از آنها را در مدار قرار داد. در حال حاضر صدها ماهواره‌ی LEO در مدار زمین قرار دارند و فناوری آنها در حال رشد می‌باشد، اما از این ماهواره‌ها جهت دستیابی اینترنت استفاده نمی‌شود و ممکن است در آینده این امر محقق گردد.

۱-۸ هسته‌ی مرکزی شبکه[5]

پس از بررسی عناصر بیرونی شبکه که اصطلاحاً به **نبش شبکه** معروف شده‌اند، اکنون نوبت تحلیل هسته‌ی مرکزی شبکه است که مجموعه‌ای از راهیاب‌های بسته و پیوندهایی که این عناصر را به یکدیگر مرتبط کرده‌اند را شامل می‌شود.

دو رویکرد اساسی جابجایی داده‌ها از درون شبکه‌ای از پیوندها و راهیاب‌ها عبارتند از: (۱) راهیابی مداری و (۲) راهیابی بسته. در شبکه‌های راهیابی مداری منابع لازم در طول یک مسیر (بافرها و سرعت انتقال پیوندها) ارتباط بین میزبان‌ها، در تمام مدت انجام ارتباط، حالت اختصاصی و تمامیت‌گر را بین سیستم‌های انتهایی نگهداری می‌کنند. در شبکه‌های راهیابی بسته این حالت وجود ندارد و بنابراین یک پیام در قالب مجموعه‌ای از داده‌ها از منابع، تقاضای استفاده‌ی بی‌درنگ می‌کند و در نتیجه ممکن است

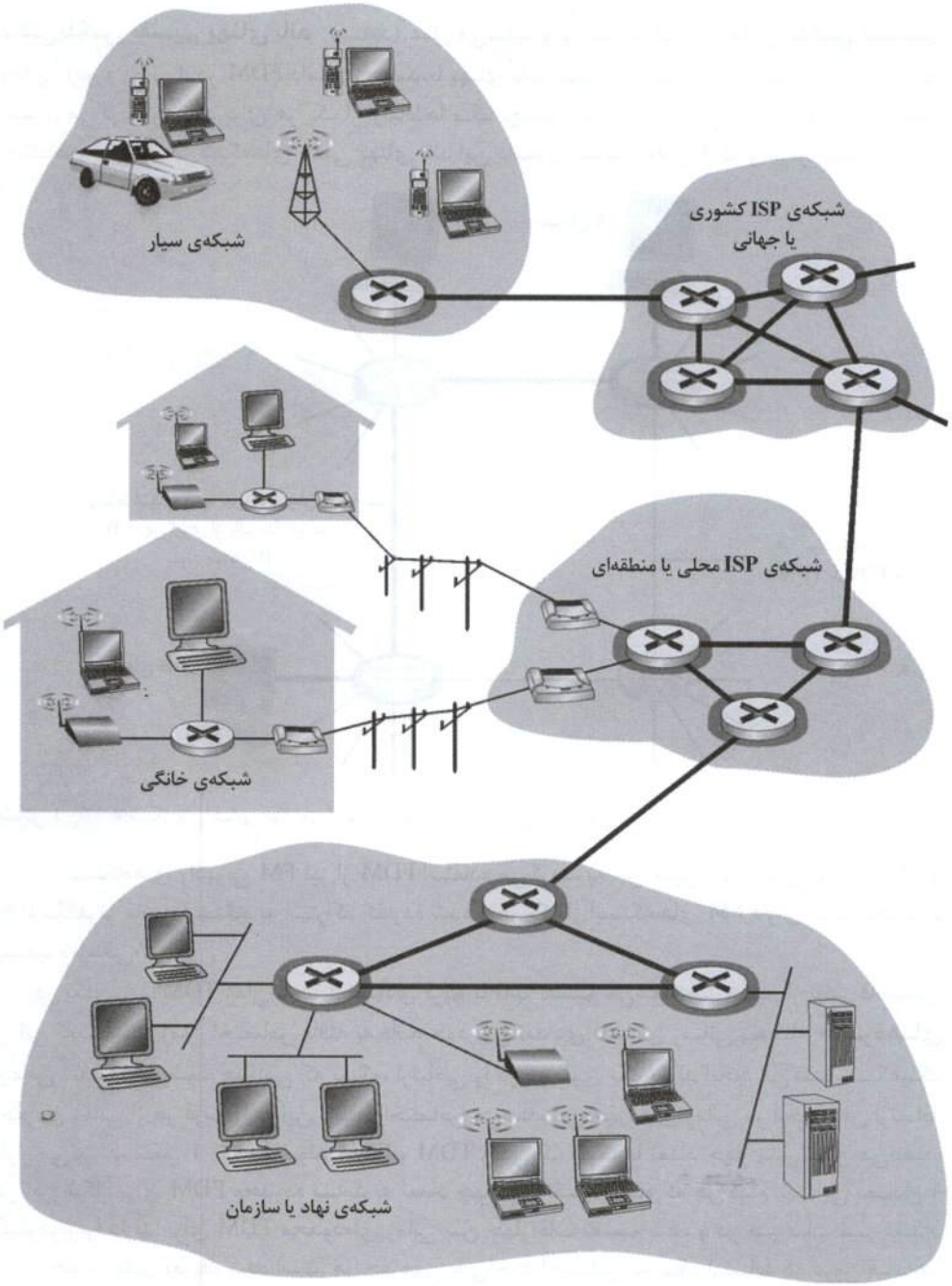
1- Ground Stations
2- Repeaters
3- Geostationary Satellites
4- Low earth orbiting satellites
5- The Network core

برای دستیابی به یک ارتباط مجبور به انتظار و قرار گرفتن در صف شود. شبکه‌های تلفنی در سراسر دنیا، نمونه‌ای از شبکه‌های راهیابی مدار است. در این شبکه‌ها قبل از آنکه دو میزبان به مبادله‌ی اطلاعات و یا پیام‌های خود از هر نوعی بپردازند، شبکه بایستی ارتباطی را بین فرستنده و گیرنده ایجاد نماید. این ارتباط مسیری را شکل می‌دهد که از مجموعه‌ای از راهیاب‌ها تشکیل شده‌اند، که وظیفه‌ی نگهداری و کنترل وضعیت ارتباط را بین دو میزبان برعهده دارند. در واژگان شبکه‌ی تلفنی این ارتباط **مدار** نامیده می‌شود. بدیهی است هنگامی که شبکه، مداری را ایجاد می‌نماید، سرعت انتقال ثابت و تغییر ناپذیری را نیز در پیوندهای شبکه در طول انجام ارتباط تعریف می‌کند. از آنجایی که پهنای باند مشخص شده‌ای برای ارتباط فرستنده و گیرنده تعریف می‌شود، فرستنده داده‌های خود را همواره با سرعت ثابتی به سوی گیرنده گسیل می‌نماید.

اینترنت در حال حاضر شبکه‌ای است الزاماً از نوع راهیابی بسته. در این شبکه بسته از طریق مجموعه‌ای از پیوندهای ارتباطی انتقال می‌یابد. اما در روش راهیابی بسته، پیام بدون آنکه پهنای باند ویژه‌ای به آن اختصاص یابد به درون شبکه گسیل می‌شود. بنابراین اگر یکی از پیوندها دچار تراکم ارسال و یا دریافت باشد و این وضعیت به دلیل درخواست بسته‌های دیگر جهت گسیل شدن از آن پیوند باشد، بسته‌ی مورد نظر بایستی در یک بافر در سمت پیوند انتقال فرستنده در نوبت قرار گیرد تا در زمان مناسب انتقال یابد. نتیجه این انتظار در قالب **زمان تأخیر بسته** در شبکه تعریف می‌شود. شبکه‌ی اینترنت از این طریق بهترین تلاش خود را می‌کند که در زمان مناسب بسته‌ها را تحویل گیرنده نماید اما این مهم را ضمانت نمی‌کند. یادآوری این نکته ضروری است که همه‌ی شبکه‌های ارتباطی به تنهایی نمی‌توانند در قالب هر یک از موارد بالا بیان گردند، بلکه در اغلب موارد به‌صورت ترکیبی از این دو می‌باشند.

از آنجائیکه موضوع این کتاب شبکه‌های کامپیوتری، اینترنت و راهیابی بسته است و به نوعی دارای ارتباط مستقیم با شبکه‌های راهیابی مداری نظیر شبکه تلفنی نمی‌باشد، در هر حال بررسی علل این عدم ارتباط می‌تواند با بررسی شبکه‌های راهیابی مداری به گونه‌ای سودمند تلقی گردد. شکل ۱ - ۱۲ یک شبکه‌ی راهیابی مداری را نشان می‌دهد. در این شبکه، چهار راهیاب مداری از طریق تعداد چهار پیوند به یکدیگر متصل شده‌اند. هر یک از این پیوندها دارای تعداد n مدار است و بنابراین هر پیوند می‌تواند n ارتباط هم‌زمان را پشتیبانی نماید. میزبان‌ها هر یک مستقیماً به یکی از راهیاب‌ها متصل شده‌اند. هنگامی‌که دو میزبان مایل به ارتباط هستند، شبکه‌یک مسیر انتها به انتها را بین آن دو ایجاد می‌نماید. بدیهی است حالت **ارتباط جمعی**[1] نیز می‌تواند وجود داشته باشد که در این صورت تعداد میزبان‌ها بیش از دو خواهد بود، اما به دلیل ساده نگاه داشتن موضوع، در این بحث به دو میزبان اکتفا شده است. بنابراین برای آنکه میزبان A پیام‌های خود را به میزبان B ارسال نماید، شبکه نخست لازم است یک مدار بر روی هر یک از دو پیوند، تعریف و ذخیره کند. از آنجایی که هر پیوند دارای n مدار است، پیوند استفاده شده در ارتباط مورد نظر، تنها مقداری معادل $\frac{1}{n}$ از همه‌ی ظرفیت پهنای باند پیوند را در طول برقراری ارتباط به خود اختصاص می‌دهد.

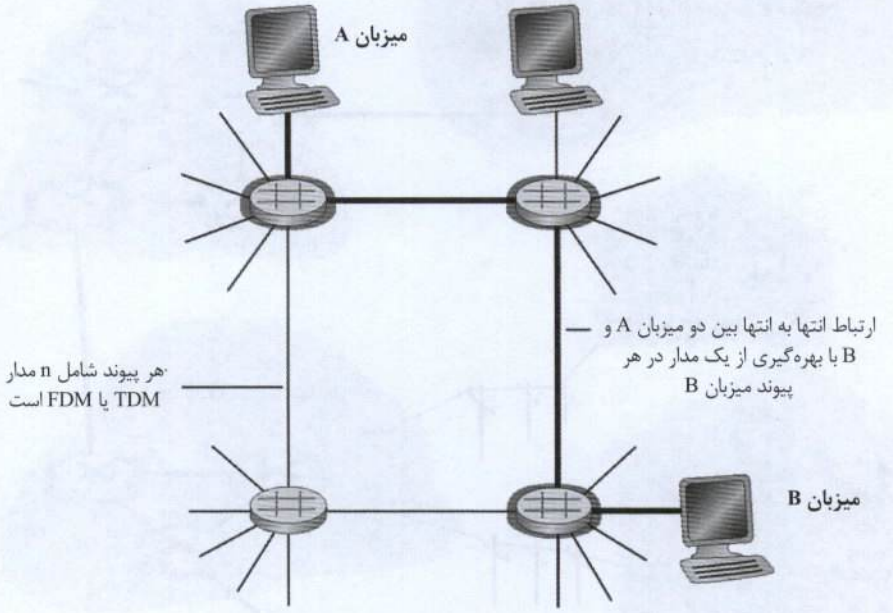
1- Conference Call



شکل ۱- ۱۱ ♦ هسته‌ی شبکه

به همین دلیل در هر پیوند، مداری پیاده‌سازی شده است که برای تحقق این مهم یا به‌صورت

مالتی‌پلکس تقسیم پهنای باند[1] (بسامد) عمل می‌نماید و یا آنکه به گونه‌ی **مالتی‌پلکس تقسیم زمان**[2] (دوره زمانی). در FDM دامنه‌ی بسامد یا پهنای باند پیوند در میان ارتباط‌های ایجاد شده تقسیم می‌گردد و پیوند برای هر یک از ارتباط‌ها یک پهنای باند ویژه را در طول برقراری ارتباط اختصاص می‌دهد. در شبکه‌های تلفنی پهنای باند این دامنه‌ی بسامد معادل 4 کیلو هرتز است.



شکل ۱- ۱۲ ♦ شبکه‌ی راهیابی مداری با تعداد چهار راهیاب و چهار پیوند

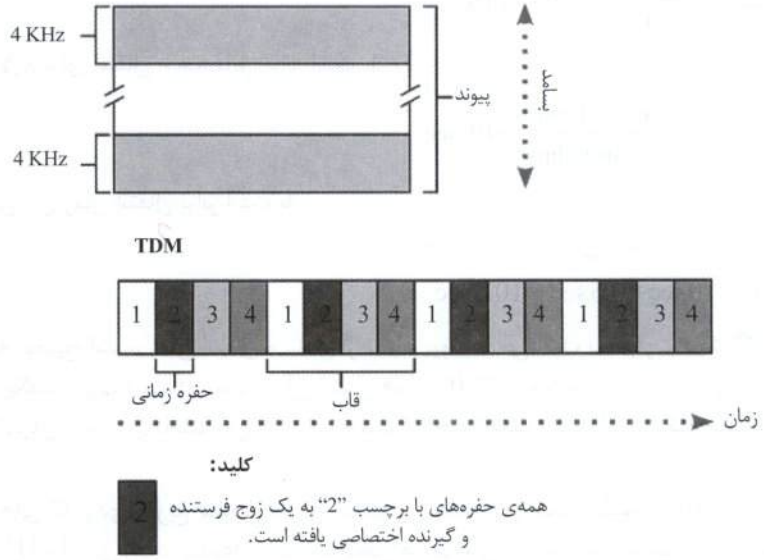
در ایستگاه‌های رادیویی FM نیز از FDM استفاده می‌گردد، به این مفهوم که پهنای باند بین 88 تا 108 مگاهرتز مابین ایستگاه به اشتراک گذارده شده و هر یک از ایستگاه‌های FM دارای یک محدوده بسامد ویژه می‌شود.

در یک پیوند TDM زمان در بین تعدادی فریم یا قاب تقسیم می‌شود. هر فریم از نظر فاصله‌ی زمانی ثابت است. زمان اختصاص یافته به قاب، خود بین تعدادی فاصله‌ی زمانی به نام **حفره‌های زمانی**[3] تقسیم می‌شود. هنگامی که شبکه، ارتباطی را در دو سوی یک پیوند ایجاد می‌کند، شبکه‌یک حفره‌ی زمانی از هر فریم را به این ارتباط اختصاص می‌دهد. این حفره‌های زمانی در انحصار آن ارتباط قرار می‌گیرند. شکل ۱-۱۳ دو پیوند FDM و TDM را در یک شبکه با تعداد چهار مدار نشان می‌دهد. در این شکل برای FDM محدوده بسامد به تعداد چهار باند تقسیم شده که هر کدام پهنایی معادل 4 کیلوهرتز را دارند. برای TDM محدوده‌ای زمانی بین چهار قاب تقسیم شده و در هر قاب نیز تعداد چهار حفره زمانی تعریف شده است. هر حفره‌ی زمانی خود اختصاص به یک ارتباط در درون قاب‌ها دارد. بدیهی است حفره‌ی اختصاص یافته به یک مدار در یک قاب غالباً در بقیه‌قاب‌ها نیز همانند است.

1- Frequency Division Multiplexing
2- Time Division Multiplexing
3- Time Slots

به عبارت دیگر اگر حفره‌ی یک در قاب نخست برای استفاده‌ی مدار شماره 3 تعریف شده، حفره‌های یک در دیگر قاب‌ها نیز به همین مدار اختصاص می‌یابد و این روند به‌صورت چرخشی ادامه پیدا می‌کند.

بنابراین در یک پیوند TDM نرخ انتقال یک مدار برابر است با سرعت فریم که در تعداد بیت‌های موجود در یک حفره‌ی زمانی ضرب می‌گردد. به‌عنوان مثال اگر تعداد قاب‌های یک پیوند در هر ثانیه 8000 و هر حفره‌ی زمانی حامل 8 بیت داده باشد، آن‌گاه سرعت انتقال مدار برابر 64 Kbps خواهد شد.



شکل ۱- ۱۳ ♦ در FDM به هر مدار بخشی از پهنای باند اختصاص می‌یابد. در TDM هر مدار از همه‌ی پهنای باند در یک فاصله‌ی زمانی کوتاه استفاده می‌کند.

علاقه‌مندان به روش راهیابی بسته همواره این استدلال را مطرح می‌کنند که روش راهیابی مداری سبب می‌شود در فواصل سکوت مدار، بخش وسیعی از پهنای باند که می‌تواند مورد استفاده قرار گیرد بیهوده تلف می‌شود. به‌عنوان مثال هنگامی که شخص در یک مکالمه‌ی تلفنی در حال انجام سکوت می‌کند، منابع شبکه که عبارت از پهنای باندها و یا حفره‌های زمانی هستند در این حالت بدون استفاده و بیکار می‌باشند در صورتی که می‌توانند در همین فاصله‌ی زمانی مورد استفاده‌ی مشترک دیگر کاربران قرار گیرند. علاوه بر موارد اشاره شده، این نکته را نیز می‌توان افزود که در روش راهیابی مداری برقراری ارتباط و اختصاص پهنای باند به هر ارتباط پیچیدگی‌هایی را نیز از لحاظ کنترل راهیاب‌های طول مسیر در پی دارد. سرانجام برای درک بهتر از نکات اشاره شده به یک مثال عملی در زیر پرداخته می‌شود.

مثال ۱

حساب کنید زمان لازم برای انتقال یک فایل به حجم 640 کیلوبیت بین دو میزبان A و B با توجه به شرایط زیر از طریق یک شبکه‌ی راهیابی مداری چقدر است؟ شرایط عبارتند از:

الف) همه‌ی پیوندها از نوع TDM با تعداد 30 حفره‌ی زمانی و سرعت انتقال معادل 1920 کیلوبیت بر ثانیه هستند.

ب) زمان لازم برای برقراری ارتباط بین دو میزبان A و B قبل از آغاز انتقال فایل برابر 500msec باشد.

راه حل:

اگر فرض شود هر کیلو بیت معادل 1000 بیت باشد آنگاه:

سرعت انتقال هر مدار برابر است با:

$$\text{سرعت هر مدار} = \frac{1920\ \text{kbps}}{30} = 64\ \text{kbps}$$

زمان لازم برای انتقال حجم فایل برابر است با:

$$\text{زمان لازم انتقال فایل} = \frac{640 \times 1000\ \text{b}}{64000\ \text{bps}} = 10\ \text{sec}$$

بنابراین کل زمان انتقال برابر است با:

زمان ایجاد ارتباط + زمان انتقال فایل = کل زمان انتقال

$$\text{کل زمان انتقال} = 10\,\text{sec} + 0.5\,\text{sec} = 10.5\,\text{sec}$$

لازم به توضیح است، زمان انتقال مستقل از تعداد پیوندها می‌باشد، به عبارت دیگر اگر مدار از یک و یا تعداد یکصد پیوند استفاده نماید زمان انتقال همان 10 ثانیه خواهد بود. این زمان را اصطلاحاً زمان تأخیر دو انتهای مسیر می‌نامند. این زمان در حقیقت تأخیر انتشار پیوند است که قبلاً به آن اشاره گردید.

برنامه‌های کاربردی توزیع شده از طریق مبادله‌ی پیام‌ها بین خود، وظیفه‌ی انتقال اطلاعات بین دو میزبان را انجام می‌دهند. پیام‌ها می‌توانند شامل هر موردی شوند که منظور طراح یک پروتکل است. پیام‌ها ممکن است یک پیامد کنترلی داشته باشند، همانند پیام‌های دست‌دهی و یا حامل داده، نظیر پیام‌های پست الکترونیکی و یا آنکه فایل‌های از نوع تصویری JPEG و یا صوتی MP3 باشند.

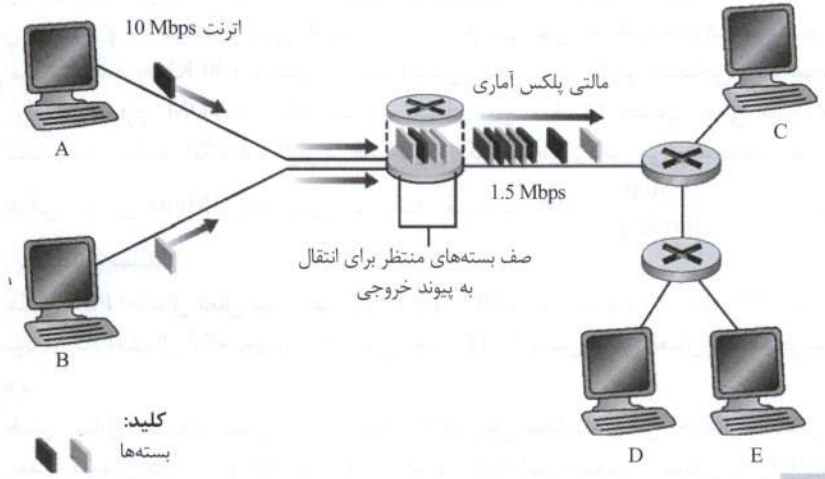
در شبکه‌های کامپیوتری فعلی، مبداء، اطلاعات را به قطعه‌های کوچکتر داده به نام بسته تقسیم نموده و سپس بین مبداء و مقصد، بسته‌ها از طریق عبور از درون پیوندهای ارتباطی و راهیاب‌های بسته که به دو گروه مسیریاب‌ها و راهیاب‌های پیوند داده تقسیم می‌شوند به حرکت درمی‌آیند. بسته‌ها با سرعت کامل هر پیوند از درون پیوندهای ارتباطی منتقل می‌شوند. اغلب راهیاب‌های بسته از روشی به نام انتقال از طریق **ذخیره و سپس ارسال**[1] استفاده می‌نمایند. پیاده‌سازی این روش در پیوندهای ورودی انجام می‌شود. انتقال ذخیره و سپس ارسال یا (SFT) این مفهوم را دارد که راهیاب ابتدا بایستی هر بسته را به‌صورت کامل آن دریافت نماید، قبل از آنکه آغاز به ارسال نخستین بیت بسته به سمت پیوند خروجی راهیاب کند. بنابراین SFT سبب ایجاد تأخیری در ورودی هر پیوند در طول مسیر بسته می‌شود.

اگر فرض شود تعداد پیوندهای بین دو میزبان برابر Q باشد و هر پیوند نیز سرعتی معادل R اختیار نماید. با تصور اینکه تنها فقط یک بسته اطلاعات در حال انتقال باشد، آنگاه ارسال بسته به طول L از نخستین پیوند در میزبان A به زمانی معادل $\frac{L}{R}$ ثانیه نیازمند است. آن بسته سپس بایستی از طریق

1- Store and forward method

هر یک از Q -1 پیوند باقیمانده‌ی شبکه، عبور نماید و به همین تعداد یعنی Q -1 مرتبه بایستی ذخیره و سپس ارسال گردد. زمان تأخیر هر مرتبه برابر $\frac{L}{R}$ ثانیه خواهد بود و بنابراین در مجموع، زمان تأخیر برابر است با $\frac{QL}{R}$ ثانیه.

هر راهیاب بسته چندین پیوند وابسته به خود دارد و بنابراین برای هر پیوند یک بافر خروجی که اصطلاحاً **صف خروجی**[1] نامیده می‌شود، وجود دارد. این صف‌ها به ذخیره‌ی بسته‌هایی می‌پردازند که روتر در حال ارسال آنها به سمت آن پیوند می‌باشد. بافر یا صف خروجی، نقش حیاتی در فناوری راهیابی بسته دارد. اگر یک بسته‌ی ورودی، نیازمند ارسال از طریق یک پیوند باشد و آن پیوند حالت اشغال را در رابطه با بسته‌ی دیگری داشته باشد، بسته‌ی ورودی بایستی در صف خروجی منتظر بماند تا نوبت ارسال آن فرا رسد. بنابراین علاوه بر زمان‌های تأخیر ذخیره و ارسال (SFT)، بسته‌ها مجبور به تحمل زمان تأخیر صف خروجی که اصطلاحاً **تأخیر صف**[2] نام دارد، نیز می‌باشند. میزان این تأخیرها متغیر و بستگی به سطح تراکم شبکه دارد. از آنجایی که فضای بافر خروجی محدود است یک بسته‌ی ورودی ممکن است با یک صف مواجه شود که بوسیله‌ی بسته‌های دیگر منتظر ارسال، پرشده باشد. در این حالت **مفقودی بسته**[3] رُخ می‌دهد و در نتیجه بسته‌ی ورودی و یا یکی از بسته‌های درون صف حذف می‌گردد. شکل ۱- ۱۴ یک شبکه‌ی ساده راهیابی بسته را نشان می‌دهد.



شکل ۱- ۱۴ ♦ راهیابی بسته

همه‌ی تصاویر بسته‌های اطلاعاتی به‌صورت صفحه‌های سه بعدی نشان داده شده‌اند. عرض (ضخامت) هر صفحه بیانگر تعداد بیت‌ها در بسته می‌باشد و همه‌ی بسته از لحاظ ضخامت و بنابراین طول یکسان در نظر گرفته شده‌اند.

اگر فرض شود میزبان‌های A و B قصد ارسال بسته‌ها را به میزبان E دارند، دو سیستم انتهایی A و B بسته‌های خود را در طول مسیر پیوند اترنت 10Mbps به نخستین راهیاب بسته گسیل می‌دارند. راهیاب بسته آنگاه این بسته‌ها را به سمت پیوند 1.5Mbps هدایت می‌نماید. چنانچه نرخ دریافت

1- Output queue
2- Queuing delays
3- Packet Loss

بسته‌ها به راهیاب فراتر از سرعتی باشد که راهیاب قادر به ارسال آنها از طریق پیوند خروجی 1.5Mbps باشد، در صف بسته‌ها در پیوند خروجی قبل از ارسال آنها تراکم رخ می‌دهد. مسایل مربوط به تراکم در بخش‌های بعدی توضیح داده شده است.

پس از بررسی هر دو روش راهیابی مداری و بسته‌ای در این مقطع لازم است مقایسه‌ای بین این دو انجام گیرد. پژوهشگران منتقد راهیابی بسته بر این باورند که این روش مناسب کاربردهای **زمان واقعی**[1] یا بی‌درنگ همانند مکالمه‌های تلفنی و کنفرانس‌های تصویری نمی‌باشد و علت آن را تأخیرهای غیر قابل پیش‌بینی و متغیر در دو انتهای مکالمه و در صف انتظار عنوان می‌کنند. از طرف دیگر طرفداران این روش استدلال می‌کنند که **شبکه‌های راهیابی بسته**[2] (PSN) به دو دلیل عمده از برتری ویژه نسبت به شبکه‌های راهیابی مدار برخوردار است. این دو علت عبارتند از:

(۱) اشتراک بهینه‌ی پهنای باند از سوی PSN و (۲) سادگی، کارایی بیشتر و هزینه‌ی کمتر در راه اندازی این‌گونه شبکه‌ها.

علت کارایی بیشتر شبکه‌های PSN را می‌توان با بررسی یک مثال ساده توضیح داد. اگر فرض شود کاربران از یک پیوند 1Mbps به‌صورت مشترک استفاده نمایند و علاوه بر آن هر کاربر در هر لحظه یا به تولید بسته‌ها با سرعت ثابت 100 Kbps می‌پردازد و یا آنکه بسته‌ای تولید نمی‌کند. اگر بر این دو، فرض سوم نیز اضافه گردد که، هر کاربر به‌صورت معمول فقط 10 درصد از زمان را به تولید بسته‌ها اختصاص می‌دهد و 90 درصد از زمان باقیمانده را در حال خواندن روزنامه صرف می‌کند. در روش راهیابی مدار، سرعت 100 Kbps بایستی در همه فواصل زمانی به هر کاربر اختصاص داده شود. به‌عنوان مثال در راهیابی مداریِ TDM، اگر یک قاب یک ثانیه‌ای بین تعداد 10 حفره‌ی زمانی هر کدام به طول 10 ms تقسیم شده باشد، آنگاه هر کاربر در قاب دارای یک حفره‌ی زمانی می‌تواند باشد و بنابراین یک پیوند راهیابی مداری 1Mbps ، همزمان می‌تواند تعداد ده $(\frac{1\ \text{Mbps}}{100\ \text{Kbps}} = 10)$ کاربر را به‌صورت هم‌زمان پشتیبانی نماید.

در حالت PSN احتمال فعال شدن هر کاربر 0.1 و یا 10 درصد است و اگر تعداد 35 کاربر در نظر گرفته شود آن‌گاه احتمال آنکه به‌صورت هم‌زمان تعداد 11 یا بیش از آن فعال باشند برابر 0.0004 خواهد بود.

در بخش مسایل تشریحی فصل، تمرین شماره ۷ چگونگی محاسبه‌ی این احتمال را نشان می‌دهد. احتمال تعداد کاربران هم‌زمان و فعال برابر 10 یا کمتر، مقدار این احتمال را معادل 0.9996 می‌کند. در این حالت مجموع نرخ ورود بسته‌ها کوچکتر و یا مساوی 1Mbps است که همان سرعت پیوند خروجی می‌باشد. بنابراین هنگامی که تعداد کاربران فعال معادل 10 و یا کمتر از این تعداد است، بسته‌های کاربران تقریباً بدون تأخیر از درون پیوند عبور می‌نمایند بنابراین، وضعیت مشابه راهیابی مداری است. اما هنگامی که این تعداد 10 و یا بیشتر از آن باشد مجموع نرخ ورود بسته‌ها از حدِ ظرفیت پیوند فراتر خواهد رفت و بنابراین صف خروجی افزایش می‌یابد. رشد صف تا زمانی ادامه پیدا می‌کند که مجموع سرعت ورودی به کمتر از 1Mbps تقلیل یابد، در این صورت است که طول صف به سمت صفر کاهش می‌یابد. از آنجایی که احتمال فعال شدن بیش از 10 کاربر هم‌زمان در این مثال بسیار ضعیف است، می‌توان به طور کلی این نتیجه را گرفت که روش راهیابی بسته کارایی، مشابه راهیابی مداری را در پی

1- Real Time

2- Packet Switching Networks

خواهد داشت. بدیهی است با وجود این تشابه در کارایی تعداد کاربران فعال PSN بیش از سه برابر روش راهیابی مداری است.

اکنون مثال ساده‌ی دیگری بیان می‌شود. اگر تعداد کاربران 10 باشد و به‌صورت ناگهانی یکی از کاربران تعداد یک هزار بسته و هر کدام به طول 1000 بیت را تولید نماید، در حالیکه 9 کاربر باقیمانده غیرفعال می‌باشند. در شرایط راهیابی مداری TDM، با تعداد 10 حفره در هر قاب و هر حفره شامل 1000 بیت، کاربر فعال تنها مجاز به استفاده از یک حفره در هر فریم برای ارسال می‌باشد، این در حالی است که 9 حفره‌ی زمانی در هر فریم بدون استفاده می‌ماند. زمان لازم برای ارسال یک میلیون بیت اطلاعات کاربر فعال برابر 10 ثانیه خواهد شد. در حالت PSN، همین کاربر فعال می‌تواند بسته‌های خود را با استفاده از سرعت کامل پیوند یعنی 1Mbps ارسال نماید، زیرا کاربران دیگر غیرفعال می‌باشند و نیاز به مالتی‌پلکس شدن با کاربر فعال ندارند. در این شرایط یک میلیون بیت اطلاعات کاربر فعال در سرعت 1Mbps به زمانی معادل یک ثانیه نیازمند است.

به عبارت دیگر سرعت انتقال PS در مقایسه با راهیابی مدار (CS) تسریع برابر 10 را در پی دارد. دو مثال بالا دو روش برای استدلال کارایی بیشتر راهیابی بسته نسبت به راهیابی مدار می‌تواند به‌شمار آید. این دو مثال تفاوت اساسی بین دو حالت اشتراک سرعت یک پیوند انتقال را در بین چندین جریان داده نشان می‌دهد. راهیابی مدار، استفاده از پیوند انتقال را بدون توجه به **نیاز** نشان می‌دهد که در آن زمان پیوند اختصاص یافته، بدون استفاده می‌ماند در حالیکه راهیابی بسته ظرفیت پیوند را براساس **نیاز** در نظر می‌گیرد و بنابراین آن را به کاربر اختصاص می‌دهد. در این روش اختصاص ظرفیت پیوند انتقال براساس بسته به بسته و تنها در بین کاربران فعال که نیازمند انتقال از طریق پیوند می‌باشند، صورت می‌پذیرد. این روش اختصاص براساس تقاضا در عوض تأمین از قبل را، جهت استفاده‌ی مشترک از منابع، **مالتی‌پلکس آماری**[1] نام نهاده‌اند.

اگرچه هر دو روش راهیابی بسته و مدار به گونه‌ی گسترده‌ای در **شبکه‌های مخابراتی**[2] (ICT) امروزی مورد استفاده قرار می‌گیرند، اما رویکرد استفاده بیشتر به سمت راهیابی بسته می‌باشد. در حال حاضر بسیاری از شبکه‌های تلفنی راهیابی مدار (CSN) به تدریج در حال تغییر به سمت شبکه‌های راهیابی بسته (PSN) می‌باشند، به‌ویژه شبکه‌های تلفنی غالباً از روش PS برای انتقال مکالمه‌های تلفنی راه دور و بین‌الملل خود که پُرهزینه می‌باشند، استفاده می‌کنند.

۱- ۹ چگونگی عبور بسته‌ها از درون PSN

قبلاً اشاره گردید یک مسیریاب (روتر) پس از دریافت یک بسته بر روی یکی از پیوندهای ارتباطی خود آن را به سمت پیوند دیگری که در خروجی آن قرار می‌گیرد ارسال می‌نماید. اکنون این سؤال مطرح می‌شود که چگونه مسیریاب به تشخیص گزینش پیوند خروجی مورد نظر از میان ده‌ها پیوند ارتباطی می‌پردازد؟

در واقع از طریق روش‌های مختلف اعمال شده بر روی انواع شبکه‌های کامپیوتری گوناگون این تصمیم‌گیری صورت می‌پذیرد. در این گفتار مقدماتی یکی از رویکردهای متداول که شبکه‌ی اینترنت از آن استفاده می‌کند، توضیح داده می‌شود.

1- Statistical Multiplexing

2- Information and Communications Technology Networks

در اینترنت هر بسته‌ی در حال عبور از شبکه در **میدان سرپیام**[1] خود حامل بخشی از اطلاعات کلی است که **آدرس مقصد**[2] (DA) نامیده می‌شود. این آدرس مشابه یک آدرس پستی دارای سلسله مراتب است. هنگامی که یک بسته به مسیریابی در شبکه وارد می‌شود، روتر بخشی از اطلاعات آدرس مقصد (DA) را پردازش و سپس بسته را براساس نتیجه‌ی پردازش به مسیریاب مجاور ارسال می‌نماید.

هر روتر برای انجام این مهم از **جدول ارسال**[3] تعریف شده در درون خود استفاده می‌نماید. مسیریاب با استفاده از این جدول آدرس مقصد و یا بخشی از آن را با جدول ارسال نگاشت می‌نماید. از نتیجه این نگاشت آدرس پیوند خروجی در مسیریاب مشخص می‌شود. بسته هنگام قرار گرفتن بر روی این پیوند از طریق آن به روتر مجاور تحویل می‌گردد. مسیریاب دریافت کننده همانند روتر نخست، بخشی از DA را پردازش و مسیر بعدی را گزینش می‌نماید. این عملیات تا رسیدن بسته به مسیریاب مقصد و از آنجا به کاربر انتهایی ادامه می‌یابد. اکنون این سؤال مطرح می‌شود که جداول ارسال خود چگونه تعریف می‌شوند؟

آیا آنها به‌صورت دستی برای هر روتر تعریف می‌شوند و یا آنکه اینترنت از طریق بکارگیری مراحل خودکار این پیکربندی‌ها را برای مسیریاب‌ها انجام می‌دهد؟

پاسخ به این پرسش موضوع فصل ۴ می‌باشد. اما در این مختصر می‌توان به مثالی از یک پروتکل مسیریابی اشاره نمود. یک پروتکل مسیریابی می‌تواند از طریق محاسبه‌ی کوتاه‌ترین مسیر از یک روتر به هر مقصد مورد نظر و استفاده از نتایج پردازش این محاسبه، جداول ارسال را در مسیریاب‌ها پیکربندی نماید.

فرآیند مسیریابی دو نقطه‌ی انتهایی را می‌توان مشابه عملکرد یک راننده خودرو که از نقشه برای رسیدن به مقصد خود استفاده نمی‌کند، تصور نمود. به‌عنوان مثال فرض کنید کوروش نامی می‌خواهد از مبداء پاسارگاد استان فارس به پلاک ۲۶- خیابان گلها در شهر دماوند- استان تهران سفر نماید. او ابتدا از مسئول پمپ بنزین پاسارگاد مسیر رسیدن به پلاک ۲۶ خیابان گلها، شهر دماوند، استان تهران را جویا می‌شود. مسئول پمپ بنزین پاسارگاد از تمامی اطلاعات آدرس مقصد، فقط بخش استان تهران آن را پردازش می‌کند و بنابراین برای کوروش توضیح می‌دهد، پس از خروج از پمپ بنزین به سمت جاده اصفهان- تهران حرکت نماید و آن جاده را تا انتها دنبال و باز هم مسیر را پرسش کند. کوروش پس از رانندگی در مسیر جاده‌ی اصفهان- تهران به شهر اصفهان وارد می‌شود و از مسئول پمپ بنزین آدرس پلاک ۲۶،..... دماوند، تهران را جویا می‌شود. مسئول پمپ بنزین تنها به بخش تهران آدرس توجه می‌کند، آنرا در ذهن خود پردازش و سپس کوروش به سمت جاده اصفهان- تهران هدایت می‌شود. پس از رسیدن به تهران کوروش از مسئولی در پمپ بنزین تهران جویای آدرس ۲۶ خیابان گلها، دماوند، تهران می‌شود. در اینجا باز مسئول پمپ بنزین تنها بخش دماوند آدرس را پردازش نموده و وی را به سمت جاده تهران- دماوند هدایت می‌کند.

سرانجام کوروش به شهر دماوند می‌رسد و از پمپ بنزین دماوند جویای آدرس ۲۶ خیابان گلها، دماوند، تهران می‌شود. مسئول مربوطه وی را به سمت خیابان گلها هدایت می‌کند یعنی تنها فقط همین بخش از آدرس را در ذهن خود پردازش می‌نماید. سرانجام کوروش وارد خیابان گلها شده و با مشاهده‌ی پسربچه دوچرخه سواری آدرس مقصد خود را از وی جویا می‌شود و پسربچه در ذهن خود

1-Header field
2- Destination address
3- Forwarding table

تنها فقط ۲۶ را پردازش و پس از نتیجه‌گیری با دست خود به پلاک ۲۶ اشاره می‌کند و کوروش به مقصد نهایی خود پای می‌نهد.

در شبیه‌سازی انجام شده از چگونگی این عملکرد، مسیریاب‌های شبکه پمپ بنزین‌ها و پسربچه نقش روترها را دارند و جداول ارسال آنها اطلاعاتی است که در طول سالیان در حافظه آنها در اثر یادگیری و تجربه اندوخته شده است. چنانچه علاقه‌مند به مشاهده مسیر بسته‌های اطلاعاتی از درون مسیریاب‌ها در اینترنت می‌باشید می‌توانید به برنامه‌ی رهگیری مسیر در آدرس اینترنتی:

http://www.traceroute.org

مراجعه نمائید.

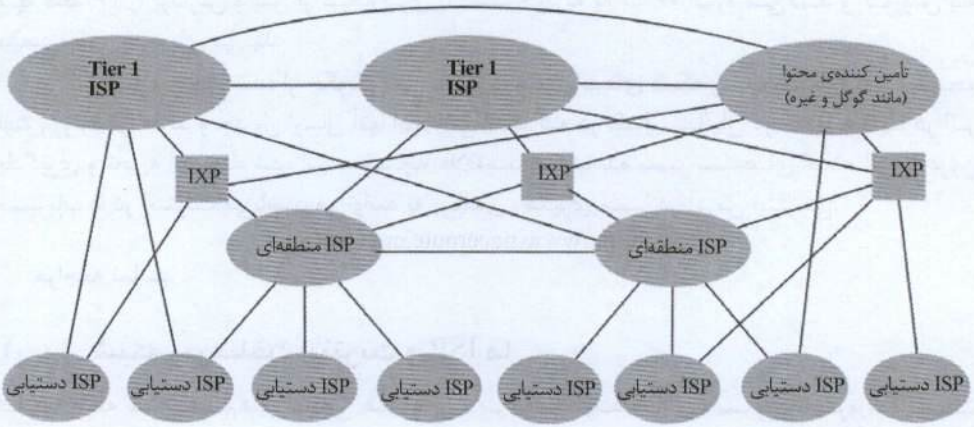
۱- ۱۰ شبکه زیرساخت اینترنت و ISP ها

قبلاً ملاحظه شد سیستم‌های انتهایی همانند نوت بوک‌ها، تبلت‌ها، وب سرورها، سرورهای پست الکترونیکی و نظایر آنها از طریق ISPهای محلی به شبکه‌ی اینترنت متصل شده‌اند. یک ISP می‌تواند امکانات ارتباطی این کاربران را از طریق ارتباط‌های کابلی نظیر خطوط DSL، نوری، FTTH و یا ارتباط‌های بدون سیم همانند Wi-Fi، Wi-MAX فراهم آورد. بدیهی است یک ISP محلی در قالب یک مجتمع دانشگاهی که امکانات دسترسی به اینترنت را برای دانشجویان، کارکنان و استادان خود فراهم می‌آورد و یا یک شرکت تجاری که دستیابی را برای کارکنان خود امکان‌پذیر می‌سازد، می‌تواند تعریف گردد.

اما برقراری ارتباط کاربران انتهایی و تأمین‌کنندگان خدمات اطلاع‌رسانی با ISPهای محلی، تنها بخش کوچکی از یک معمای بزرگ به‌شمار می‌آید. این معمای بزرگ عبارت است از: ارتباط صدها میلیون سیستم انتهایی و صدها هزار شبکه‌ی کامپیوتری به یکدیگر که مجموعاً تشکیل شبکه‌ی اینترنت را می‌دهند. بنابراین در یک عبارت خلاصه می‌توان اینترنت را **شبکه‌ای از شبکه‌ها** دانست. درک این عبارت ساده کلید یافتن راه حل این معما است. در شبکه‌ی اینترنت عمومی، ISPهای قابل دسترس در نبش اینترنت از طریق یک **سلسله مراتب لایه‌ای**[1] به دیگر بخش‌های اینترنت متصل شده‌اند. همانگونه که در شکل ۱- ۱۵ نشان داده شده است، ISPهای در دسترس در بخش زیرین این سلسله مراتب قرار می‌گیرند، در حالیکه بالاترین بخش این ساختار شامل تعداد اندکی ISP است که اصطلاحاً به آنها **ISP لایه‌ی اول**[2] گفته می‌شود. غالباً ISPهای لایه‌ی اول مشابه هر شبکه‌ی کامپیوتری هستند، به این مفهوم که دارای پیوندها و مسیریاب‌ها می‌باشند و با دیگر شبکه‌های کامپیوتری در ارتباط می‌باشند. سرعت پیوندهای این ISPها برابر 622Mbps و یا بالاتر است و ISPهای از این نوع در قالب وسعت بیشتر دارای پیوندهای با سرعت 2.5 تا 10گیگا بیت بر ثانیه می‌باشند که در نتیجه مسیریاب‌های آنها ارسال بسته‌ها را با سرعت فوق‌العاده چشمگیر انجام می‌دهند. این نوع ISPها هم چنین دارای ویژگی‌های ذیل هستند:

- دارای ارتباط مستقیم با خود و دیگر ISPهای لایه اول می‌باشند.
- به تعداد انبوهی از ISPهای از نوع لایه دوم و دیگر شبکه‌های کاربران لایه‌های دیگر متصل می‌باشند.
- پوشش بین المللی دارند.

1- Hierarchical layers
2- Tier 1- ISP



شکل ۱ - ۱۵ ♦ ارتباط بین ISPها

این نوع ISPها را اصطلاحاً **شبکه‌های زیرساخت**[1] (مادر) اینترنت نیز می‌نامند. در محدوده‌ی پوشش محلی و کشوری، ISPها از نوع رده‌ی اول می‌باشند که تنها به تعداد محدودی ISP در لایه اول متصل شده‌اند. بنابراین برای دستیابی به بخش وسیع‌تر شبکه‌ی اینترنت این نوع ISPها نیازمند انتقال ترافیک خود از درون یک یا بیش از یک ISP رده‌ی اول که به آن متصل شده‌اند، می‌باشند. این نوع ISPها را اصطلاحاً مشتری ISPهای لایه‌ی اول می‌خوانند. بسیاری از شرکت‌های بزرگ و سازمان‌ها در سراسر گیتی شبکه‌های خود را به یکی از این دو نوع ISP متصل می‌کنند و بنابراین به‌عنوان مشتری آنها حساب می‌شوند. یک ISP تأمین‌کننده، از ISPهای مشتری بابت استفاده از خدمات به نسبت سرعت انتقال پیوندهای بین آنها هزینه طلب می‌نماید.

علاوه بر این موارد شبکه‌های ISP لایه‌ی دوم می‌توانند به‌صورت مستقیم به یکدیگر متصل شده و انتقال ترافیک را بین خود و بدون نیاز به لایه‌ی اول انجام دهند. در بخش زیرین ISPهای لایه‌ی دوم دیگر ISPهای سطح پائین قرار می‌گیرند، که از طریق ISPهای بالاتر از خود به اینترنت جهانی متصل می‌شوند و سرانجام در بخش زیرین این ISPها، تأمین کننده‌های خدمات قابل دسترس واقع شده‌اند. یادآوری این نکته ضروری است برخی از ISPهای لایه‌ی اول خود از نوع لایه‌ی دوم نیز می‌باشند، بنابراین قادرند خدمات خود را بدون واسطه به کاربران انتهایی و تأمین‌کنندگان خدمات اینترنت به فروش رسانند. هنگامی که دو ISP بدون واسطه به یکدیگر متصل می‌شوند و هر دو از لحاظ لایه‌یکسان می‌باشند آنها را **نظیر**[2] یکدیگر می‌نامند. در بین شبکه‌ی ISPها، نقطه‌ای که معمولاً هر نوع ISP به ISP دیگر متصل می‌شود، **نقاط حضور**[3] (POPs) نامیده می‌شود. یک POP عبارت است از یک گروه شامل یک یا بیشتر از یک مسیریاب در شبکه‌ی ISP که روترهای دیگر در ISPهای مختلف و یا شبکه‌های متعلق به مشتریان می‌توانند به آنها متصل گردند. یک تأمین کننده‌ی لایه اول دارای تعداد زیادی POP می‌باشد که از لحاظ جغرافیایی پراکنده شده‌اند و شبکه‌ی مشتریان و ISPهای دیگر به هر یک از این POPها متصل شده‌اند. برای اتصال یک شبکه‌ی مشتری به POP تأمین کننده‌ی خدمات

1- Internet backbone
2- Peer
3- Points of Presence

معمولاً از پیوندهای سریع اجاره‌ای که توسط شرکت‌های مخابراتی فراهم می‌شود و به‌صورت مستقیم به یکی از مسیریاب‌های POP تأمین‌کننده متصل شده است، استفاده می‌شود. علاوه بر همه‌ی این موارد دو ISP ممکن است شامل چندین نقطه‌ی هم نظیر باشند که هر یک از آنها به دیگری از طریق چندین جفت POP متصل شده است.

به‌منظور ساختن شبکه‌ای که شباهت بسیار نزدیک با اینترنت امروزی دارد بایستی عواملی مانند: نقاط حضور (POPs)، چند خانگی، نظیرسازی و نقاط تبادل اینترنتی (IXP) را به سلسله‌مراتب ساختار شبکه در سطح سوم اضافه نمود. نقاط حضور اینترنتی در همه‌ی سطوح سلسله‌مراتب به جز در سطح دستیابی شبکه حضور دارند. یک POP معمولاً گروهی از مسیریاب‌های مقیم در یک مکان است که در مالکیت یک ISP قرار دارد و مشترکین ISPها امکان ارتباط با آن را دارند. برای آنکه یک مشترک شبکه قادر به ارتباط با POP باشد نیازمند اجاره‌ی یک پیوند سرعت بالا از یک شرکت مخابراتی است که مستقیماً یکی از مسیریاب‌های خویش را به مسیریابی در آن POP متصل نماید. هر ISP به غیر از ISPهای مقیم در رده‌ی یک یعنی tier-1 ممکن است از امکان اصطلاحاً **چندخانگی**[1] بهره‌گیری نماید تا آن را قادر به ارتباط با دو یا چند تأمین‌کننده‌ی ISP نماید. به‌عنوان مثال یک ISP دستیابی ممکن است بهره‌مند از امکان چندخانگی با دو ISP محلی یا منطقه‌ای باشد و یا آنکه فقط با دو ISP منطقه‌ای و نیز یک ISP در رده‌ی نخست (tier-1) چند خانه گردد. به‌گونه‌ی مشابه یک ISP محلی ممکن است با چندین ISP از نوع نخست (tier-1) چندخانه شود. هنگامی که یک ISP بهره‌مند از ویژگی چندخانگی باشد آنگاه در صورت وقوع یک اختلال برای یکی از تأمین‌کنندگان خود باز هم قادر به ارسال و یا دریافت بسته‌ها از اینترنت خواهد بود. قبلاً اشاره گردید مشترکین ISPها هزینه‌ای را بابت دسترسی به ارتباط‌های جهانی خود به آنها پرداخت می‌نمایند. هزینه‌ای که یک مشترک ISP به تأمین‌کننده‌ی خدمات مربوطه می‌پردازد در واقع منعکس‌کننده‌ی مقدار ترافیک مبادله‌شده بین وی و ISP می‌باشد. به منظور کاهش این هزینه‌ها یک جفت ISP که در یک سطح مشابه از سلسله‌مراتب قرار دارند می‌توانند نقش نظیر یا **قرین**[2] یکدیگر را بازی نمایند به سخن دیگر آنها می‌توانند مستقیماً شبکه‌های یکدیگر را به هم متصل نموده و درنتیجه تمامی حجم ترافیک بین آنها از طریق یک ارتباط مستقیم به جای ارتباط‌های واسطی و یا از طریق دیگر ISPها عبور داده شود. هنگامی که دو ISP نظیرسازی یکدیگر را انجام می‌دهند معمولاً هیچ‌یک از آنها هزینه‌ای را به دیگری بابت انتقال ترافیک نمی‌پردازد. همان‌گونه که قبلاً ملاحظه شده ISPهای واقع در tier-1 امکان نظیر شدن با یکدیگر را دارند. علاوه بر موارد اشاره شده یک شرکت تجاری ثالث می‌تواند اقدام به ایجاد یک **نقطه‌ی تبادل اینترنتی**[3] **(IXP)** نماید. این امکان در محیط خصوصی آن شرکت و از طریق راهیاب‌های اختصاصی آن نیز انجام‌پذیر می‌باشد. معمولاً یک IXP نقطه‌ی تلاقی چندین ISP است که امکان نظیرسازی یکدیگر را نیز دارند. در حال حاضر تقریباً تعداد 300 مورد IXP در سراسر شبکه‌ی اینترنت وجود دارند. این سیستم‌های اصطلاحاً همزیست که شامل ISPهای دستیابی، ISPهای منطقه‌ای یا محلی، ISPهای رده‌ی نخست یا tier-1، ISPهای چندخانگی، POPها، نظیرسازها و نقاط تبادل اینترنتی (IXP) می‌شوند اصطلاحاً **ساختار شبکه‌ی سطح 4**[4] نامیده شده‌اند.

1- Multi - home
2- Internet Exchange Point
3- Peering
4- Network Structure 4

سرانجام **ساختار شبکه‌ی سطح 5** قرار دارد که شبکه‌ی اینترنت 2012 را توصیف می‌کند. این ساختار در شکل ۱-۱۵ نشان داده شده است و بر فراز ساختار شبکه‌ی 4 از طریق افزودن **شبکه‌های تأمین محتوا**[1] **(CPN)** ساخته می‌شود. شبکه‌ی CPN گوگل در حال حاضر یکی از طلایه‌داران شبکه‌های تأمین محتوا به‌شمار می‌آید. تا زمان نگارش ویراست ششم این اثر تقریباً شرکت گوگل بهره‌مند از تعداد 30 تا 50 مرکز داده‌ی توزیع شده در سراسر آمریکای شمالی، اروپا، آسیا، آمریکای جنوبی و استرالیا بوده است. پاره‌ای از این مراکز داده محل اسکان بیش از یکصد هزار سرور هستند در حالی‌که مراکز داده‌ی دیگر شامل صدها سرور از این گونه می‌باشند. مراکز داده‌ی گوگل همگی از طریق شبکه‌ی TCP/IP اختصاصی گوگل به یکدیگر متصل شده‌اند و تقریباً این شبکه در سراسر گیتی گسترده شده است و از شبکه‌ی اینترنت عمومی نیز مجزا می‌باشد. بدیهی است شبکه‌ی اختصاصی گوگل انحصاراً ترافیک بین سرورهای گوگل را انتقال می‌دهد و همان‌گونه که شکل ۱-۱۵ نشان می‌دهد این شبکه از طریق نظیرسازی با ISPهای واقع در tier-1 سعی در پرهیز از ارتباط با رده‌های بالاتر و در نتیجه حذف پرداخت هزینه را دارد. گوگل این مهم را از طریق ارتباط مستقیم با اینگونه شبکه‌ها و یا از طریق ارتباط با آنها در سطح IXP ها محقق می‌سازد. از آنجا که بسیاری از ISPهای دستیابی هنوز هم فقط از طریق انتقال ترافیک عبوری از شبکه‌های tier-1 قابل دسترس هستند شبکه‌ی گوگل به این‌گونه ISPها متصل شده و هزینه‌ی ترافیک عبوری را نیز به آنها پرداخت می‌نماید. از طریق ایجاد یک شبکه‌ی اختصاصی، یک تأمین‌کننده‌ی محتوا نه‌تنها هزینه‌های خود به ISPهای واقع در رده‌های بالاتر را کاهش می‌دهد بلکه کنترل به مراتب بیشتری را نیز در مورد چگونگی تحویل خدمات به کاربران انتهایی خویش اعمال می‌کند. در فصل ۷ این موضوع با جزئیات بیشتری دنبال خواهد شد.

در پایان شبکه‌ی اینترنت امروزی به‌عنوان شبکه‌ای از شبکه‌ها شامل پیچیدگی شگرف است که از ترکیب تعدادی ISP در رده‌ی نخست (tier-1) و صدها هزار ISP در رده‌های پایین‌تر تشکیل شده است. این ISPها از نقطه‌نظر پوشش و خدمات کاملاً واگرا و در سراسر گیتی گسترده هستند. چگونگی ارتباط آنها به‌صورت ISPهای رده‌های پایین به رده‌های بالاتر و سپس ارتباط این ISPها به یکدیگر می‌باشد. کاربران معمولی و تأمین‌کنندگان محتوا همگی مشترکین ISPهای رده‌های پایین‌تر هستند و این ISPها خود مشتریان رده‌های بالاتر به‌شمار می‌آیند. از نقطه‌نظر سازماندهی، در ساختار شبکه‌ی **1** اتصال همه‌ی ISPهای دستیابی با یک ISP ترانزیت جهانی و در ساختار شبکه‌ی **2** صدها هزار ISP دستیابی با چندین ISP ترانزیت جهانی قرار می‌گیرند. ساختار شبکه‌ی **3** عبارت است از سلسله‌مراتب رده‌های چندگانه به یکدیگر، به‌عنوان مثال در کشور چین ISPهای دستیابی در هر شهر حضور دارند و به ISPهای استانی متصل شده‌اند و این ISPها نیز به نوع کشوری یا ملی خود مرتبط شده‌اند و سرانجام آنها نیز به ISPهای واقع در tier-1 اینترنت متصل می‌شوند. در سال‌های اخیر تأمین‌کنندگان مهم محتوا نسبت به ایجاد شبکه‌های اختصاصی خود اقدام نموده‌اند و این شبکه‌ها را هر کجا که امکان آن میسر بوده است مستقیماً به ISPهای واقع در رده‌های پایین‌تر به‌منظور پرهیز از پرداخت هزینه متصل ساخته‌اند.

۱-۱۱ تأخیر، تضعیف و بازدهی در شبکه‌های PSN

قبلاً اشاره گردید اینترنت را می‌توان به‌عنوان یک زیرساخت جهت فراهم‌آوری خدمات به نرم‌افزارهای کاربردی توزیع شده در سیستم‌های انتهایی قلمداد نمود.

1- Content Provider Networks

هدف اصلی، انتقال حداکثری اطلاعات در قالب داده‌ها و به‌صورت بی‌درنگ در بین میزبان‌ها بدون اتلاف آنها می‌باشد. در عمل، دستیابی به این هدف تقریباً غیرممکن است زیرا شبکه‌های کامپیوتری محدودیتی در قالب میزان بازدهی یا به عبارت دیگر مقدار داده‌ی قابل انتقال در هر ثانیه را بر فرآیند انتقال تحمیل می‌نمایند.

این محدودیت‌ها شامل تأخیرها بین میزبان‌ها و یا تلف شدن بسته‌ها می‌شود. از یک طرف این ناخوشایندی وجود دارد که واقعیت قوانین فیزیکی سبب بروز تأخیر و اتلاف و نیز محدودیت در بازدهی می‌گردد و از طرف دیگر وجود این مشکلات در شبکه‌های کامپیوتری سبب دامن زدن به موضوعاتی می‌شود که یافتن راه کار برای هر یک از آنها در قالب رساله‌های پژوهشی پیشرفته امکان‌پذیر می‌باشد. لازم به یادآوری است، یک بسته حرکت خود در یک مسیر شبکه را از طریق یک میزبان (مبداء) آغاز می‌کرد و سپس با عبور از درون گروهی از مسیریاب‌ها (روترها) سرانجام خود را به میزبان مقصد می‌رسانید.

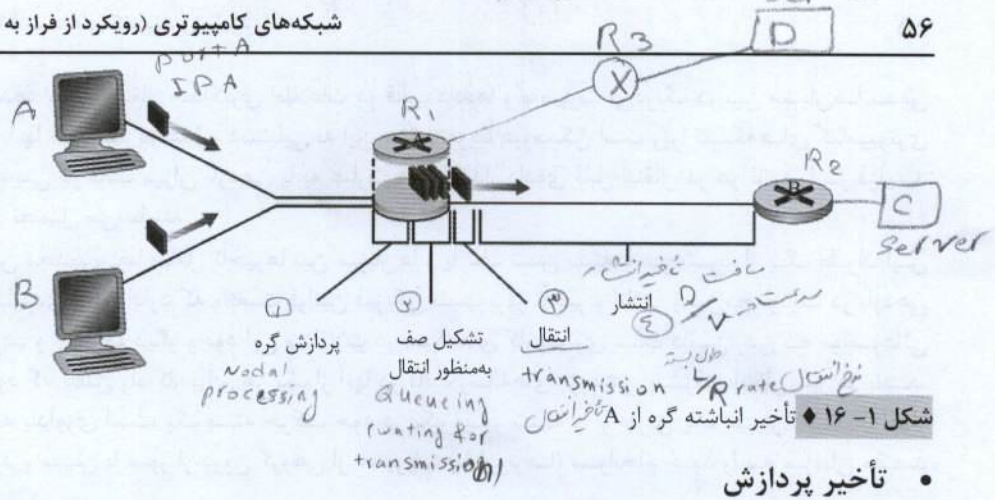
هنگام عبور یک بسته از یک گره (میزبان یا مسیریاب) به گره‌ی دیگر (میزبان یا مسیریاب) در طول یک مسیر، انواع گوناگونی از تأخیرها در هر یک از گره‌ها به بسته تحمیل می‌گردد. مهمترین این تأخیرها عبارتند از:

- **تأخیر پردازش گره**[1]
- **تأخیر صف**[2]
- **تأخیر انتقال**[3]
- **تأخیر انتشار**[4]

مجموع این چهار تأخیر در واژگان شبکه‌های کامپیوتری **تأخیر انباشته‌ی گره** نامیده شده است. با توجه به شکل ۱- ۱۶ انواع این تأخیرها را می‌توان چنین بررسی نمود: **یک بسته‌ی ارسالی به‌عنوان بخشی از مسیر نقطه به نقطه‌ی خود بین مبداء و مقصد از درون مسیریاب A گذر نموده و به B می‌رسد.**

اگر هدف شناسایی تأخیر انباشته در گره‌ی A باشد، آنگاه بایستی توجه نمود که روتر A دارای یک پیوند خروجی به سمت مسیریاب B می‌باشد که قبل از این پیوند یک صف (بافر) وجود دارد. هنگامی که بسته وارد مسیریاب A می‌شود، اطلاعات سرپیام بسته بوسیله‌ی A بررسی شده و از نتیجه‌ی پردازش اطلاعات درون عنوان بسته پیوند خروجی مناسب برای بسته گزینش و سپس بسته به سمت آن پیوند هدایت می‌شود. در این مثال پیوند خروجی برای بسته سرانجام به روتر B منتهی می‌گردد. یک بسته فقط هنگامی می‌تواند بر روی یک پیوند جهت ارسال قرار گیرد که در آن لحظه هیچ بسته‌ی دیگری در حال انتقال بر روی آن پیوند نباشد و یا آنکه بسته‌های دیگری قبل از آن در صف انتظار جهت ارسال قرار نگرفته‌اند، در غیر این صورت بسته بایستی به صف انتظار جهت اعزام بپیوندد.

1- Nodal Processing Delay
2- Queuing Delay
3- Transmission Delay
4- Propagation Delay



شکل ۱- ۱۶ ♦ تأخیر انباشته گره از A

- **تأخیر پردازش**

زمان لازم جهت پردازش اطلاعات درون عنوان بسته و نتیجه حاصل از آن به‌عنوان بخشی از تأخیر پردازش به‌شمار می‌آید. این تأخیر علاوه بر موارد اشاره شده می‌تواند شامل عوامل دیگر از قبیل زمان لازم برای ارزیابی و تشخیص عدم وقوع خطا بر روی بیت‌های بسته به هنگام گسیل آنها به سمت روتر A نیز باشد.

در مسیریاب‌های سریع طول زمان تأخیر پردازش در محدوده چند میلی‌ثانیه و یا کمتر از آن می‌باشد. پس از تحمل این تأخیر پردازش بسته به‌وسیله‌ی مسیریاب به صف (بافر) هدایت می‌شود که قبل از پیوند ارسال بسته به روتر B قرار دارد.

- **تأخیر صف**

زمان انتظار بسته در صف جهت انتقال به پیوند تأخیر صف نامیده می‌شود. طول زمان تأخیر صف برای هر بسته به تعداد بسته‌های قبلی در حال انتظار صف و زمان انتظار هر یک برای انتقال از طریق پیوند بستگی دارد. اگر صف تُهی باشد و بسته‌ی دیگری در حال عبور از پیوند نباشد، این زمان تقریباً برابر صفر است. از طرف دیگر اگر حجم ترافیک ورودی به صف سنگین باشد و تعداد قابل توجهی از بسته‌های دیگر در انتظار ارسال می‌باشند، زمان تأخیر صف طولانی خواهد شد.

بدیهی است تعداد بسته‌هایی که یک بسته در حال ورود به یک صف ممکن است با آن مواجه گردد تابعی است از شدت و نوع ترافیک ورودی به صف. اندازه تأخیر صف در عمل در محدوده‌ی چند میکرو یا میلی ثانیه است.

- **تأخیر انتقال**

با فرض آنکه بسته‌ها براساس روش **نخستین ورودی به صف اولین خروجی از صف**[1] **(FIFO)** ارسال می‌شوند، بنابراین یک بسته‌ی ورودی تنها فقط هنگامی فرصت انتقال را به‌دست می‌آورد که همه‌ی بسته‌های قبل از آن گسیل شده باشند. اگر طول بسته از لحاظ تعداد بیت L فرض شود و سرعت انتقال پیوند A به B برابر Rbps در نظر گرفته شود، آنگاه زمان تأخیر انتقال برابر $\frac{L}{R}$ خواهد شد. این زمان مشابه و یا به عبارت دیگر همان زمان ذخیره و ارسال می‌باشد. این زمان در واقع برابر است با زمان لازم جهت انتقال همه‌ی بیت‌های یک بسته از طریق یک پیوند، نظیر پیوند A به B. اندازه‌ی این زمان در محدوده‌ی چند میکرو یا میلی ثانیه است.

1- First in First Out

• تأخیر انتشار

هنگامی که یک بیت در درون پیوند قرار می‌گیرد بایستی به سمت مسیریاب B انتشار یابد. زمان لازم جهت انتشار از نقطه‌ی آغاز پیوند تا رسیدن آن به روتر B **زمان انتشار** نامیده شده است. انتشار بیت متأثر از سرعت انتشار پیوند است. این سرعت وابسته به رسانه‌ی فیزیکی پیوند است. سرعت انتشار یک پیوند با توجه به رسانه‌ی فیزیکی آن در محدوده‌ی 2×10^8 m/s تا 3×10^8 m/s می‌باشد. زمان تأخیر انتشار عبارت است از نسبت فاصله‌ی بین دو مسیریاب به سرعت انتشار. اگر d فاصله‌ی بین دو مسیریاب و v سرعت انتشار پیوند باشد، آنگاه تأخیر انتشار برابر است با $\frac{d}{v}$. هنگامی که آخرین بیت بسته به گره‌ی B می‌رسد، آن بیت و همه‌ی بیت‌های قبل از آن در مسیریاب B ذخیره می‌شوند. بدیهی است تمامی این فرآیندها در روتر B نیز دنبال می‌شود. در شبکه‌ی گسترده (WAN) اندازه‌ی این زمان در محدوده‌ی چند میلی‌ثانیه است. ممکن است برای بسیاری از خوانندگان تفاوت بین دو زمان تأخیر انتقال و تأخیر انتشار محسوس نباشد بنابراین، برای روشن شدن موضوع مثالی آورده می‌شود.

مجموعه‌ای از بزرگراه‌های بهم پیوسته را تصور نمائید که در فاصله 100 کیلومتر از یکدیگر دارای اتاقک‌های دریافت عوارض بزرگراه می‌باشند. در این حالت بزرگراه‌ها را می‌توان پیوندها و اتاقک‌های دریافت عوارض را مسیریاب‌ها در نظر گرفت. شکل ۱- ۱۷ قطعه‌های مختلف بزرگراه را بین اتاقک‌های دریافت عوارض بزرگراه نشان می‌دهد.



شکل ۱- ۱۷ ♦ تفاوت بین تأخیر انتقال و انتشار

اگر فرض شود سرعت حرکت خودروها به گونه‌ی یکنواخت معادل 100 کیلومتر در ساعت باشد و تعداد 10 خودرو در قالب یک کاروان در پی یکدیگر در بزرگراه در حال حرکت باشند. در این صورت هر خودرو را می‌توان معادل یک بیت و کاروان شامل 10 خودرو را یک بسته در نظر گرفت. علاوه بر موارد بالا باز هم اگر فرض شود، زمان پردازش و انتظار هر خودرو جهت پرداخت عوارض معادل 12 ثانیه باشد و اگر نخستین خودرو (بیت) به هنگام ورود در مقابل اتاقک عوارض منتظر تعداد نُه خودرو دیگر جهت پیوستن به آن قبل از پرداخت عوارض باشد، آنگاه زمان لازم برای انتقال آخرین خودرو (دریافت عوارض) به‌وسیله‌ی اتاقک دریافت عوارض برابر است با: فرآیند دریافت عوارض 12 ثانیه برای هر خودرو و یا به عبارت دیگر یک دقیقه برای هر 5 خودرو می‌باشد، بنابراین تعداد 10 خودرو معادل 2 دقیقه زمان جهت پرداخت عوارض لازم دارند. اگر تعداد خودروها L و سرعت دریافت عوارض R باشد آنگاه می‌توان نتیجه گرفت: 2 دقیقه عبارت است از زمان تأخیر انتقال.

از طرف دیگر فاصله‌ی هر قطعه از بزرگراه 100 کیلومتر فاصله‌ی پیوند یا مسافت بین دو مسیریاب (d) و سرعت حرکت (انتشار) هر خودرو نیز 100 کیلومتر در ساعت است، بنابراین زمان لازم جهت انتشار هر خودرو از یک اتاقک به اتاقک دیگر برابر است با $\frac{d}{v}$ و یا یک ساعت (60 دقیقه) و به عبارت دیگر در این مثال 60 دقیقه زمان انتشار است. در نتیجه طول زمان از هنگامی که کاروان خودروها در مقابل یک اتاقک صف می‌کشند (ذخیره می‌شوند) تا زمانی که آنها در برابر اتاقک بعدی صف تشکیل دهند برابر 62 دقیقه است.

حال اگر تصور شود زمان پردازش هر خودرو از لحاظ دریافت عوارض به‌وسیله‌ی اتاقک‌ها در مقایسه با زمان حرکت خودروها دچار افزایش چشمگیری گردد، آنگاه چه تأثیری را در فرآیند نتیجه نهایی ایجاد خواهد نمود؟ به عبارت دیگر کاهش چشمگیر زمان انتشار چه تأثیری را بر روی نتیجه نهایی انتقال از خود به جای می‌گذارد؟

اگر تصور شود سرعت یکنواخت حرکت خودروها از 100 به 1000 کیلومتر در ساعت و زمان پرداخت عوارض نیز از 12 ثانیه به یک دقیقه برای هر خودرو افزایش یابد. آنگاه زمان انتقال (پرداخت عوارض کاروان) برابر 10 دقیقه و زمان انتشار (پیمودن فاصله‌ی بین دو اتاقک) برابر 6 دقیقه خواهد شد. مفهوم مقایسه‌ی این دو زمان آن است که چندین خودرو از کاروان هنگامی که وارد اتاقک عوارض دوم می‌شوند هنوز تعدادی از خودروهای کاروان در انتظار پرداخت عوارض در اتاقک اول می‌باشند. وقوع این حالت به‌صورت واقعی در شبکه‌های راهیابی بسته رخ می‌دهد. نخستین بیت‌های یک بسته در حالی وارد یک مسیریاب می‌شوند که تعداد قابل توجهی از آنها در حال انتظار جهت ارسال در مسیریاب قبل به سر می‌برند. اکنون اگر چهار زمان تأخیر نامبرده در بالا را به ترتیب d_{proc}، d_{queue}، d_{trans} و d_{prop} نامگذاری نماییم، آنگاه زمان تأخیر انباشته در یک گره اگر d_{nodal} نامیده شود برابر است با:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \qquad (۱-۱)$$

تأثیر این عوامل تأخیر می‌تواند به‌صورت چشمگیری متفاوت باشد. به‌عنوان مثال d_{prop} اگر در حد میکروثانیه برای یک مسیر کوتاه باشد به راحتی می‌توان از آن صرف نظر نمود در حالیکه همین زمان برای یک مسیر بسیار طولانی نظیر پیوندهای ماهواره‌ای چشمگیر و تأثیرگذار است. به گونه‌ی مشابه زمان d_{trans} برای شبکه‌های محلی (LAN) با سرعت 10Mbps و یا بیشتر بسیار کوتاه می‌باشد. در صورتی که همین زمان تأخیر برای بسته‌های اینترنتی (از لحاظ حجم طولی زیاد) و پیوندهای کُند نظیر مودم معمولی به صدها میلی‌ثانیه خواهد رسید و بنابراین بر روی نتیجه زمان تأخیر انباشته گره تأثیر چشمگیری را به همراه دارد و نمی‌توان از آن صرف نظر نمود. در اغلب موارد زمان d_{proc} بسیار ناچیز است اما در بازدهی مسیریاب‌ها بسیار مؤثر است زیرا تعیین‌کننده‌ی حداکثر سرعت انتقال بسته‌ها از طریق روتر می‌باشد.

۱-۱۲ تأخیر صف و اتلاف بسته‌ها

یکی از پیچیده‌ترین عناصر تأخیر انباشته‌ی گره، تأخیر صف است. بررسی تأثیرهایی که این عامل بر روی عملکرد کلی شبکه‌های کامپیوتری می‌گذارد به اندازه‌ای قابل توجه است که هزاران مقاله‌ی پژوهشی و پایان نامه‌های دانشگاهی پیرامون آن نوشته شده است و هنوز هم از جایگاه ویژه‌ای در میان مهمترین موضوعات پژوهشی برخوردار است. برخلاف سه تاخیر d_{proc}، d_{trans} و d_{prop} این تأخیر می‌تواند در میان بسته‌ها متفاوت و متغیر باشد. به‌عنوان مثال اگر تعداد 10 بسته هم‌زمان به یک صف تهی وارد شوند نخستین بسته مجبور به تحمل زمان تأخیر صف نمی‌باشد، درحالیکه آخرین بسته مجبور به تحمل زمان تأخیر صف طولانی می‌باشد زیرا بایستی در انتظار گسیل نُه بسته دیگر باشد. بنابراین هنگامی که در مورد ویژگی‌های تأخیر صف مطالعه می‌شود معمولاً از عوامل اندازه گیری آماری نظیر میانگین تأخیر صف، واریانس تأخیر صف و یا احتمال آنکه تأخیر صف از یک میزان معین فراتر گردد، استفاده می‌شود. پاسخ به پرسش‌هایی از قبیل چه هنگام تأخیر صف بسیار طولانی است؟ و

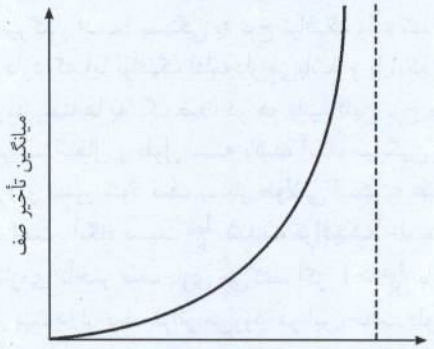
یا چه هنگامی این تأخیر تأثیرگذار است؟ بستگی به نرخ ترافیک وارد شده به یک صف، سرعت انتقال پیوند و نوع ترافیک ورودی دارد که آیا ترافیک ادامه‌دار می‌باشد و یا آنکه لحظه‌ای است.

اگر فرض شود a نرخ ورود بسته‌ها به یک صف در هر ثانیه (این نرخ معمولاً به‌صورت بیت بر ثانیه است) و R و L به ترتیب سرعت انتقال و طول بسته باشد، آنگاه میانگین ورود بیت‌ها به یک صف برابر La بیت بر ثانیه است. حال اگر تصور شود صف بسیار طولانی است به عبارت دیگر تعداد بیت‌های ذخیره شده در آن نامحدود است، آنگاه نسبت $\frac{La}{R}$ **شدت ترافیک**[1] نامیده می‌شود و در اغلب موارد نقش مهمی را در تعیین اندازه‌ی تأخیر صف بازی می‌کند. اگر $\frac{La}{R} > 1$ باشد آنگاه میانگین سرعت ورود بیت‌ها به صف از نرخ انتقال بیت‌ها از صف فراتر می‌رود. در این حالت طول صف بدون محدودیت رشد می‌کند و در نتیجه زمان تأخیر آن نیز به سمت بی نهایت میل می‌کند. بنابراین یکی از کلیدی‌ترین اصول طراحی مهندسی ترافیک آن است که همواره طراحی یک شبکه بایستی به سمتی هدایت گردد که شدت ترافیک $(\frac{La}{R})$ از مقدار یک فراتر نرود.

از طرفی اگر $\frac{La}{R} \leq 1$ باشد نوع ترافیک ورودی در تأخیر صف تأثیرگذار خواهد بود. به‌عنوان مثال اگر بسته‌ها به‌صورت لحظه‌ای یعنی هر بسته نسبت به بسته بعدی در فاصله‌ی زمانی $\frac{L}{R}$ ثانیه وارد صف شوند آنگاه هر بسته وارد یک صف تهی شده و در نتیجه تأخیر صف وجود ندارد. از سوی دیگر اگر بسته‌ها به‌صورت جمعی اما لحظه‌ای وارد شوند، تأثیر چشمگیری را بر اندازه‌ی تأخیر صف از خود به جای می‌گذارند. به‌عنوان مثال: اگر تعداد N بسته به‌صورت هم‌زمان در فاصله‌های زمانی $(\frac{L}{R})N$ ثانیه وارد صف شوند. آنگاه نخستین بسته ارسالی دچار تأخیر صف نخواهد شد در حالیکه بسته‌ی دوم با تأخیر صف $\frac{L}{R}$ ثانیه و به طور کلی بسته‌ی n با تأخیر صف معادل $\left[(n-1)\frac{L}{R}\right]$ ثانیه رو به رو می‌شود. این بر عهده‌ی خواننده است که مقدار میانگین تأخیر صف را برای این مثال محاسبه نماید.

بدیهی است در عمل فرآیند ورود بسته‌ها به یک صف کاملاً تصادفی است، به عبارت دیگر بسته‌ها بدون رعایت الگوی ویژه و یا فواصل معین و حساب شده‌ای به صف وارد می‌شوند و حتی فواصل بین ورود آنها نیز تصادفی است. در این شرایط ارزش تعریف شده به‌وسیله‌ی عبارت $\frac{La}{R}$ نمی‌تواند تعریف درستی را از تأخیر صف به‌صورت آماری ارائه نماید. در هر حال درک تأثیر تأخیر صف باز هم دارای اهمیت می‌باشد به‌ویژه اگر شدت ترافیک نزدیک به صفر باشد، آنگاه تعداد بسته‌های ورودی اندک و فاصله زمانی ورودی بین هر کدام نیز طولانی است و در نتیجه احتمال آنکه بسته‌ی ورودی با بسته دیگری در صف رو به رو گردد تقریباً غیر ممکن است. بنابراین میانگین تأخیر صف نزدیک به صفر است. از طرف دیگر هنگامی که شدت ترافیک نزدیک به یک باشد، سرعت ورود بسته‌ها بیش از سرعت انتقال خواهد شد، زیرا نرخ ورود بسته‌ها به صف متغیر است، در نتیجه فاصله‌های زمانی ایجاد شده بین ورود بسته‌ها سبب تشکیل صف می‌شود. در این شرایط اگر نرخ ورود بسته‌ها کمتر از سرعت انتقال گردد، طول صف نیز کاهش می‌یابد. هنگامی که شدت ترافیک به سمت یک میل نماید، میانگین طول صف هم بزرگتر و بزرگتر می‌شود. وابستگی کیفی میانگین تأخیر صف بر شدت ترافیک به‌وسیله‌ی شکل ۱- ۱۸ نشان داده شده است.

1- Traffic intensity



شکل ۱- ۱۸ ♦ وابستگی کیفی میانگین تأخیر صف بر شدت ترافیک

یکی از برجسته‌ترین نکات شکل ۱- ۱۸ آن است که به هنگام نزدیک شدن شدت ترافیک به یک میانگین، تأخیر صف نیز به سرعت افزایش می‌یابد. درصد ناچیزی در افزایش شدت سبب افزایش تأخیر با درصد بسیار بالا خواهد شد. بسیاری از افراد این واقعیت را در بزرگراه‌ها تجربه نموده‌اند. اگر مسیر حرکت روزانه‌ی یک نفر از طریق یک بزرگراه دایماً از لحاظ ترافیک متراکم باشد مفهوم آن نزدیک شدن اندازه‌ی شدت ترافیک به ارزش یک می‌باشد. حالا اگر وقوع رویدادی سبب شود که میزان ترافیک از حدّ معمول خود به اندازه‌ی درصد اندکی رشد حاصل نماید، آنگاه تأخیر تجربه شده به‌وسیله‌ی رانندگان در این بزرگراه بسیار چشمگیر خواهد شد. به طور کلی اگر نرخ ورود بسته‌ها به اندازه‌ی کافی به گونه‌ای افزایش یابد که شدت ترافیک به یک نزدیک شود، آنگاه ملاحظه می‌شود که با گذشت زمان به تدریج صف تشکیل می‌شود.

آنچه در بالا توضیح داده شد با تصور این واقعیت است که تعداد بسته‌های ذخیره شده در صف بی نهایت می‌باشد اما در عمل ظرفیت یک صف، قبل از پیوند محدود بوده و بستگی به قیمت و ساختار فیزیکی مسیریاب استفاده شده در شبکه دارد. از آنجائیکه ظرفیت صف محدود می‌باشد، تأخیرهای بسته هنگامی که شدت ترافیک به سمت یک نزدیک شود، برخلاف تصور محدود می‌شود و در نتیجه یک بسته‌ی ورودی با یک صف کاملاً پُر رو به رو می‌شود. از آنجا که مکانی جهت ذخیره‌ی چنین بسته‌ای یافت نمی‌شود در نتیجه یک مسیریاب، بسته را حذف می‌کند به عبارت دیگر **اتلاف بسته**[1] حاصل می‌شود. این سرریزی در یک صف از نقطه‌نظر یک سیستم انتهایی (میزبان) شبیه به وارد شدن بسته به هسته‌ی شبکه است که هرگز در مقصد خود را نشان نمی‌دهد. رابطه‌ی بین مقدار اتلاف بسته‌ها و شدت ترافیک یک رابطه‌ی مستقیم است. بنابراین، کارایی یک گره در شبکه در اغلب موارد فقط محدود به اندازه گیری مقدار تأخیر صف نمی‌شود بلکه احتمال اتلاف بسته نیز در محاسبه‌ی کلی دخالت داده می‌شود. بدیهی است همانگونه که در فصل‌های آتی نشان داده خواهد شد یک بسته‌ی تلف شده، مجدداً بین دو طرف ارسال و دریافت، انتقال داده می‌شود تا اطمینان از دریافت بسته به‌وسیله گیرنده از سوی فرستنده حاصل گردد.

1- Packet Loss

۱-۱۳ تأخیر دو نقطه‌ی انتهایی

تاکنون نقطه‌ی تمرکز بر موارد مربوط به تأخیره گره قرار داشت، به عبارت دیگر تـأخیری کـه تنهـا در یک مسیریاب ایجاد می‌شد. در ادامه‌ی این بخش از فصل مجموع تأخیر از مبداء تا مقـصد، مـورد نظـر است. برای درک آسان تر مطلب فرض کنید تعداد N -1 مسیریاب در بین مـسیر مبـداء و مقـصد قـرار گرفته‌اند با تصور به اینکه شـبکه دارای ترافیـک عـادی و غیـر متـراکم اسـت و در نتیجـه مـی‌تـوان از تأخیرهای صف صرف نظر نمود. اگر تأخیر پردازش در هر روتر و میزبان مبداء d_{proc} و سرعت خروجـی سیستم انتهایی مبداء و هر یک از مسیریاب‌ها نیز R بیت بر ثانیه باشد و نیز اگر تأخیر انتشار متعلق به هر پیوند نیز d_{prop} فرض گردد، آنگاه تأخیر انباشتی دو نقطه‌ی انتهایی (بین دو سیستم انتهایی مبـداء و مقصد) برابر است با:

$$d_{end\ to\ end} = N\ (d_{proc} + d_{trans} + d_{prop}) \qquad (۲-۱)$$

در رابطه‌ی (۱-۲) ارزش تأخیر انتقال برابر است با: $d_{trans}=\frac{L}{R}$ که در این رابطه L اندازه‌ی بسته بر مقیاس بیت می‌باشد. بدیهی است در شرایط واقعی به رابطه‌ی (۱-۲) بایستی تأثیر مربوط بـه میـانگین تأخیر صف در هر گره را نیز اضافه نمود. برای آنکـه بتـوان تـأخیر دو نقطـه‌ی انتهـایی در یـک شـبکه کامپیوتری را بهتر درک نمود، این مهم از طریق استفاده از برنامه‌ی **رهگیری مسیر**[1] انجام پذیر است. این برنامه‌ی ساده بر روی هر میزبان اینترنت قابلیت اجرایی دارد. هنگامی که کاربر نام میزبان مقصد را در سیستم انتهایی مبداء مشخص نماید آنگاه برنامه‌ی رهگیری مسیر از میزبان مبداء چنـدین بـسته‌ی ویژه را به سمت مقصد، از طریق عبور آنها از میان تعدادی مسیریاب گسیل می‌دارد. با گذر این بسته‌ها به سوی مقصد و هنگام وارد شدن یکی از این بسته‌های ویژه به یک روتر، آن مسیریاب پیام کوتـاهی را که حامل نام و آدرس آن روتر است به سوی سیستم انتهایی مبداء، بازگشت می‌دهد. بنابراین اگر تعداد مسیریاب‌های بین مبداء و مقصد N -1 باشد، آنگاه مبداء تعـداد N بـسته‌ی ویـژه را بـه شـبکه ارسـال می‌نماید که همراه هر بسته آدرس نهایی مقصد مورد نظر قرار گرفته است. لازم به توضیح اسـت کـه از سوی مبداء، تعداد N بسته‌ی ویژه به‌صورت 1 تا N شماره گذاری می‌شوند.

هنگامی که مسیریاب شماره‌ی n، بسته‌ی علامت گذاری شده‌ی n را دریافت می‌کند آن مـسیریاب بسته را به سمت مقصد ارسال نمی‌کند بلکه پیامی را به مبداء بازگشت می‌دهد.

زمانی که میزبان مقصد بسته‌ی N را دریافت نماید او نیز پیامی را به سمت مبداء گـسیل مـی‌دارد. در این هنگام مبداء، زمان سپری شده بین ارسال یک بسته و دریافت پیـام متعلـق بـه آن بـسته را بـه همراه نام و آدرس روتر و یا میزبان مقصد که آن پیام را بازگشت داده است، ثبت می‌کند. از طریق ایـن روش است که میزبان مبداء، جریان مسیر گزینش شده به‌وسیله‌ی بسته‌ها را از مبداء تا مقصد بازسازی می‌نماید و بنابراین مبداء می‌تواند تأخیرهای دوره‌ی رفـت و برگـشت پیـام یعنـی زمـان RTT را بـرای همه‌ی مسیریاب‌ها محاسبه نماید.

برنامه‌ی رهگیری مسیر فرآیند توضیح داده شده در بالا را سه مرتبه تکرار می‌کند و بنابراین مبـداء درواقع تعداد N × 3 بسته را به مقصد گسیل می‌نماید. سـند [RFC1393] جزئیـات برنامـه‌ی رهگیـری مسیر را به‌صورت کامل توصیف نموده است. در ادامه نمونه‌ای از خروجی یک برنامه رهگیـری مـسیری

1- Traceroute Program

نشان داده شده است. در این خروجی مسیر رهگیری شده از مبداء (دانشگاه umass) به مقصد (دانشگاه پلی تکنیک poly) می‌باشد. نتیجه‌ی خارج شده شامل تعداد 6 ستون است که نخستین ستون: ارزش n یعنی شماره مسیریاب در طول مسیر، ستون دوم: نام روتر، ستون سوم: آدرس آن روتر که دارای قالب xxx • xxx • xxx • xxx می‌باشد. سه ستون آخر: تأخیر RTT برای سه بار تکرار می‌باشد. اگر تعداد پیام‌های دریافت شده به‌وسیله‌ی مبداء کمتر از سه پیام از هر یک از مسیریاب‌ها باشد، مفهوم آن اتلاف بسته در شبکه است و بنابراین برنامه‌ی رهگیری مسیر ستاره‌ای (*) را به دنبال شماره‌ی مسیریاب وارد می‌نماید و تعداد RTT کمتر از سه برای آن مسیریاب گزارش می‌نماید.

```
1 cs-gw (128.119.240.254) 1.009 ms 0.899 ms 0.993 ms
2 128.119.3.154. (128.119.3.154)0.931 ms 0.441 ms 0.651 ms
3 border4-rt-gi-1-3.gw.umass.edu (128.119.2.194) 1.032 ms 0.484 ms
0.451 ms
4 acr1-ge-2-1-0. Boston.cw.net (208.172.51.129) 10.006 ms 8.150 ms
8.460 ms
5 agr4-loopack.NewYork.cw.net (206.24.194.104) 12.272 ms 14.344 ms
13.267 ms
6 acr2-loopack.NewYork.cw.net (206.24.194.62) 13.225 ms 12.292 ms
12.148 ms
7 pos10-2.core2.NewYork1.Level3.net (209.244.160.133) 12.218 ms
11.823 ms 11.793 ms
8 gige9-1-52.hsipaccess1.NewYork1.Level3.net (64.159.17.39) 13.081
ms 11.556 ms 13.297 ms
9 p0-o.polyu.bbnplanet.net (4.25.109.122) 12.716 ms 13.052 ms
12.786 ms
10 cis.poly.edu (128.238.32.126) 14.080 ms 13.035 ms 12.802 ms
```

در خروجی برنامه‌ی رهگیری مسیر که در بالا نشان داده شده است، تعداد 9 مسیریاب بین مبداء و مقصد وجود دارد. اغلب این مسیریاب‌ها دارای یک نام و همگی آنها حامل آدرس می‌باشند. به عنوان مثال مسیریاب شماره‌ی 3 دارای نام: border4 - rt - gi - 1 - 3. gw. umass. edu و آدرس: 128. 119. 2. 194 می‌باشد. اگر به اطلاعات برنامه‌ی رهگیری مسیر برای این مسیریاب توجه بیشتری شود ملاحظه می‌شود که در میان سه مرتبه گسیل پیام، زمان RTT برای مرتبه‌ی نخست 1.03 میلی ثانیه و برای دو مرتبه تکرار بعدی به ترتیب 0.48 و 0.45 میلی ثانیه بوده است. این زمان‌های RTT در درون خود شامل همه تأخیرهایی است که قبلاً توضیح داده شده است. در میان این تأخیرها از آنجایی که تأخیر صف متأثر از زمان است، بنابراین زمان تأخیر RTT مربوط به بسته‌ای همانند بسته‌ی شماره‌ی n که بسوی مسیریاب n گسیل شده است، اغلب اوقات طولانی تر از تأخیر RTT مربوط به بسته‌ی ارسالی شماره‌ی $n+1$ به مسیریاب $n+1$ می‌باشد. این رویداد را به وضوح می‌توان در مثال بالا مشاهده نمود. تأخیرهای مربوط به مسیریاب 6 بیشتر از تأخیرهای متعلق به روتر 7 است !

برای مشاهده بیشتر عملکرد برنامه‌ی رهگیری مسیر، خوانندگان به سامانه‌ی:

www.traceroute.org

هدایت می‌شوند. در این برنامه خوانندگان می‌توانند با گزینش یک مبداء و معرفی نام میزبان هر مقصد نتیجه‌ی رهگیری شده را مشاهده نمایند. از میان برنامه‌های آزاد در شبکه‌ی اینترنت برنامه‌ی

Ping Plotter نیز توصیه می‌گردد. علاوه بر زمان‌های تأخیر پردازش، انتقال و انتشار، زمان‌های تأخیر قابل توجه دیگری نیز می‌تواند در سیستم‌های انتهایی پدیدار گردد. از جمله‌ی این زمان‌ها به‌عنوان مثال در مودم می‌توان از زمان مدولاسیون و یا عملیات مربوط به عکس آن نام برد که از لحاظ مقیاس زمانی معادل ده‌ها میلی ثانیه می‌باشد. این زمان برای فناوری‌های دیگر همانند خطوط DSL، پیوندهای اترنت و مودم کابلی چندان قابل اهمیت نمی‌باشند. یک سیستم انتهایی که نیازمند انتقال یک بسته از طریق یک رسانه‌ی مشترک نظیر WiFi یا اترنت می‌باشد، ممکن است عمداً به دلیل اشتراک رسانه با دیگر میزبان‌ها به‌صورت بخشی از پروتکل خود تأخیری را به شبکه تحمیل نماید. فصل ۵ این موارد تأخیر را با جزئیات بیشتری بررسی می‌کند.

یکی از تأخیرهای مهم دیگر مربوط به تأخیر بسته‌بندی در رسانه می‌باشد که اغلب مواقع در کاربردهای انتقال صدا از طریق پروتکل اینترنت[1] (VOIP) روی می‌دهد. در VIOP طرف ارسال، نخست بایستی یک بسته را با اطلاعات مکالمه (صدا) به‌صورت رمز شده‌ی دیجیتالی پُر نماید و آنگاه بسته را به شبکه‌ی اینترنت گسیل نماید. زمان لازم برای پُر شدن بسته را اصطلاحاً **زمان تأخیر بسته‌بندی** رسانه نامیده‌اند. این زمان می‌تواند چشمگیر باشد و تأثیر قابل توجهی را نیز در کیفیت برداشت کاربر از یک مکالمه VIOP به همراه داشته باشد.

۱- ۱۴ بازدهی شبکه‌های کامپیوتری

علاوه بر اتلاف و تأخیر بسته‌یکی دیگر از مقیاس‌های مهم اندازه‌گیری کارایی در شبکه‌های کامپیوتری **بازدهی**[2] انتها به انتها در شبکه است. برای آنکه بازدهی را بتوان تعریف نمود، حالتی را در نظر بگیرید که اطلاعات متعلق به یک فایل بزرگ از میزبان A به میزبان B از طریق شبکه بایستی منتقل گردد.

این انتقال می‌تواند در قالب یک فایل تصویری به‌صورت اشتراک فایل P2P باشد. بازدهی لحظه‌ای در هر ذره‌ی زمانی عبارتست از سرعتی (bps) که میزبان B اطلاعات فایل را دریافت می‌کند.

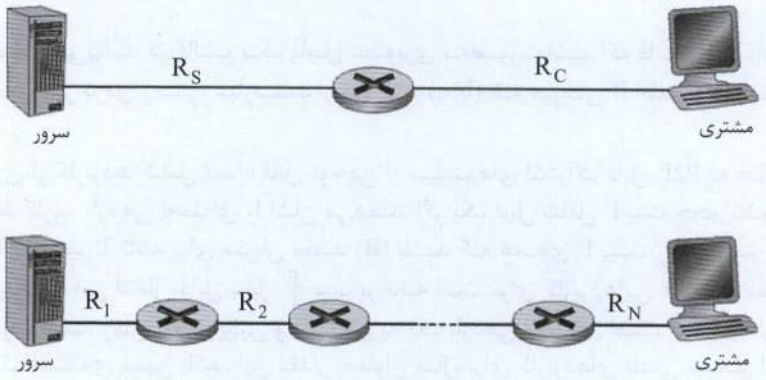
بسیاری از کاربردها شامل تعداد قابل توجهی از سیستم‌های اشتراک فایل P2P به هنگام دانلود در مدار واسط کاربر، بازدهی لحظه‌ای را نشان می‌دهند. اگر یک فایل شامل F بیت حجم باشد و زمان لازم برای انتقال آن نیز T ثانیه برای میزبان مقصد (B) باشد که همه‌ی F بیت را دریافت نماید، آنگاه میانگین بازدهی این انتقال فایل برابر $\frac{F}{T}$ بیت بر ثانیه است. برای کاربردهایی همانند تلفن اینترنتی آنچه مطلوب است، زمان تأخیر پائین و دستیابی به یک بازدهی لحظه‌ای است، که مقدار آن بایستی بیش از یک آستانه‌ی معین باشد. این مقدار به‌عنوان مثال برای کاربردهای تلفنی بیشتر از 24Kbps و برای کاربردهای تصویری زمان واقعی نیز بیش از 256Kbps است. برای کاربردهای دیگر نظیر انتقال فایل این تأخیر چندان حیاتی نمی‌باشد، اگرچه دستیابی به بالاترین بازدهی ممکن، هدف مطلوب است.

برای روشن شدن بیشتر اهمیت اصل بازدهی در شبکه‌های کامپیوتری در ادامه چند مثال دیگر بررسی می‌گردد. در شکل ۱- ۱۹ بخش (الف) دو سیستم انتهایی نشان داده شده است. یکی سرویس‌دهنده و دیگری مشتری. این دو از طریق دو پیوند ارتباطی و یک مسیریاب به یکدیگر متصل

1- Voice over internet protocol
2- Network Throughput

شده‌اند. اگر بازدهی انتقال یک فایل از سرور به مشتری مورد نظر باشد و اگر فرض گردد R_S سرعت پیوند بین سرور و مسیریاب و R_C سرعت پیوند بین مسیریاب و مشتری باشد و باز هم اگر تصور شود تنها بیت‌های ارسالی در سراسر شبکه متعلق به سرور می‌باشد، آنگاه پاسخ این سؤال که بازدهی این شبکه‌ی مطلوب سرویس‌دهنده و سرویس‌گیرنده چقدر است؟ اینگونه توصیف می‌گردد:

اگر جریان بیت‌ها همانند مایع در جریان از درون یک خط لوله که معادل پیوندهای ارتباطی است تصور شود، روشن است که سرور توان ارسال بیت‌ها از درون پیوند ارتباطی خود را بیش از R_S ندارد و مسیریاب نیز نمی‌تواند بیت‌ها را با سرعت بیش از R_C به سمت جلو هدایت نماید. بنابراین اگر $R_S < R_C$ باشد آنگاه بیت‌های ارسالی از سوی سرور به راحتی در درون مسیریاب جاری شده و سپس با سرعت R_S بیت بر ثانیه به مشتری تحویل می‌گردد و در نتیجه بازدهی برابر R_S بیت بر ثانیه است. از طرف دیگر اگر $R_C < R_S$ باشد آنگاه روتر نمی‌تواند سرعت ارسال بیت‌ها را بیش از سرعت آنها دریافت نماید، در نتیجه سرعت خروج بیت از مسیریاب برابر R_C بیت بر ثانیه خواهد بود. به عبارت دیگر بازدهی دو انتهای ارسال و دریافت R_C بیت بر ثانیه است. بدیهی است اگر بیت وارد شده در مسیریاب با سرعت R_S باشد و سرعت خروج آنها R_C بیت بر ثانیه آنگاه انباشت بیت‌های منتظر در مسیریاب ممکن است تا اندازه‌ای رشد نماید که وضعیت نامطلوبی را فراهم آورد. بنابراین برای این شبکه‌ی ساده که شامل دو پیوند است اندازه‌ی بازدهی برابر است با: $\min\{R_S \text{ و } R_C\}$ به عبارت دیگر سرعت انتقال در **محدوده‌ی تنگراه**[1] پیوند قرار می‌گیرد. با روشن شدن وضعیت بازدهی یک شبکه، آنگاه می‌توان به گونه‌ای تقریبی زمان لازم برای انتقال یک فایل پر حجم F بیت را از سوی یک سرور به سمت یک مشتری به‌صورت رابطه‌ی: $\left(F/\min\{R_s,R_c\}\right)$ تعریف نمود.



شکل ۱- ۱۹ ♦ بازدهی انتقال یک فایل از سرور به مشتری

مثال:

اگر در یک شبکه یک فایل MP3 به حجم 32 میلیون بیت از طریق یک سرور با پیوند 2 مگابیت در ثانیه در حال دریافت باشد در صورتی‌که سرعت پیوند دستیابی 1 مگا بیت بر ثانیه باشد. آنگاه زمان

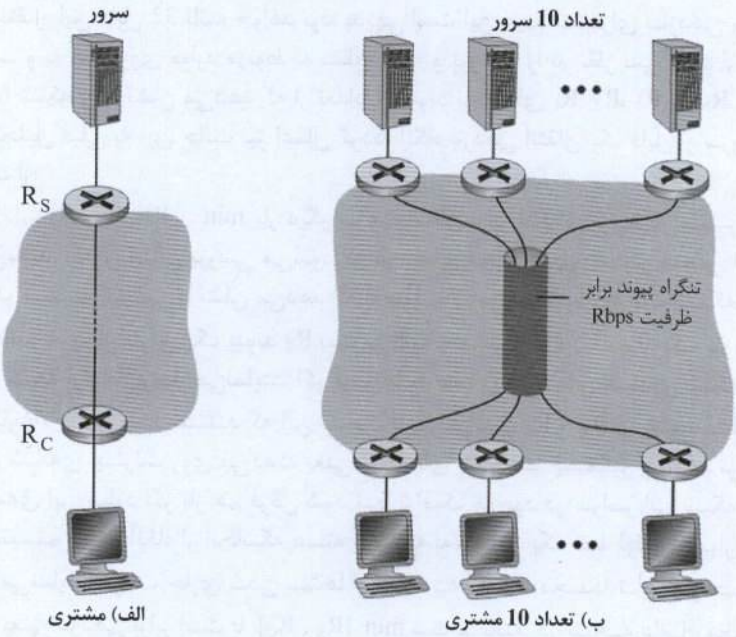
1- Bottleneck

لازم برای انتقال این فایل 32 ثانیه خواهد بود. بدیهی است این عبارات برای بازدهی و زمان انتقال تقریبی است و به هیچ روی موارد مربوط به سطح بسته و پروتکل را در نظر نمی‌گیرد. از طرفی شکل ۱- ۱۹ (ب) شبکه‌ای را نشان می‌دهد که از تعداد N پیوند با نام‌های R_1، R_2، R_3 ، ... ،R_N. تشکیل شده است. اگر تحلیل قبلی به این حالت نیز اعمال گردد، آنگاه بازدهی انتقال یک فایل از سرور به مشتری عبارت است از:

$\min \{ R_1 , R_2 , R_3 , ... , R_N \}$ بار دیگر سرعت انتقال برابر تنگراه پیوند در مسیر بین سرور و مشتری می‌باشد. اکنون مثالی بررسی می‌شود که در دنیای مجازی فعلی بیشتر متجلّی است. شکل ۱- ۲۰ (الف) دو سیستم انتهایی را نشان می‌دهد که به‌صورت سرور و مشتری به یک شبکه‌ی کامپیوتری متصل شده‌اند. سرور از طریق یک پیوند R_S بیت بر ثانیه و مشتری با استفاده از یک پیوند R_C بیت بر ثانیه با این شبکه ارتباط برقرار می‌نمایند. اگر فرض شود همه‌ی پیوندهای هسته‌ی شبکه‌ی ارتباطی از سرعت بسیار بالایی بهره‌مند هستند که این سرعت بسیار فراتر از R_S و R_C می‌باشد، مشابه آنچه امروز در هسته‌ی شبکه‌ی اینترنت روی می‌دهد، یعنی پیوندهای با سرعت چشمگیر و تراکم ترافیک ناچیز و علاوه بر همه‌ی این موارد اگر باز هم فرض شود تنها ترافیک موجود در سراسر این شبکه فقط منحصر به این دو سیستم باشد، آنگاه از آنجائیکه هسته‌ی شبکه به گونه‌ی یک خط لوله‌ی بسیار عریض در این مثال عمل می‌نماید، سرعت جاری شدن بیت‌ها از مبداء به مقصد مجدداً تابعی از حداقل R_S و R_C خواهد شد یعنی بازدهی برابر است با $\min \{R_S , R_C\}$ بیت بر ثانیه. در نتیجه عامل محدودیت برای بازدهی در اینترنت امروز، تنها شبکه‌ی دستیابی آن می‌باشد. سرانجام به‌عنوان آخرین مثال شکل ۱- ۲۰ (ب) را در نظر بگیرید.

در این شکل تعداد ده سرویس‌دهنده و ده مشتری به هسته‌ی یک شبکه کامپیوتری متصل شده‌اند. در این مثال تعداد 10 دانلود هم‌زمان از سوی تعداد 10 سرور و مشتری در حال انجام است. اگر فرض شود که تنها ترافیک شبکه در زمان جاری متعلق به این 10 دانلود باشد. بنابر آنچه در شکل نشان داده شده است، همه‌ی آنها در حال استفاده از یک پیوند می‌باشند. اگر R سرعت انتقال این پیوند باشد و اگر پیوندهای دستیابی همه‌ی سرورها و همه‌ی مشتری‌ها به ترتیب نیز سرعت مشابه R_S و R_C بیت بر ثانیه را دارا باشند، در صورتی که سرعت همه‌ی پیوندها در هسته به جز پیوند مشترک دارای سرعت R بیت بر ثانیه، سرعت‌هایی به مراتب بالاتر از R_S، R_C و R داشته باشند، بازدهی عملیات دانلود در این شرایط چنین بیان می‌شود.

اگر سرعت پیوند مشترک R صدها برابر R_S و R_C باشد آنگاه بازدهی هر یک از عملیات دانلود برابر حداقل R_S و R_C می‌باشد. اما اگر سرعت پیوند مشترک به اندازه‌ی سرعت R_S و R_C باشد، آنگاه بازدهی تفاوت چشمگیری را نسبت به حالت قبل پیدا می‌کند. به‌عنوان مثال اگر فرض شود R_S و R_C و R به ترتیب هر کدام دارای سرعت‌های 2 و 1 و 5 مگا بیت بر ثانیه باشند و سرعت پیوند مشترک R نیز به تساوی بین 10 عملیات دانلود توزیع شود، آنگاه تنگراهِ هر یک از دانلودها دیگر سرعت شبکه‌ی دستیابی نخواهد بود بلکه تنگراه پیوند مشترک مورد استفاده‌ی آنها در هسته‌ی شبکه می‌باشد. در این مورد هر یک از دانلودها دارای بازدهی 500 کیلو بیت بر ثانیه خواهند شد. بنابراین بازدهی انتها به انتها برای هر یک به 500 Kbps محدود خواهد شد.



شکل ۱- ۲۰ ♦ بازدهی دو نقطه‌ی انتهایی: (الف) دریافت فایل از سرور به‌وسیله‌ی مشتری، ب) دریافت از سوی 10 مشتری با تعداد 10 سرور

مثال‌های عنوان شده در دو شکل ۱- ۱۹ و ۱- ۲۰ (الف) وابستگی بازدهی را به سرعت انتقال پیوندهایی که داده‌ها بر روی آنها در جریان می‌باشد نشان می‌دهد. ملاحظه گردید هنگامی که ترافیک مداخله گر دیگری در شبکه وجود ندارد بازدهی به شکل ساده‌ی خود به‌صورت تقریبی برابر است با حداقل سرعت انتقال در طول مسیر بین مبداء و مقصد.

مثال شکل ۱- ۲۰ (ب) نشان می‌دهد، بازدهی در حالت کلی آن فقط وابسته به سرعت انتقال پیوندهای طول مسیر نیست بلکه، وابستگی به ترافیک مداخله‌گر نیز پیدا می‌کند. به‌ویژه یک پیوند با سرعت بسیار بالا ممکن است پیوند تنگراه برای یک انتقال فایل نباشد، البته در صورتی که تعداد زیاد دیگری جریان داده از درون آن پیوند در حال عبور باشند.

۱- ۱۵ لایه‌های پروتکلی و مدل‌های خدمات آنها

با توجه به مطالبی که تاکنون بررسی گردیده است، این نکته آشکار می‌شود که شبکه‌ی اینترنت یک سیستم فوق‌العاده پیچیده‌ای است. عناصر بسیار متعددی در تشکیل و گسترش این شبکه دخالت دارند. تعداد بی شماری برنامه‌های کاربردی و پروتکل‌ها، انواع مختلف سیستم‌های انتهایی، راهیاب‌های بسته و تنوع گونه‌های متفاوت از رسانه‌ها در سطوح مختلف از جمله این موارد می‌باشند. با توجه به این پیچیدگی‌های گسترده آیا می‌توان به سازماندهی یک معماری شبکه و یا حداقل به شفاف نمودن معماری این شبکه امیدوار بود؟

هدف اصلی این بخش پاسخ به این سؤال اساسی است. لازم است قبل از پرداختن به معماری شبکه‌ی اینترنت، تصوری از آنچه که هر روز در اطراف ما می‌گذرد داشته باشیم. به‌عنوان نمونه، اگر از

فردی خواسته شود سیستم حمل و نقل هوایی را توصیف نماید. چگونه می‌توان ساختاری را برای این سیستم پیچیده تعریف نمود که شامل:

نمایندگی‌های دفاتر فروش بلیط، کنترل‌کننده‌های بار و بلیط، مسافران، کارکنان سالن‌های ورود و خروج مسافران، خلبانان، هواپیماها، کارکنان بخش کنترل ترافیک هوایی و یک سیستم مسیریابی خطوط هوایی به مقیاس تمامی کره‌ی زمین، شود.

شاید ساده‌ترین روش برای توصیف این سیستم تقسیم‌بندی و تعریف مجموعه‌ی عملیاتی است که یک فرد هنگام پرواز با یک خط هوایی انجام می‌دهد. این مراحل عبارتند از: خرید بلیط، تحویل بار، رفتن به سالن انتظار پروازهای خروجی و سرانجام سوار شدن به هواپیما.

هواپیما پس از پرواز از طریق فرآیندهای مسیریابی به مقصد خود هدایت می‌شود و پس از فرود در فرودگاه مقصد، مراحل پیاده شدن از هواپیما، تحویل بار از سالن پروازهای ورودی و سپس خروج نهایی از محوطه‌ی فرودگاه انجام می‌شود. اگر احتمالاً خدمات پرواز برای مسافری ناخوشایند باشد آنگاه وی از تجربه‌ی تلخ خود به دفتر فروش شکایت می‌برد. بدیهی است برای این تلاش ثمره‌ای در آخر حاصل نخواهد شد.

شکل ۱- ۲۱ همه‌ی این موارد را نشان می‌دهد.



شکل ۱- ۲۱ ♦ عملیات یک پرواز

شاید تا به حال پی به شباهت این مراحل با یک شبکه‌ی کامپیوتری برده باشید. در این شبکه نیز یک بسته‌ی اطلاعاتی از سمت مبداء به سوی میزبان مقصد در اینترنت گسیل می‌شود. با توجه به شکل ۱- ۲۱ ملاحظه می‌شود عملیات مربوط به فرآیند بلیط در هر یک از دو سوی شبکه وجود دارد. علاوه بر این موارد بایستی عملیات مربوط به فرآیندهای تحویل بار، ورود و خروج در سالن‌های انتظار و دیگر موارد را نیز اضافه نمود. برای مسافرینی که موفق به انجام این مراحل شده‌اند، بایستی مراحل پرواز، فرود و عملیات در طول پرواز شامل فرآیند مسیریابی پرواز را نیز اضافه نمود. توضیح این موارد به این مفهوم است که از لحاظ عملیاتی این مراحل را در شکل ۱- ۲۱ می‌توان به‌صورت یک رویکرد افقی در نظر گرفت. از طرفی شکل ۱- ۲۲ همین مراحل را در قالب یک چارچوب لایه‌بندی شده که می‌توان از آن به‌عنوان انجام یک سفر هوایی نام برد، نشان می‌دهد.

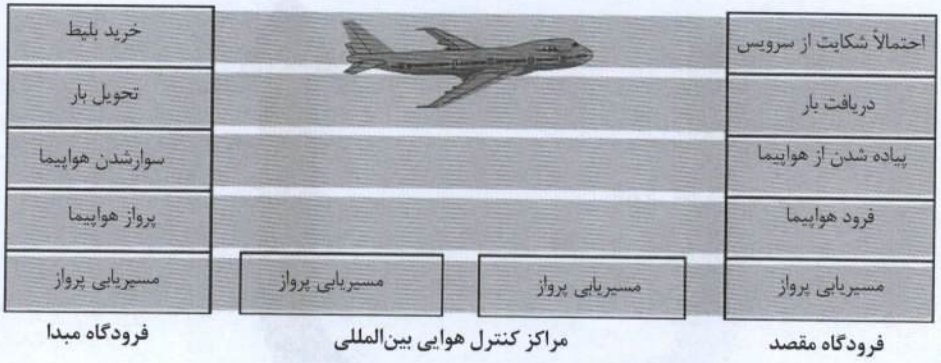
با توجه به شکل ۱- ۲۲ ملاحظه می‌شود که هر لایه با لایه‌های زیرین خود ترکیب شده و فرآیندی را سبب می‌شود که می‌تواند نوعی سرویس یا خدمات تلقی شود. در لایه‌ی صدور بلیط و لایه‌ی زیرین آن، پایانه‌ی صدور بلیط و دریافت کارت پرواز انجام می‌شود. در لایه‌ی تحویل بار و لایه‌ی زیرین آن، تحویل بار و سپس دریافت بار صورت می‌پذیرد.

بدیهی است خدمات این لایه فقط شامل آن دسته از مسافرانی است که مراحل صدور بلیط و دریافت کارت پرواز را انجام داده باشند. در لایه سالن انتظار، مراحل خروج و سوار شدن بر هواپیما و نیز فرود و خارج شدن از هواپیما برای آن دسته از مسافرانی که مراحل تحویل بار و دریافت را انجام داده‌اند صورت می‌پذیرد و سرانجام لایه‌ی پرواز و فرود انتقال از باند پرواز به باند فرود مسافران را در بر می‌گیرد که مراحل قبل را انجام داده‌اند.

بنابراین می‌توان این نتیجه‌گیری را نمود که هر لایه خدماتی را فراهم می‌آورد که عبارت است از:

(۱) انجام یک مجموعه عملیات در بین آن لایه به‌عنوان مثال لایه سالن ورود و خروج، سوار و پیاده نمودن مسافران را به هواپیما و یا از آن انجام می‌دهد.

(۲) با استفاده از خدمات لایه‌های پائین‌تر از خود به‌عنوان مثال: لایه باندهای فرود و خروج که لایه سالن ورود و خروج از آن استفاده می‌کند، خدمات مربوط به انتقال مسافران صورت می‌پذیرد.



شکل ۱- ۲۲ ♦ مراحل عملیات فرآیند افقی لایه‌ها

یک معماری لایه‌بندی شده باعث می‌شود بتوان به توصیف بخش‌های ویژه از یک سیستم بزرگ و پیچیده پرداخت. این ساده‌سازی خود دارای ارزش قابل ملاحظه‌ای از نقطه‌نظر توزیع عملیات، و در نتیجه آسان‌سازی اجرا و تغییر آن در یک پیاده‌سازی انجام شده، می‌باشد. این آسان‌سازی به‌ویژه در ارتباط با خدمات فراهم‌آوری شده به‌وسیله‌ی لایه‌ها بیشتر محسوس است. تا زمانی که یک لایه خدمات همانندی را برای لایه‌ی بالاتر از خود فراهم می‌آورد و از خدمات مشابه لایه‌های زیرین نیز بهره می‌برد، بخش‌های دیگر سیستم در صورت اعمال تغییر در پیاده‌سازی یک لایه بدون دگرگونی باقی می‌مانند. بدیهی است که تغییر در چگونگی پیاده‌سازی یک سرویس کاملاً از تغییرات اعمال شده در همان سرویس متفاوت می‌باشد. به‌عنوان مثال: اگر در عملیات سوار و پیاده شدن مسافران تغییری اعمال شود بطوریکه این عمل براساس اندازه‌ی قد افراد صورت پذیرد، بقیه مراحل سفر هوایی بدون تغییر به وظایف خود عمل خواهند نمود زیرا هنوز هم بخش مربوط به سوار و پیاده‌سازی مسافران وظیفه‌ی خود را انجام می‌دهد. تنها تفاوت، چگونگی فرآیند پیاده‌سازی خدمات ارائه شده به‌وسیله‌ی آن می‌باشد.

برای یک سیستم بزرگ و پیچیده که مرتباً در حال به روز شدن است، توانمندی در پیاده‌سازی

تغییرات اعمال شده در یک سرویس بدون تأثیرگذاری بر روی عناصر دیگر یک سیستم از ویژگی‌های مهم به‌شمار می‌آید.

با توجه به موارد اشاره شده اکنون می‌توان به بحث پیرامون لایه‌بندی پروتکل‌های شبکه و اعمال آن پرداخت. معمولاً برای طراحی ساختار یک شبکه از سازماندهی پروتکل‌ها در شبکه استفاده می‌شود. به عبارت دیگر نرم‌افزار و سخت‌افزار شبکه پروتکل‌ها را در درون لایه‌های خود پیاده‌سازی می‌کنند. پیاده‌سازی پروتکل‌ها بر این اصل پایه‌ریزی می‌شود که هر پروتکل در ارتباط با یک لایه باشد. به همان صورت که هر عملکردی در سفر هوایی شکل ۱- ۲۲ وابسته به یک لایه می‌شد. آنچه برای یک طراح شبکه اهمیت دارد خدماتی است که یک لایه در اختیار لایه‌ی بالاتر از خود قرار می‌دهد. این مورد آخر در اصطلاح واژگان شبکه **الگوی خدمات** یک لایه نام دارد.

همانگونه که از مثال سفر هوایی می‌توان دریافت هر لایه، خدماتی را فراهم می‌آورد که شامل: (۱) انجام عملیات ویژه‌ای است در بین آن لایه و (۲) استفاده از خدمات لایه‌ای که مستقیماً در زیر آن قرار گرفته است. به‌عنوان مثال خدمات فراهم شده به‌وسیله لایه‌ی n ممکن است شامل مواردی از قبیل تحویل مطمئن بسته‌ها و پیام‌ها از یک نبش شبکه به لبه‌ی دیگر شبکه باشد. این مورد می‌تواند پیام‌ها را با استفاده از خدمات تحویل غیرمطمئن لبه به لبه‌ی شبکه که لایه‌ی n -1 فراهم می‌آورد، جابجا نماید. بدیهی است در این مثال با اضافه نمودن عملیات لایه n به این خدمات، مراحل تشخیص و انتقال مجدد یک پیام تلف شده انجام می‌گیرد. یک لایه‌ی پروتکلی می‌تواند در هر دو مورد نرم‌افزاری و سخت‌افزاری و یا ترکیبی از هر دو اعمال گردد.

پروتکل‌های لایه‌بندی کاربردها همانند HTTP و یا SMTP تقریباً همیشه در نرم‌افزارهای سیستم‌های انتهایی پیاده‌سازی می‌شوند. پروتکل‌های لایه‌ی انتقال نیز وضعیت مشابهی را دارند. از آنجائیکه لایه‌های فیزیکی و پیوند داده مسئولیت عملیات ارتباطی بر روی یک پیوند ویژه را بر عهده دارند، آنها در درون یک کارت شبکه همانند کارت‌های اترنت و یا WiFi پیاده‌سازی می‌شوند. پروتکل‌های لایه شبکه در اغلب موارد به‌صورت ترکیب سخت و نرم‌افزاری پیاده‌سازی می‌شوند.

بدیهی است همانگونه که عملیات در معماری لایه‌بندی شده‌ی خطوط هوایی در بین فرودگاه‌های مختلف و مراکز کنترل پرواز که سیستم را تشکیل می‌دهند، توزیع شده است، یک لایه‌ی پروتکلی n نیز در بین سیستم‌های انتهایی، راهیاب‌های بسته و دیگر عناصری که تشکیل یک شبکه را می‌دهند توزیع گردیده است. به عبارت دیگر غالباً قطعه‌ای از یک لایه‌ی پروتکل n در هر یک از عناصر این شبکه نیز وجود دارد. ساختار لایه‌بندی پروتکلی شبکه، نقاط قوت قابل توجهی را به همراه دارد از جمله سادگی اعمال تغییرات، توانمندی در گسترش شبکه، انعطاف‌پذیری در طراحی و تولید قطعات و محصولات سخت و نرم‌افزاری، سهولت در اعمال به روز رسانی قطعات و فناوری شبکه و ده‌ها مورد دیگر که می‌توان عنوان نمود. در میان این نقاط قوت در هر حال می‌توان ضعف‌هایی را نیز برای این رویکرد شناسایی نمود. به‌عنوان مثال یک نقطه ضعف را می‌توان دوباره کاری در عملیات لایه‌های زیرین به‌وسیله‌ی یک لایه دانست.

نمونه‌ای از این مورد که در اغلب **پشته‌های پروتکلی**[1] اتفاق می‌افتد، مربوط به فرآیند بازیابی خطا می‌باشد. این فرآیند هم در لایه‌ی پیوند داده‌ها براساس هر پیوند و هم در سیستم‌های انتهایی واقع می‌شود. نقطه ضعف دیگر آن است که عملکردی در یک لایه ممکن است نیازمند اطلاعاتی باشد

1- Protocol Stack

که در لایه‌ی دیگر وجود دارد. این مورد آخر منظور و هدف جداسازی لایه‌ها را نقض می‌کند. هنگامی که هر دو در نظر گرفته شوند، فناوری لایه‌بندی پروتکلی از جوانب مختلف مطلوب‌تر به نظر می‌رسد.

زمانی که پروتکل‌های لایه‌های مختلف به‌صورت جمعی در نظر گرفته شوند، مجموعه‌ی آنها اصطلاحاً **پشته‌ی پروتکلی** نام دارد. مجموعه‌ی پروتکلی اینترنت شامل پنج لایه می‌باشد که عبارتند از: لایه‌های فیزیکی، پیوند، شبکه، انتقال و کاربرد.

شکل ۱- ۲۳ (الف) این لایه‌ها را نشان می‌دهد. فصل‌های این کتاب براساس این پنج لایه پایه‌ریزی شده‌اند که از آن به‌عنوان پشته پروتکلی اینترنت نام برده شده است و رویکرد آن به‌صورت از فراز به فرود می‌باشد. این رویکرد به مفهوم آن است که نخست لایه‌ی کاربرد و به دنبال آنها لایه‌های دیگر بررسی شده‌اند.



(الف) 5 لایه‌ی پشته‌ی پروتکلی اینترنت (ب) 7 لایه‌ی مدل مرجع ISO/OSI

شکل ۱- ۲۳ ♦ پشته پروتکلی اینترنت و پشته مدل مرجع **OSI**

• لایه‌ی کاربرد

لایه‌ی کاربرد جایگاهی است که در آن نرم‌افزارهای کاربردی شبکه و پروتکل‌های مربوط به آن جای دارند. لایه‌ی کاربرد اینترنت شامل بسیاری از پروتکل‌ها همانند HTTP برای واکشی و تحویل اسناد و صفحه‌های وب، SMTP برای انتقال پیام‌های الکترونیک و FTP جهت انتقال اطلاعات فایل‌ها بین سیستم‌های انتهایی می‌باشند. علاوه بر این موارد فرآیندهای دیگری نظیر ترجمه‌ی یک نام متنی مانند www.ietf.org به یک آدرس 32 بیت نیز به کمک یک پروتکل لایه‌ی کاربردی ویژه که **DNS**[1] نام دارد، صورت می‌پذیرد.

در فصل ۲ طراحی و پیاده‌سازی یک پروتکل لایه‌ی کاربرد بررسی خواهد شد. معمولاً پروتکل لایه‌ی کاربرد بر روی چندین میزبان (سیستم انتهایی) توزیع می‌گردد که هر کاربردی در یک میزبان از آن پروتکل برای تبادل بسته‌های اطلاعاتی با کاربرد دیگر در یک سیستم انتهایی دیگر استفاده می‌نماید. بسته‌های اطلاعاتی در لایه کاربرد **پیام**[2] نامیده می‌شوند.

1- Domain Name Service
2- Message

- **لایه‌ی انتقال**

لایه‌ی انتقال اینترنت پیام‌های لایه‌ی کاربرد را بین کاربردهای انتهایی انتقال می‌دهد. در شبکه‌ی اینترنت دو پروتکل انتقال تعریف شده است. این دو عبارتند از:

TCP [1] و UDP [2] هر یک از آنها به تنهایی می‌تواند پیام‌های لایه‌ی کاربرد را انتقال دهد. پروتکل TCP سرویس‌های انتقال را به‌صورت اتصال‌گرا برای کاربردهای خود فراهم می‌آورد. این خدمات شامل ضمانت تحویل پیام‌های لایه‌ی کاربرد به مقصد و نیز کنترل جریان می‌باشد که عبارت از نگاشت سرعت ارسال و دریافت است. علاوه بر این موارد، پروتکل TCP پیام‌های طولانی را به قطعه‌های کوتاهتر نیز تبدیل می‌نماید و راه کارهای مربوط به کنترل تراکم را نیز اعمال می‌کند، به طوری که یک منبع بتواند سرعت خود را براساس نوع و شدت تراکم موجود در شبکه تنظیم نماید.

پروتکل UDP خدمات غیر اتصال‌گرا را برای کاربردهای خود فراهم می‌آورد. این خدمات عاری از ضمانت تحویل پیام‌های لایه کاربرد به مقصد، کنترل جریان یا کنترل تراکم شبکه می‌باشند. در این لایه بسته‌های لایه‌ی انتقال **قطعه** [3] نامیده می‌شوند.

- **لایه‌ی شبکه**

لایه‌ی شبکه در اینترنت عهده دار جابجایی بسته‌های لایه‌ی شبکه بین میزبان‌ها است که به آنها **داده گرام** [4] گفته می‌شود. پروتکل لایه‌ی انتقال اینترنت TCP و یا UDP در یک میزبان منبع قطعه‌های لایه انتقال و نیز آدرس مقصد را به لایه‌ی شبکه منتقل می‌کند. این مورد شبیه به تحویل یک پاکت نامه به همراه آدرس مقصد آن به شبکه‌ی خدمات پست می‌باشد.

لایه‌ی شبکه آنگاه خدمات تحویل قطعه‌ها به لایه‌ی انتقال در میزبان مقصد را فراهم می‌آورد. لایه‌ی شبکه در اینترنت، شامل پروتکل‌هایی از قبیل IP [5] است که به تعریف میدان‌هایی در داده گرام و نیز چگونگی رفتار سیستم‌های انتهایی و مسیریاب‌ها بر روی این میدان‌ها می‌پردازد. تنها یک پروتکل IP در این لایه تعریف شده است و همه‌ی عناصر اینترنت که حامل لایه‌ی شبکه هستند بایستی بر روی این پروتکل (IP) اجرا گردند.

لایه‌ی شبکه در اینترنت هم چنین شامل پروتکل‌هایی است که به محاسبه و شناسایی مسیرهایی که داده گرام‌ها بایستی بین منابع و مقاصد گزینش نمایند، می‌پردازند. شبکه‌ی اینترنت شامل پروتکل‌های مسیریابی بی‌شماری است. قبلاً اشاره گردید، اینترنت شبکه‌ای از شبکه‌ها می‌باشد و در بین یک شبکه، پیاده‌سازی و اجرای هر پروتکل مسیریابی که مطلوب باشد، امکان‌پذیر است. اگر چه لایه‌ی شبکه شامل هر دو پروتکل IP و پروتکل‌های بی‌شمار مسیریابی است، اما در اغلب موارد به‌صورت ساده تری آن را لایه‌ی IP می‌خوانند. این مطلب انعکاس این حقیقت است که IP همانند گیره‌ای است که عناصر اینترنت را به یکدیگر نگاه می‌دارد.

- **لایه‌ی پیوند**

لایه‌ی شبکه در اینترنت مسیریابی داده گرام‌ها از درون مجموعه‌ای از روترهای بین منابع و مقاصد را

1- Transport Control Protocol
2- User datagram protocol
3- Segment
4- Datagram
5- Internet Protocol

برعهده دارد. برای جابجایی یک بسته از یک گره (میزبان و یا مسیریاب) به گره‌ی بعدی در یک مسیر، لایه‌ی شبکه وابسته به خدمات لایه‌ی پیوند می‌باشد. در هر گره لایه‌ی شبکه داده گرام‌ها را به سمت لایه‌ی پیوند هدایت نموده و سپس همین لایه‌ی داده گرام‌ها را به گره‌ی بعدی در مسیر تحویل می‌نماید. در گره‌ی بعدی لایه‌ی پیوند شبکه داده گرام‌ها را به لایه‌ی شبکه تسلیم می‌نماید. خدمات فراهم‌آوری شده به‌وسیله‌ی لایه‌ی پیوند وابسته به پروتکل ویژه‌ای می‌باشد که آن پیوند به کار گرفته است. به‌عنوان مثال بعضی از پروتکل‌های لایه‌ی پیوند شرایط تحویل قابل اطمینان را از گره‌ی ارسال‌کننده بر روی یک پیوند به گره‌ی دریافت‌کننده بر روی پیوند دیگر فراهم می‌آورند.

بدیهی است این خدمات تحویل قابل اطمینان متفاوت از خدمات تحویل مطمئن TCP است که در آن تحویل قابل اطمینان از یک میزبان به سیستم انتهایی دیگر فراهم می‌گردید.

نمونه هایی از پروتکل‌های پیوند عبارتند از: اترنت، WiFi و پروتکل نقطه به نقطه (PPP).

از آنجائیکه داده گرام‌ها نیازمند عبور از چندین پیوند جهت پیمایش مسیر خود از مبداء تا مقصد می‌باشند، یک داده گرام ممکن است با چندین پروتکل مختلف لایه‌ی پیوند موجود در مسیر رو به رو گردد. به‌عنوان مثال یک داده گرام ممکن است بر روی یک پیوند به‌وسیله پروتکل اترنت پردازش شود، در حالیکه بر روی پیوند بعدی از طریق پروتکل PPP پردازش گردد. بسته‌های مربوط به لایه‌ی پیوند شبکه **قاب** یا **فریم**[1] نامیده می‌شوند.

• لایه‌ی فیزیکی

در حالی‌که وظیفه‌ی اصلی لایه‌ی پیوند در اینترنت جابه ایی همه‌ی فریم‌ها از یک عنصر شبکه به عنصر دیگری در شبکه‌ی مجاور است. وظیفه‌ی لایه فیزیکی جابه‌جایی تک تک بیت‌های درون فریم‌ها از یک گره به گره‌ی بعدی است. پروتکل‌های این لایه نیز وابسته به پیوندها می‌باشند و علاوه بر همه‌ی این موارد وابستگی شدیدی نیز به نوع رسانه‌ی انتقال به کار گرفته شده دارند. به‌عنوان مثال: پروتکل‌های لایه‌ی فیزیکی اترنت شامل پروتکل‌های مربوط به رشته‌های مسی زوج به هم تابیده، کابل‌های هم‌محور، الیاف نوری و غیره می‌تواند قلمداد گردد. در هر یک از این موارد تنها یک بیت به صورت‌های مختلف از طرف یک پیوند انتقال می‌یابد.

• مدل OSI

در اینجا لازم است توضیحات هر چند مختصری در ارتباط با مدل مرجع **OSI**[2] ارائه گردد. این مدل در دهه‌ی هفتاد میلادی به‌وسیله سازمان جهانی استاندارد (ISO) پیشنهاد گردید. سازماندهی این مدل در اطراف هفت لایه می‌باشد. شکل ۱- ۲۳ (ب) هفت لایه‌ی مدل مرجع را نشان می‌دهد. این لایه‌ها عبارتند از: کاربرد، ارائه، اجلاس، انتقال، شبکه، پیوند داده و فیزیکی. عملکرد پنج لایه‌ی این الگو شباهت بسیار زیادی با لایه‌های نامبرده در اینترنت دارد. دو لایه‌ی اضافه در این مدل شامل لایه‌های ارائه و اجلاس می‌باشند. نقش لایه‌ی ارائه (لایه‌ی ششم)، فراهم‌آوری خدماتی است که از طریق آنها همه‌ی کاربردهای ارتباطی قادر به تفسیر مفهوم داده‌های در حال تبادل با یکدیگر باشند.

این خدمات شامل فشرده‌سازی داده و رمزگذاری داده می‌باشد. علاوه بر آن خدمات مربوط به آنچه **توصیف داده‌ها** نامیده شده است، از جمله‌ی وظایف این لایه است. این مورد به مفهوم آزادسازی کاربردها از نقطه‌نظر تصمیم‌گیری در رابطه با قالب داخلی که داده‌ها بایستی بیان و یا ذخیره گردند،

1- Frames
2- Open System Interconnection

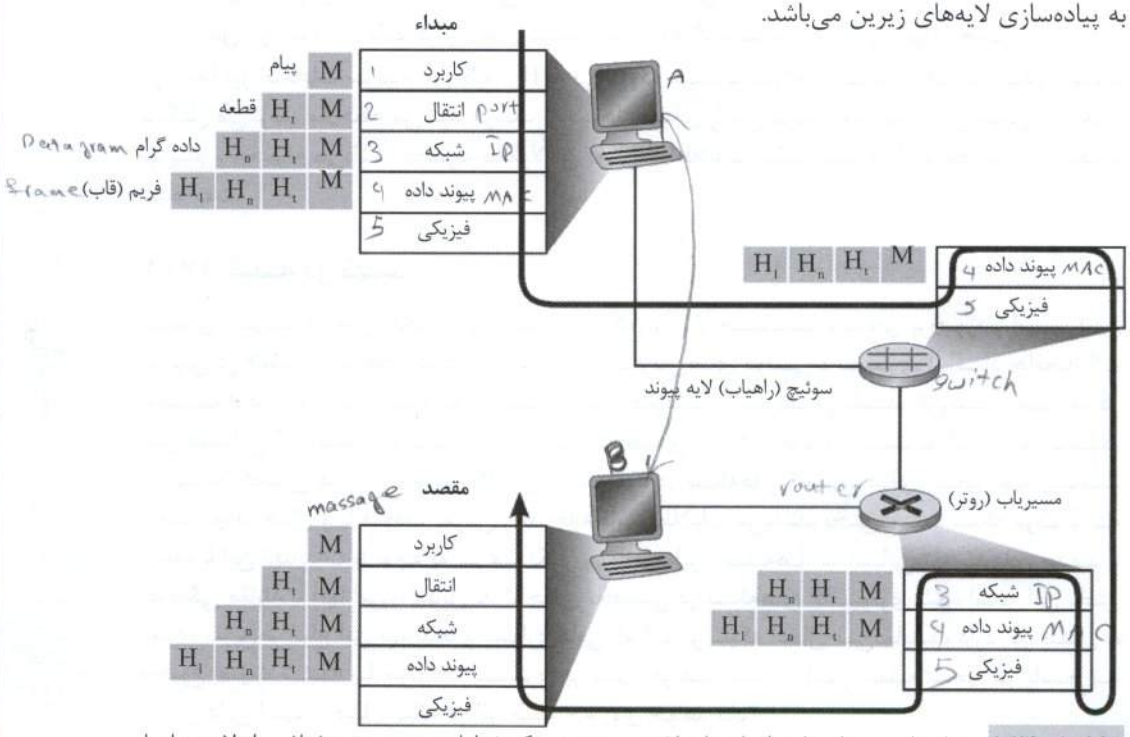
می‌شود، قالب‌هایی که ممکن است از یک کامپیوتر به کامپیوتر دیگر متفاوت باشد. لایه‌ی اجلاس، خدمات مربوط به هم‌زمانی و عدم محدودیت مبادله‌ی داده‌ها، شامل راه کارهای بازیابی و نقاط آزمایش داده‌ها را تعریف می‌کند.

علت آنکه اینترنت عاری از این دو لایه‌ی مدل مرجع می‌باشد، ممکن است پرسش‌هایی را در ذهن ایجاد نماید. از جمله‌ی این پرسش‌ها آن است که آیا خدمات فراهم شده به‌وسیله‌ی این لایه‌ها دارای اهمیت نیستند و یا آنکه اگر یک کاربرد، نیازمند یکی از این خدمات باشد، آنگاه چه باید کرد؟

اینترنت پاسخ هر دو سؤال را اینگونه طرح می‌کند: اهمیت یک سرویس بستگی به طراح آن برنامه کاربردی دارد و اگر آن سرویس دارای اهمیت است این وظیفه‌ی طراح است که امکانات پیاده‌سازی آن را در برنامه کاربردی فراهم آورد و آن را به شکل کامل پیاده‌سازی نماید.

۱- ۱۶ پیام‌ها، قطعه‌ها، داده گرام‌ها و قاب‌ها

شکل ۱- ۲۴ مسیر فیزیکی یک داده را به هنگام عبور از پشته‌ی پروتکل‌های میزبان مبداء و به سمت بالا و یا پائین مجموعه پروتکل‌های لایه‌ی پیوند راهیاب و مسیریاب و سپس حرکت به طرف بالا در مجموعه‌ی پروتکلی میزبان مقصد نشان می‌دهد. در فصل آتی توضیح داده شده است که مسیریاب‌ها و راهیاب‌های لایه‌ی پیوند هر دو از لحاظ عملکرد به‌صورت راهیاب‌های بسته می‌باشند. علاوه بر این مطلب روترها و راهیاب‌های لایه‌ی پیوند از لحاظ سازماندهی سخت و نرم‌افزاری همانند میزبان‌ها از یک رویکرد لایه‌ای استفاده می‌کنند. با این تفاوت که آنها همه‌ی لایه‌های موجود در پشته‌ی پروتکل‌ها را در دورن خود اعمال نمی‌کنند. آنچه آنها به‌ویژه به‌عنوان پروتکل‌های مورد نیاز به‌کار می‌گیرند مربوط به پیاده‌سازی لایه‌های زیرین می‌باشد.



شکل ۱- ۲۴ ♦ میزبان‌ها، مسیریاب‌ها و راهیاب‌های لایه‌ی پیوند، هر یک شامل مجموعه‌ی مختلفی از لایه‌ها با عملکردهای متفاوت

همانگونه که شکل ۱- ۲۴ نشان می‌دهد، راهیاب‌های لایه‌ی پیوند فقط به پیاده‌سازی لایه‌های اول و دوم و مسیریاب‌ها به اعمال لایه‌های اول، دوم و سوم می‌پردازند. این به معنی آن است که راهیاب‌های اینترنت قادرند پروتکل IP را که یک پروتکل لایه‌ی سوم است پیاده‌سازی نمایند. در حالیکه، راهیاب‌های لایه‌ی پیوند از این امکان بی بهره‌اند.

در آینده ملاحظه می‌شود اگرچه راهیاب‌های لایه‌ی پیوند قادر به شناسایی آدرس‌های IP نمی‌باشند، اما آنها می‌توانند آدرس‌های مربوط به لایه دوم، نظیر آدرس‌های اترنت را تشخیص دهند. لازم به یادآوری است که میزبان‌ها تمام پنج لایه را پیاده‌سازی می‌کنند و بیان این واقعیت تأکید بر این دیدگاه است که معماری اینترنت بیشترین پیچیدگی خود را به لبه‌های شبکه انتقال می‌دهد. شکل ۱- ۲۴ اصل مهم **کپسول‌بندی**[1] را نیز نشان می‌دهد. میزبان مبداء پیام لایه‌ی کاربرد M را در شکل ۱- ۲۴ به لایه‌ی انتقال تحویل می‌دهد. در ساده‌ترین حالت لایه‌ی انتقال، پیام را با اضافه نمودن اطلاعات اضافه تر که به آن سر پیام لایه‌ی انتقال (H_t) گفته می‌شود، تکمیل می‌نماید. لایه‌ی انتقال در سمت دریافت از H_t استفاده می‌نماید. این مجموعه‌ی شامل M و H_t **قطعه‌ی لایه‌ی انتقال** نامیده می‌شود. قطعه‌ی لایه‌ی انتقال بنا بر این پیام M را کپسول‌بندی می‌نماید.

اطلاعات اضافه شده ممکن است این امکان را برای لایه‌ی انتقال در سمت دریافت فراهم آورد که پیام را به پروتکل کاربردی مناسب آن تحویل نماید و از بیت‌های تشخیص خطا جهت شناسایی اینکه آیا بیت‌های اصلی پیام به هنگام گذر از مسیر، تغییر یافته‌اند یا خیر، استفاده نماید. لایه‌ی انتقال سپس قطعه را به لایه‌ی شبکه منتقل می‌نماید و در آنجا این لایه عنوان H_n را که شامل اطلاعاتی از قبیل آدرس‌های دو میزبان مبداء و مقصد می‌باشد به آن اضافه می‌کند.

حاصل این افزودن، داده گرام لایه‌ی شبکه است. داده گرام سپس به لایه‌ی پیوند تحویل می‌شود و در اینجا نیز اطلاعات سرپیام این لایه H_ℓ به آن افزوده شده و سرانجام قاب یا فریم لایه‌ی پیوند تشکیل می‌گردد. ملاحظه می‌شود هر یک از لایه‌ها بسته‌ای را می‌سازند که به طور کلی شامل دو بخش سرپیام و محموله می‌باشد. محموله معمولاً آن بخش از اطلاعات بسته است که از لایه‌ی بالاتر دریافت شده است.

۱- ۱۷ شبکه در تهدید

شبکه‌ی اینترنت در حال حاضر برای بسیاری از کاربران از حساسیت ویژه‌ای برخوردار است. این کاربران در قالب شرکت‌های کوچک، دانشگاه‌ها، سازمان‌های دولتی و بسیاری از افراد جامعه که به‌صورت فردی به نوعی وابسته به دریافت و ارسال اطلاعات از اینترنت می‌باشند، گروه‌بندی شده‌اند. در پس همه این کاربردها و تقاضاها تهدیدی است که همه‌ی این گروه‌ها با آن دست به گریبان می‌باشند. آن تهدید کوشش عده‌ای از همین کاربران جهت نفوذ در شبکه‌ها و کامپیوترهای متصل به اینترنت جهت ایجاد اخلال و یا متوقف نمودن روند جابه‌جایی اطلاعات می‌باشد. بخش امنیت شبکه مربوط به مقابله با این تهدیدات با توجه به سرعت تکرار و نیز نوع این حمله‌ها، شناسایی تهدیدهای جدید و چگونگی مقابله با آنها امروز تبدیل به شاخه‌ای تخصصی در شبکه‌های کامپیوتری شده است. این بخش گذری سریع بر انواع این تهدیدها و دشواری‌هایی که آنها بر شبکه اعمال می‌نمایند، دارد. در فصل ۸ موضوع امنیت شبکه با جزئیات بیشتری بررسی خواهد شد. در این مقطع سعی در پاسخ به پرسش‌هایی است از قبیل اینکه در اثر حمله چه روی خواهد داد؟

1- Encapsulation

شبکه‌های کامپیوتری چه نقاط ضعفی در برابر تهدیدها دارند؟ و انواع تهدیدهایی که امروز می‌تواند این شبکه‌ها را مورد هدف قرار دهد چیست؟

برای پاسخ به این پرسش‌ها موارد مختلف بررسی می‌گردد. از جمله‌ی این موارد می‌تواند با این موضوع آغاز گردد: تجهیزاتی که امروز به اینترنت متصل می‌شوند غالباً در جهت دریافت و یا ارسال از اینترنت می‌باشند. این موارد طیف گسترده‌ای از امکانات و کاربردها نظیر صفحه‌های وب، پست الکترونیکی، MP3، ارتباط‌های تلفنی، پخش تصاویر زنده، نتایج حاصل از موتورهای جستجو و غیره را شامل می‌شوند. اما در مقابل همه‌ی این زیبایی‌ها عوامل تهدیدگر و مخرب نیز وجود دارند که به‌صورت گروهی **ابزار تخریب**[1] (بدافزار) نام گرفته‌اند. این نام از آن جهت است که اینها می‌توانند با ورود خود و آلوده نمودن تجهیزات آنها را فلج نمایند. هنگامی که ابزار تخریب (بدافزار) وسیله‌ای از ما را آلوده نماید، آنگاه به صورت‌های مختلف می‌تواند تأثیر تخریبی خود را اعمال نماید. از جمله‌ی این تأثیرها می‌تواند حذف فایل‌های حیاتی بر روی کامپیوتر و یا شبکه باشد و یا آنکه با نصب یک **ابزار جاسوسی**[2] به جمع آوری اطلاعات خصوصی از ما نظیر شماره حساب‌ها، اطلاعات بانکی، رمزهای عبور و کلیدهای اختصاری پرداخته و سپس آنها را از طریق اینترنت به سوی تهدیدکننده گسیل نماید. میزبان آلوده شده‌ی ما ممکن است در شبکه‌ای ثبت شده باشد که آن شبکه خود شامل هزاران وسیله‌ی آلوده شده‌ی مشابه باشد. این مجموعه و حالت بوجود آمده را در اصطلاح بحث امنیت شبکه **ویروس ربات اینترنتی**[3] نامیده‌اند. این مجموعه در حقیقت به‌وسیله‌ی تهدیدگران کنترل و از طریق **کرم‌های**[4] الکترونیکی در سراسر اینترنت توزیع می‌گردند. در برخی موارد این‌گونه حمله‌ها ممکن است سبب جلوگیری از رسیدن و دریافت اطلاعات از یک شبکه و یا یک سیستم شوند. اصطلاحاً این پدیده را **توزیع جلوگیری از دسترسی**[5] به شبکه می‌نامند.

بسیاری از این ابزارهای تخریب در حال حاضر به گونه‌ی خود تکرار عمل می‌نمایند. خود تکراری به مفهوم آن است که هنگامی که این ابزار میزبانی را آلوده نمایند از طریق آن میزبان به جستجوی دیگر قربانیان (میزبان‌ها) از طریق اینترنت می‌پردازند و باز هم از طریق میزبان‌های جدید آلوده شده به دنبال قربانیان بعدی می‌گردند. با این روند ابزار تخریبگر خود تکرار به‌صورت تصاعدی و با سرعت زیاد به سرتاسر شبکه‌ی اینترنت گسترش می‌یابند. به‌عنوان مثال تعداد سیستم‌های آلوده شده از این طریق تا انتهای سال 2003 میلادی به‌وسیله کرم سفایر و اسلامر بیش از 90 درصد میزبان‌های قابل تهدید بود که در ظرف مدت کوتاه ده دقیقه آلوده شدند. بعضی اوقات یک ابزار تخریب می‌تواند به گونه‌ی یک ویروس رفتار نماید همانند یک کرم و یا ویروس اسب تراوا[6].

ویروس‌ها ابزار تخریبی هستند که معمولاً جهت آلوده نمودن تجهیزات کاربر به نوعی از تعامل با کاربر نیازمند می‌باشند. نمونه‌ی کلاسیک آن پیوست‌های پست الکترونیکی است که کاربر بدون اختیار آن ابزار تخریب را بر روی سیستم خود اجرا می‌نماید. اغلب این ویروس‌ها به گونه‌ی خود تکرار عمل می‌نمایند. هنگامی که ویروس اجرا می‌شود، پیامی را به‌صورت مشابهی با یک کُد تخریب به همه‌ی

1- Malware
2- Spyware
3- Robot Software net.
4- Spams
5- Denial of service attack
6- Worm / Trojan horse

آدرس‌های اضافه شده به دفترچه‌ی آدرس آن کاربر ارسال می‌نماید. کرم‌ها نظیر کرم اسلامر ابزارهای تخریبی هستند که بدون نیاز به تعامل کاربر وارد یک وسیله می‌شوند. به‌عنوان مثال یک کاربر ممکن است در حال اجرای یک برنامه‌ی کاربردی شبکه که در تهدید است قرار داشته باشد و تهدید کننده از همین طریق ابزار تخریبی خود را به سمت آن ارسال می‌نماید.

در بسیاری از موارد برنامه‌ی کاربردی بدون دخالت کاربر ابزار تخریبی را از اینترنت دریافت و سپس آنرا اجرا می‌نماید، حاصل آن تولید یک کرم است. کرم تولید شده در سیستم جدید سپس شبکه‌ی اینترنت را برای اجرای خود بر روی میزبان‌های جدید و قابل تهدید کاوش می‌نماید و از طریق همان برنامه‌ی کاربرد مشابه دامنه‌ی نفوذ خود را گسترش می‌دهد. گسترش این نفوذ به این‌گونه است که با یافتن میزبان‌های مناسب نسخه‌ای از خود را به سمت آن گُسیل می‌دارد. سرانجام ابزارهای تخریب از نوع اسب تراوا می‌باشند. اینگونه تهدیدکننده به‌صورت بخش پنهان بعضی از نرم‌افزارهای سودمند عمل می‌نمایند. در حال حاضر مبارزه و خنثی نمودن ابزارهای تخریب پر هزینه و تقریباً غیر ممکن می‌باشد.

یک گروه گسترده تر از تهدیدهای امنیتی شبکه جلوگیری از ارائه‌ی خدمات[1] (DoS) نام گرفته‌اند. همانگونه که از مفهوم آن استنباط می‌شود یک تهدید یا حمله‌ی DoS سبب می‌شود یک شبکه، میزبان و یا دیگر تجهیزات زیرساخت شبکه عملاً از طریق کاربران مجاز قابل دسترسی نباشند. سرورهای وب، پست الکترونیکی، DNS و شبکه‌های نهادهای مختلف علمی، دولتی و تجاری معمولاً در معرض این‌گونه تهدیدها می‌باشند.

تهدیدهای اینترنتی از نوع DoS بسیار متداول و هزاران حمله از این نوع در هر سال رخ می‌دهد. به طور کلی تهدیدهای از نوع DoS به سه گروه زیر از نقطه‌نظر پژوهشگران امنیت شبکه تقسیم می‌شوند:

- **حساس در برابر حمله**

این گروه تهدیدها از طریق ارسال تعداد محدودی پیام کُدنویسی شده‌ی پیشرفته به سمت یک برنامه‌ی کاربردی حفاظت نشده و یا یک سیستم عامل مورد هدف در یک میزبان صورت می‌پذیرد. اگر تعدادی بسته‌ی مناسب جهت حمله به این سیستم‌ها ارسال و به‌وسیله‌ی آنها دریافت و اجرا گردد، نتیجه آن توقف سرویس و یا در بدترین حالت به هم ریختگی و سقوط میزبان می‌باشد.

- **سیلابی نمودن پهنای باند**

در این حالت مهاجم به‌صورت سیلابی، بسته‌های آلوده کننده را به سمت یک میزبان هدف گسیل می‌دارد. انبوه بسته‌های ارسالی سبب می‌شوند پیوند دستیابی به آن میزبان به اندازه‌ای کند شود که دیگر قابل دسترسی نباشد. نتیجه‌ی آن عدم دسترسی کاربران مجاز به آن پیوند و سرانجام به آن میزبان خواهد بود.

- **سیلابی نمودن ارتباط**

در این نوع تهدید، مهاجم به ایجاد تعداد انبوهی ارتباط‌های یک و یا دو طرفه‌ی باز از نوع TCP می‌پردازد. این ارتباط‌ها در میزبان هدف ایجاد می‌شوند. نتیجه‌ی آن اشتغال میزبان به پردازش این

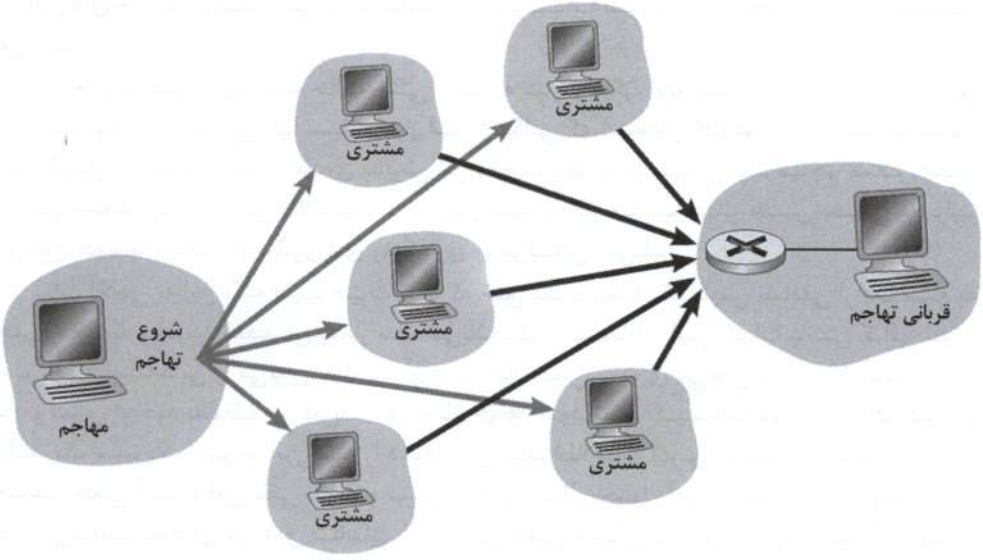
1- Denial of Service

آلودگی‌ها و ارتباط‌های مربوط به آنها می‌شود که سرانجام آن عدم پذیرش ارتباط‌های مجاز و در نتیجه فلج شدن دسترسی به آن میزبان خواهد شد.

در این مقطع به بررسی تأثیر سیلابی نمودن پهنای باند (سرعت) می‌پردازیم. قبلاً ملاحظه گردید که تأثیر تأخیر و اتلاف بر روی یک سرور چگونه خواهد بود. اگر یک سرور دارای سرعت دستیابی R بیت بر ثانیه باشد، آنگاه یک مهاجم کافی است یک ترافیک، با سرعت تقریباً R بیت بر ثانیه را به سمت آن گسیل نماید تا به آن زیان لازم را وارد نماید. اگر مقدار R بسیار بالا باشد یعنی پیوند دارای سرعت بسیار بالایی باشد، تنها حمله‌ی یک منبع ممکن است ترافیک کافی برای به زانو درآوردن آن سرور را تولید ننماید. علاوه بر آن اگر همه‌ی ترافیک از سوی یک منبع انجام پذیرد، ممکن است مسیریابی که در بین این ترافیک تا هدف قرار گرفته است این حمله را شناسایی و کلیه‌ی ترافیک حاصل از آن منبع را قبل از آنکه به سرور هدف وارد شود مسدود نماید.

در این صورت برای جلوگیری از این وضعیت مهاجمان معمولاً DoS را به‌صورت **توزیع‌شده**[1] (DDoS) انجام می‌دهند.

شکل ۱- ۲۵ این نوع حمله را نشان می‌دهد. در تهدید DDoS مهاجم چندین منبع را کنترل می‌نماید و هر منبع ترافیک مورد نظر را به سمت هدف توزیع می‌کند.



شکل ۱- ۲۵ ♦ تهاجم توزیع شده‌ی DoS

در این رویکرد نرخ ترافیک انباشتی در همه‌ی منابع کنترل شده بایستی تقریباً برابر R باشد، تا سرور هدف را فلج نماید. تهاجم DDoS سبب می‌شود، هزاران میزبان در هر مقطع آلوده گردند. این نوع تهدید در حال حاضر بسیار متداول است و یک تهاجم DDoS از لحاظ تشخیص و دفاع بسیار دشوارتر از یک تهدید DoS ناشی از یک منبع می‌باشد.

در حال حاضر بسیاری از کاربران دسترسی به اینترنت را از طریق تجهیزات بدون سیم نظیر

1- Distributed DoS

لَپ‌تاپ‌های متصل از طریق WiFi و یا گوشی‌های همراه انجام می‌دهند. اگرچه دستیابی‌های از این‌گونه بسیار مناسب و استفاده از نرم‌افزارهای کاربردی جدید و فوق‌العاده کارآمد را برای کاربران سیار فراهم می‌آورد اما از طرف دیگر ضعف‌های امنیتی به مراتب بالاتری را برای اینگونه کاربران ایجاد می‌کند.

با قرار گرفتن یک **گیرنده‌ی خنثی**[1] در نزدیکی یک فرستنده‌ی بدون سیم آن گیرنده می‌تواند نسخه‌هایی از هر بسته‌ی اطلاعاتی در حال ارسال را بدست آورد !

این بسته‌های اطلاعاتی می‌توانند حامل همه نوع داده‌های حساس باشند، از جمله رمزهای عبور، اطلاعات حساب‌های بانکی، اسرار تجاری و پیام‌های شخصی. یک گیرنده‌ی خنثی اگر بتواند به نسخه برداری از هر بسته‌ی در حال گذر از شبکه‌ی بدون سیم بپردازد، به آن **حسگر بسته**[2] گفته می‌شود. حسگرهای بسته می‌توانند در محیط‌های با سیم نیز مستقر شوند. در یک محیط فراپخش با سیم نظیر اترنت LAN یک حسگر بسته می‌تواند نسخه‌هایی از همه‌ی داده‌های در حال انتقال از آن شبکه را بدست آورد.

علاوه بر آن یک مهاجم که قادر باشد به روتر دستیابی یک نهاد و یا پیوند دستیابی به اینترنت وارد شود، می‌تواند نسبت به کاشت یک حسگر که از طریق آن بتواند هر بسته‌ی اطلاعاتی وارد و یا خارج شده را کپی نماید و سپس آنها را در فرصت مناسب و خارج از ارتباط تحلیل کند، مبادرت نماید. نرم‌افزارهای حسگر بسته از طریق سامانه‌های اینترنت به‌صورت آزاد و تجاری قابل دسترسی می‌باشند.

از آنجائیکه حسگرهای بسته حالت خنثی دارند، به عبارت دیگر، آنها بسته‌ای را به شبکه تزریق نمی‌کنند بنابراین شناسایی آنها بسیار دشوار است. هنگامی که بسته‌های اطلاعات به یک پیوند بدون سیم گسیل می‌شوند، بایستی آماده‌ی پذیرش این واقعیت شد که در نقطه‌ای یک مهاجم ممکن است در حال نسخه برداری از این اطلاعات باشد. بدیهی است راه کار مناسب جهت پوشش دفاعی در برابر این نوع حمله‌ها استفاده از فناوری‌های رمزگذاری و رمزگشایی[3] می‌باشد.

شاید آگاهی به این نکته سبب حیرت شود که خلق یک بسته با یک آدرس تصادفی از یک منبع به همراه محموله و آدرس مقصد آن بسیار ساده می‌باشد. در ادامه مطالب این کتاب دانش کافی برای خلق چنین بسته‌هایی را می‌توان به دست آورد. با ایجاد چنین بسته‌ای می‌توان آن را در اینترنت قرار داد و سپس او خود به وظیفه‌ی تعریف شده عمل خواهد نمود. تصور کنید هنگامی که یک گیرنده مانند یک مسیریاب اینترنت این بسته را دریافت می‌کند، اطلاعات آدرس مقصد آن بسته را که در حقیقت جعلی است واقعی تلقّی نموده و سپس با به روز رسانی جدول آدرس مسیرها این آدرس را نیز وارد می‌نماید، نتیجه‌ی این عمل شناسایی این آدرس جعلی به‌عنوان یک آدرس مجاز است که هدف خالق بسته نیز همین می‌باشد.

در اصطلاح ادبیات شبکه تبدیل این‌گونه آدرس‌های جعلی به آدرس‌های مجاز را **جعل آدرس پروتکل اینترنت**[4] می‌نامند. این راهکار تنها یکی از روش‌های جعل آدرس به‌وسیله‌ی کاربران می‌باشد. برای مبارزه با این چالش، راهکار مناسب اعتبارسنجی در نقطه‌های انتهایی شبکه می‌باشد. مفهوم این عبارت آن است که: مکانیزم هایی بایستی فراهم شود که با درصد بالایی این اطمینان را ایجاد نمایند

1- Passive reciever
2- Packet Sniffer
3- Cryptography
4- IP Spoofing

که پیام دریافت شده جعلی نیست و از اعتبار لازم برخوردار است. فصل ۸ این اثر موضوعات مربوط به این مهم را بررسی نموده است.

در انتها نوع دیگری از این تهدیدها تحلیل می‌گردد که اصطلاحاً **تهاجم‌های از طریق واسطه**[1] نام گرفته‌اند. در این گروه از تهدیدها، مهاجم وارد یک مسیر ارتباطی بین دو موجودیت در حال محاوره می‌شود. فرض کنید دو موجودیت در حال ارتباط، A و B باشند.

این دو می‌توانند دو مسیریاب، دو سرور پست الکترونیکی و نظایر آن باشند. در این نوع تهدید، مسیریاب واقع در پیوند ارتباط و یا یک نرم‌افزار ساکن در یکی از میزبان‌های انتهایی در یک لایه‌ی زیرین پشته‌ی پروتکلی می‌تواند مورد هجوم قرار گیرد. در تهاجم از طریق واسطه، عامل تهدید می‌تواند همه‌ی بسته‌هایی که مابین A و B در حال عبور می‌باشند را نسخه برداری نماید و یا آنکه اطلاعات جدیدی را به آنها تزریق نماید و یا تغییراتی را در اطلاعات بسته به‌وجود آورد و یا آنکه به‌طور کلی آن اطلاعات را حذف کند. در این شرایط یکپارچگی و اصالت داده‌ها در یک بسته مورد تهدید قرار می‌گیرد.

در فصل ۸ راهکارهای لازم برای مقابله با این تهدیدها از طریق ایجاد حفاظت‌های لازم در برابر حسگرها و یا اعتبارسنجی انتهایی که از طریق آن بتوان پی به اصالت یک پیام برد، بررسی گردیده است.

یادآوری این نکته ضروری است که بنیان‌گذاران اینترنت اولیه نیازی را برای تأمین امنیت انتقال اطلاعات متصوّر نبودند، زیرا بنیان انتقال براساس اطمینان شالوده‌ریزی شده بود. در حال حاضر موضوع اینترنت، تبدیل به انتقال اطلاعات به‌صورت محرمانه، غیر قابل اطمینان و سوء رفتار شده است. افراد جهت مبادله‌ی اطلاعات می‌خواهند در سایه قرار گیرند و یا از طریق امکانات ثالث و غیر مستقیم نظیر سرورهای نهان و یا عامل‌های **دستیار متحرک**[2] جهت منظور خود استفاده نمایند، آنها ممکن است هیچگونه اطمینانی به سخت‌افزار و نرم‌افزارهای استفاده شده و یا حتی عبوری از مسیر ارتباطی خود نداشته باشند. بنابراین ملاحظه می‌شود در بخش امنیت شبکه یک کارشناس شبکه با چالش‌های امنیتی بسیاری رو به رو می‌باشد.



تمرین‌های فصل اول

۱- چه تفاوتی بین یک میزبان و یک سیستم انتهایی می‌باشد. توضیح دهید. انواع سیستم‌های انتهایی را نام برده و مشخص کنید آیا یک وب سرور، سیستم انتهایی به‌شمار می‌آید؟

۲- تعداد 6 فناوری دستیابی به شبکه را نام برده و سپس آنها را در قالب مسکونی، تجاری و سیار گروه‌بندی نمائید.

۳- برنامه مشتری چیست؟ برنامه سرور چیست؟ آیا برنامه‌ی سرور می‌تواند درخواستی را از برنامه مشتری داشته باشد؟

۴- رسانه‌های فیزیکی که شبکه‌ی اترنت از آن استفاده می‌کند را توضیح دهید؟

1- Man in the middle attacks

2- Mobility assisting agents

۵- فناوری‌های متداول دستیابی بی‌سیم به اینترنت را که در حال حاضر مورد استفاده قرار می‌گیرد نام برده، آنها را با یکدیگر مقایسه نمائید؟

۶- سرعت انتقال در شبکه‌ی محلی اترنت چقدر است؟ برای یک سرعت انتقال تعریف شده هر کاربر می‌تواند به‌صورت مستمر با همان سرعت به انتقال اطلاعات بپردازد؟

۷- فرض کنید تنها یک راهیاب بسته بین یک میزبان ارسال و دریافت وجود دارد. اگر سرعت انتقال بین میزبان ارسال و راهیاب R1 و بین میزبان دریافت و راهیاب R2 باشد و راهیاب از روش ذخیره و ارسال بدون در نظر گرفتن تأخیر صف، تأخیر انتشار و تأخیر پردازش استفاده نماید و طول بسته نیز L بیت باشد آنگاه زمان تأخیر انتها به انتها برای ارسال یک بسته چقدر است؟

۸- فرض کنید یک پیوند 2 Mbps بین کاربران به اشتراک گذارده شده است. هم چنین فرض کنید هر کاربر به‌صورت مستمر با سرعت 1 Mbps مشغول ارسال اطلاعات می‌باشد. چنانچه هر کاربر فقط 20 درصد از زمان را به انتقال اختصاص دهد و از روش مالتی‌پلکس آماری استفاده نماید آنگاه:

الف) اگر از راهیابی مداری استفاده شود چند کاربر امکان استفاده از شبکه را دارند؟

ب) اگر از راهیابی بسته استفاده گردد. توضیح دهید چرا اصولاً تأخیر صف در نقطه‌ی پیش از یک پیوند هنگامی که دو و یا اندکی کاربر هم‌زمان در حال ارسال باشند پدید نمی‌آید؟ چرا اگر تعداد کاربران هم‌زمان 3 گردد، تأخیر صف حاصل خواهد شد؟

ج) با توجه به شرایط بند (ب) احتمال آنکه یک کاربر در حال ارسال باشد چقدر است؟

د) اگر فرض شود 3 کاربر در حال استفاده می‌باشند احتمال آنکه هر 3 کاربر هم‌زمان مشغول انتقال اطلاعات باشند با توجه به بند (ب) چقدر است؟ و در این شرایط، زمان رشد صف چگونه بیان می‌شود؟

۹- چرا روش راهیابی بسته از مالتی‌پلکس آماری استفاده می‌کند؟ مقایسه‌ای را بین مالتی‌پلکس آماری و آنچه در TDM به‌عنوان مالتی‌پلکس روی می‌دهد، انجام دهید.

۱۰- فرض کنید در حال گسیل یک بسته از یک میزبان مبداء و از طریق یک مسیر ثابت به یک میزبان مقصد می‌باشید. صورتی از عوامل تأخیرهای انتها به انتها را بنویسید. کدامیک از این تأخیرها ثابت و کدامیک متغیرند؟

۱۱- فرض کنید میزبان A در حال ارسال یک فایل پرحجم به میزبان B می‌باشد. مسیر انتقال بین این دو میزبان 3 پیوند است، هر کدام با سرعت‌های:

R3 = 1 Mbps ،R2 = 2 Mbps ،R1 = 500 Kbps

الف) با فرض آنکه ترافیک دیگری در شبکه وجود ندارد. مقدار بازدهی یک انتقال فایل چقدر است؟

ب) اگر طول فایل معادل 4 میلیون بایت باشد، زمان تقریبی انتقال از A به B چقدر است؟

د) اگر سرعت R2 به 100 Kbps تقلیل پیدا کند آنگاه پاسخ بندهای (الف) (ب) چه خواهد بود؟

۱۲- چه مدت زمان طول می‌کشد تا یک بسته به طول 1000 بایت از طریق یک پیوند 2500 km با سرعت انتشار 2.5×10^8 m/s و سرعت 2 Mbps انتقال یابد؟ به‌صورت کلی زمان لازم برای انتقال یک بسته به طول L، در فاصله d، سرعت انتشار s و سرعت پیوند Rbps

چقدر است؟ آیا این زمان وابسته به طول بسته است؟ آیا می‌تواند وابسته به سرعت انتقال باشد، چرا؟

۱۳- کدام لایه در پشته‌ی پروتکلی اینترنت مورد پردازش مسیریاب (روتر) واقع می‌شود؟ کدام لایه را راهیاب لایه پیوند پردازش می‌کند؟ و سرانجام کدام لایه به‌وسیله‌یک میزبان پردازش می‌شود؟

۱۴- پنج وظیفه‌ای که هر لایه انجام می‌دهد چیست؟ آیا امکان انجام یک و یا بیش از یک وظیفه که از طریق دو یا بیش از آن لایه انجام گردد وجود دارد؟

۱۵- تفاوت بین یک ویروس، یک کرم و یک اسب تراوا چیست؟ توضیح دهید.

۱۶- توضیح دهید یک ویروس ربات اینترنتی چگونه تشکیل می‌شود و از آن چگونه برای یک تهاجم DDoS می‌توان استفاده نمود؟

۱۷- شبکه‌ی راهیابی مدار شکل ۱- ۸ را که دارای n مدار در هر پیوند است در نظر بگیرید:

الف) حداکثر تعداد ارتباط‌های هم‌زمان در هر لحظه از زمان در این شبکه چقدر است؟

ب) اگر همه ارتباط‌ها بین راهیاب بالا در گوشه‌ی سمت چپ و راهیاب پائین در گوشه سمت راست باشد، حداکثر تعداد ارتباط‌های هم‌زمان در هر لحظه چقدر خواهد شد؟

۱۸- با توجه به مثال ناوگان خودرو در بخش ۱- ۱۱ اگر سرعت انتشار معادل 100 Km/hr (یکصد کیلومتر بر ساعت) فرض شود آنگاه:

الف) فرض کنید ناوگان طول 200 کیلومتر را با شروع از مقابل یک اتاقک عوارض و سپس عبور از اتاقک دوم و دقیقاً قبل از رسیدن به اتاقک سوم به پایان برساند. اندازه‌ی زمان تأخیر انتها به انتها چقدر است؟ تعداد خودروها در ناوگان 10 می‌باشد.

ب) اگر فرض شود تعداد خودرو در ناوگان 7 باشد آنگاه پاسخ پرسش‌های بند (الف) چه خواهد بود؟

۱۹- هدف این پرسش بررسی دو اصل پایه‌ای در شبکه‌ها می‌باشد. این دو عبارتند از تأخیر انتقال و تأخیر انتشار. دو میزبان A و B که از طریق یک پیوند با سرعت Rbps به یکدیگر متصل شده‌اند را در نظر بگیرید. فرض کنید فاصله‌ی این دو میزبان از یکدیگر m متر و سرعت انتشار در مسیر پیوند s متر بر ثانیه باشد اگر میزبان A بخواهد بسته‌ای به طول L بیت را به میزبان B ارسال نماید آنگاه:

الف) زمان تأخیر انتشار d_{prop} را برحسب m و s بیان کنید.

ب) زمان انتقال بسته d_{trans} را برحسب L و R بیان کنید.

ج) بدون در نظر گرفتن تأخیرهای صف و پردازش، عبارتی را برای تأخیر انتها به انتها بدست آورید.

د) فرض کنید میزبان A شروع به ارسال بسته‌ای در زمان $t = 0$ نماید. در زمان $t = d_{trans}$ آخرین بیت بسته در کدام نقطه است؟

ه) اگر فرض شود d_{prop} بزرگتر از d_{trans} است. آنگاه در زمان $t = d_{trans}$ نخستین بیت بسته در کدام نقطه قرار دارد؟

و) اگر d_{prop} کوچکتر از d_{trans} باشد در زمان $t = d_{trans}$ نخستین بیت بسته در کجا قرار دارد؟

ز) اگر مقدار $s = 2.5\times10^8$ m/s، L = 100 bits و R = 28 Kbps باشد. آنگاه فاصله‌ی m برای آنکه $d_{prop} = d_{trans}$ باشد، چقدر است؟

۲۰- در این پرسش ارسال یک مکالمه تلفنی (صوت) به‌صورت زمان واقعی از میزبان A به B از طریق یک شبکه‌ی راهیابی بسته (VOIP) منظور می‌باشد. میزبان A صدای آنالوگ را به یک سیگنال دیجیتال 64 Kbps برای حرکت تبدیل می‌کند. میزبان A آنگاه بیت‌ها را در قالب گروه‌های 48 بایت سازماندهی می‌نماید. اگر یک پیوند بین A و B موجود است و سرعت انتقال آن نیز 1 Mbps و زمان تأخیر انتشار آن نیز 2 ms باشد.

در صورتی‌که A بلافاصله پس از جمع آوری یک بسته آن را به B ارسال نماید و اگر بلافاصله پس از آنکه B همه‌ی بسته را دریافت نمود بیت‌های بسته را به یک سیگنال آنالوگ تبدیل نماید. زمان لازم از لحظه‌ی خلق بیت‌ها در میزبان A تا رمزگشایی آنها در میزبان B چقدر است؟

۲۱- موضوع مربوط به مالتی‌پلکس آماری در بخش ۱- ۸ که در آن مثالی از یک پیوند 1 Mbps بیان شده بود را در نظر بگیرید. اگر کاربران به هنگام اشتغال داده‌هایی را با سرعت 100 Kbps تولید کنند و احتمال این تولید داده نیز p = 0.1 باشد. اگر فرض شود پیوند 1 Mbps با یک پیوند 1 Gbps جایگزین گردد. آنگاه:

الف) مقدار N حداکثر تعداد کاربران هم‌زمان که می‌توان تحت شرایط راهیابی مداری پشتیبانی نمود چقدر است؟

ب) اگر از روش راهیابی بسته استفاده شود و جمعیت کاربران نیز M باشد، رابطه‌ای را برحسب P، M و N تعریف کنید که در آن احتمال ارسال داده به‌وسیله‌ی تعداد کاربران بیش از N امکان‌پذیر باشد.

۲۲- در پرسش ۲۱ اگر $R_1=R_2=R$ و $d_{proc}=0$ و راهیاب بسته نیز از روش ذخیره و سپس ارسال استفاده نکند و در عوض بلافاصله هر یک بیت دریافتی را ارسال نماید و منتظر دریافت بسته نباشد زمان تأخیر انتها به انتها چقدر است؟

۲۳- اگر فرض شود تعداد N بسته به‌صورت هم‌زمان وارد یک پیوند گردد که در حال حاضر بسته‌ای را در حال ارسال و یا انتظار در صف ندارد. اگر هر بسته دارای طولی معادل L بیت و سرعت انتقال پیوند نیز R بیت بر ثانیه باشد، آنگاه میانگین زمان تأخیر صف برای تعداد N بسته چقدر است؟

۲۴- اگر تأخیر صف در یک بافر مسیریاب در نظر گرفته شود و I شدت ترافیک باشد که مفهوم آن $I=\frac{L\cdot a}{R}$ می‌باشد. چنانچه تأخیر صف به‌صورت رابطه ی: $\frac{I.L}{R(1-I)}$ برای $I<1$ تعریف شده باشد. آنگاه:

الف) نموداری را رسم نمائید که مجموع تأخیر به‌صورت تابعی از $\frac{L}{R}$ نشان دهد.

ب) رابطه‌ای برای مجموع تأخیر که شامل تأخیر صف و تأخیر انتقال باشد اثبات نمائید.

۲۵- مثال بازدهی شبکه مربوط به شکل ۱- ۱۹ (ب) را در نظر بگیرید. اگر فرض شود تعداد مشتری و سرورها به جای 10 برابر N باشد و R_s، R_c و R به ترتیب سرعت پیوندهای سرور، مشتری و پیوندهای شبکه باشند. علاوه بر آن اگر فرض شود ترافیک دیگری در شبکه به غیر از

ترافیک تولید شده به‌وسیله‌ی M زوج مشتری و سرور وجود ندارد. یک رابطه‌ی کلی برای بازدهی برحسب R_s، R_c، R و M تعریف نمائید.

۲۶- اگر در پرسش ۲۵ سرعت R = 1 Gbps فرض شود آنگاه:

الف) عرض یک بیت بر مقیاس متر در این پیوند چقدر است؟

ب) مقدار حاصلضرب تأخیر و پهنای باند یعنی $R.d_{prop}$ چقدر است؟

ج) اگر ارسال یک فایل به حجم 400000 بیت از میزبان A به B هدف باشد و فرض شود داده‌های فایل در قالب یک پیام طولانی و به‌صورت پی در پی ارسال گردد آنگاه حداکثر تعداد بیت هایی که در پیوند در هر لحظه‌ای زمانی وجود دارد چقدر است؟

۲۷- با اشاره به پرسش ۲۵ اگر اندازه‌ی R تغییر نماید. برای چه ارزشی از R عرض یک بیت برحسب متر برابر طول پیوند خواهد شد؟

۲۸- با توجه به پرسش ۲۵ به موارد زیر پاسخ دهید:

الف) اگر فرض شود حجم فایل به تعداد 10 بسته هر کدام با بزرگی 40000 بیت تقسیم گردد و اگر زمان ارسال و تأئید هر بسته ناچیز باشد و نیز فرستنده قادر به ارسال یک بسته تا زمانیکه تأئید بسته‌ی قبلی اعلام نشده است، نمی‌باشد. آنگاه زمان لازم برای ارسال فایل چقدر است؟

ب) اگر فایل به‌صورت پی در پی ارسال گردد این زمان چقدر است؟

ج) دو نتیجه‌ی (الف) و (ب) را با یکدیگر مقایسه نمائید چه نتیجه‌ای می‌گیرید؟

۲۹- فرض کنید یک پیوند مایکروویو با سرعت 10 Mbps بین یک ماهواره‌ی ثابت و ایستگاه زمینی آن وجود دارد. اگر در هر دقیقه ماهواره یک تصویربرداری دیجیتال نماید و آن تصویر را به ایستگاه زمین گسیل نماید. چنانچه سرعت انتشار پیوند 2.4×10^8 m/s باشد، آنگاه:

الف) حاصلضرب پهنای باند و تأخیر $R.d_{prop}$ چقدر است؟

ب) حداکثر تأخیر انتشار پیوند چقدر است؟

ج) اگر x اندازه‌ی تصویر باشد، حداقل مقدار x چقدر بایستی در نظر گرفته شود تا پیوند مایکروویو بتواند به‌صورت مستمر اطلاعات آن را ارسال نماید.

۳۰- فرض کنید دو میزبان A و B در فاصله 10000 کیلومتر از یکدیگر قرار گرفته‌اند و از طریق یک پیوند مستقیم R =1Mbps به یکدیگر متصل شده‌اند. اگر سرعت انتشار پیوند برابر 2.5×10^8 m/s باشد. حساب کنید:

الف) حاصلضرب تأخیر و پهنای باند $R.d_{prop}$ چقدر است؟

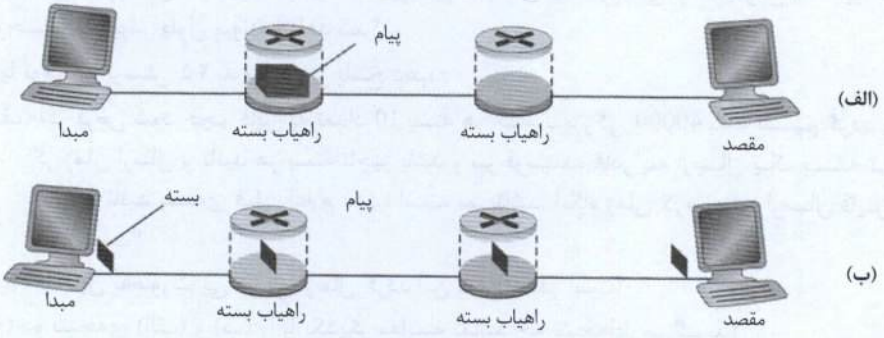
ب) اگر یک فایل به حجم 400000 بیت از میزبان A به B در حال گسیل باشد، حداکثر تعداد بیت‌ها در هر لحظه از زمان در پیوند در صورتی‌که ارسال فایل به‌صورت پی در پی باشد چقدر است؟

ج) تفسیری را از حاصلضرب تأخیر و پهنای باند ارائه نمائید.

د) عرض یک بیت برحسب متر در این پیوند چقدر است؟ آیا این عرض از یک میدان فوتبال بیشتر است؟

ه) سعی کنید یک عبارت را برای عرض یک بیت بر مقیاس زمان انتشار s، سرعت انتقال R و طول پیوند m تعریف کنید.

۳۱- در شبکه‌های راهیابی بسته‌ی مدرن، میزبان مبداء پیام‌های طولانی لایه کاربرد نظیر یک فایل تصویری و یا موسیقی را به قطعه‌های کوچکتری تقسیم می‌نماید و سپس آنها را ارسال می‌دارد. میزبان دریافت کننده پس از دریافت این بسته‌ها آنها را به یکدیگر پیوند می‌دهد و پیام اصلی مجدداً بازیابی می‌شود. شکل ۱- ۲۶ این انتقال انتها به انتها را از یک پیام به‌صورت قطعه‌بندی شده و بدون آن نشان می‌دهد. اگر فرض شود اندازه‌ی پیام ارسالی 7.5×10^6 بیت باشد و از طریق مبداء شکل ۱- ۲۶ به مقصد آن از طریق پیوندی که سرعت آن 1.5 Mbps است، بدون در نظر گرفتن تأخیرهای انتشار، صف و پردازش ارسال گردد آنگاه:



شکل ۱- ۲۶ ♦ انتقال انتها به انتهای پیام: الف) بدون قطعه‌بندی ب) همراه با قطعه بندی

الف) اگر پیام بدون قطعه‌بندی از مبداء به مقصد گسیل گردد، زمان لازم جهت حرکت پیام به سمت نخستین راهیاب چقدر است؟ اگر هر راهیاب از روش ذخیره و سپس ارسال استفاده نماید آنگاه مجموع زمان انتقال از میزبان مبداء به میزبان مقصد چقدر است؟

ب) هنگامی که از قطعه‌بندی پیام استفاده گردد، مجموع زمان انتقال از مبداء به مقصد چقدر است؟ دو نتیجه را با یکدیگر مقایسه نمائید.

ج) اگر فرض شود پیام به تعداد 5000 قطعه هر کدام به طول 1500 بیت تقسیم گردد آنگاه زمان لازم برای انتقال نخستین بسته از میزبان مبداء به نخستین راهیاب چقدر است؟ اگر هنگامی که نخستین بسته در حال ارسال از راهیاب اول به دوم است، دومین بسته از میزبان مبداء به راهیاب اول گسیل گردد. آنگاه در چه زمانی دومین بسته به‌صورت کامل به‌وسیله اولین راهیاب تحویل گرفته می‌شود.

د) چه نقاط ضعفی را می‌توان برای روش قطعه‌بندی پیام برشمرد؟

۳۲- فرض کنید هدف، ارسال یک فایل پرحجم F از میزبان A به B می‌باشد. اگر دو پیوند و یک راهیاب بین A و B باشد و پیوندها نیز کاملاً غیر متراکم باشند به عبارت دیگر تأخیر صف وجود ندارد. چنانچه میزبان A فایل را به قطعه‌های S بیت تقسیم نماید و به هر قطعه یک عنوان 80 بیت اضافه نماید که سرانجام آن، تشکیل بسته‌هایی به طول $L = 80 + S$ بیت باشد و سرعت هر پیوند نیز Rbps در نظر گرفته شود آنگاه مقدار S چقدر بایستی در نظر گرفته شود

تا حداقل تأخیر در جابه جایی فایل از A به B صورت پذیرد؟ از زمان تأخیر انتشار صرف نظر نمائید.

۳۳- نرم‌افزار Skype اجازه می‌دهد یک مکالمه‌ی تلفنی، تصویری از یک PC به یک تلفن معمولی انجام گیرد. این به معنی آن است که مکالمه بایستی از درون هر دو شبکه‌ی اینترنت و تلفنی عبور نماید. توضیح دهید این عمل چگونه انجام می‌شود؟

۳۴- سرویس SMS یا ارسال پیام‌های کوتاه چیست؟ آیا ارسال SMS از طریق یک سامانه‌ی اینترنتی به یک گوشی تلفن همراه امکان‌پذیر است؟ توضیح دهید.

۳۵- خدمات Skype به‌صورت PC به PC را توضیح دهید. آیا تا به حال آنرا تجربه کرده اید؟ برای اعضای کلاس خود توضیح دهید.

۳۶- آیا تا به حال از فناوری بدون سیم 802.11 برای ایجاد یک شبکه‌ی خانگی در منزل خود استفاده کرده‌اید؟ چگونگی تجربه‌ی ایجاد آن را توضیح دهید.

۳۷- آیا می‌توانید حداقل نام 5 شرکت که خدمات اشتراک فایل P2P فراهم می‌آورند را نام ببرید؟ به نظر شما هر یک از آنها با چه نوع فایل هایی کار می‌کنند؟

۳۸- آیا می‌توانید دستیابی اینترنت از طریق WiFi و 3G را با یکدیگر مقایسه کنید؟ سرعت انتقال هر یک از این دو سرویس چقدر است؟

۳۹- منظور از جریان تصویر زنده‌ی P2P چیست؟ سامانه‌های وب که در حال حاضر این خدمات را فراهم می‌آورند کدامند؟

۴۰- آیا می‌توانید توضیح دهید شبکه‌های اجتماعی کدامند؟ و چگونه عمل می‌کنند؟



فصل ۲

لایه‌ی کاربرد

نرم‌افزارهای کاربردی علت بقای اصلی شبکه‌های کامپیوتری و از جمله اینترنت به‌شمار می‌رود. اگر این برنامه‌های کاربردی مفید نبودند هیچگاه نیازی به طراحی پروتکل‌هایی که آنها را پشتیبانی نمایند، احساس نمی‌شد. از زمان ظهور اینترنت تاکنون برنامه‌های کاربردی بی‌شماری در قلمروهای وسیع علمی، تجاری و تفریحی خلق و به کاربران عرضه شده است. همگی این کاربردها در حقیقت انگیزه‌های لازم برای موفقیت اینترنت را تأمین کرده‌اند و سبب شده‌اند که اینترنت تبدیل به یک عامل مهم و اجتناب‌ناپذیر کاربردی در همه‌ی عرصه‌های زندگی روزانه‌ی مردمان از تجارت، امور جاری کشورداری، نیازهای علمی و کاربردی مراکز آموزشی از ابتدایی تا سطوح عالی و سرانجام اداره امور جاری در منازل مسکونی گردد. نمونه‌های آن عبارتند از: پست الکترونیکی، **دستیابی راه دور**[1]، انتقال فایل، **گروه‌های خبری یا اطلاع‌رسانی و گفتگوی متنی**[2]. نرم‌افزارهای کاربردی فراگیر دهه‌ی 1990 نظیر انواع ابزارهای وب گردی، جستجو و تجارت الکترونیکی و به دنبال آنها دو برنامه‌ی کاربردی گسترده‌تر انتقال پیام به همراه پیوست‌های متنوع و نیز اشتراک فایل P2P نمونه‌های دیگر این شگفت آوری‌ها قلمداد می‌شوند. از سوی دیگر از آغاز سال 2000 میلادی تاکنون شاهد انفجاری عظیم در کاربردهای صوتی و تصویری از قبیل انتقال صدا از طریق پروتکل اینترنت (VOIP)، توزیع محتوای تصویری تولید شده از سوی کاربران از طریق امکانات کاربردی مانند یوتیوب، دریافت فایل‌های سینمایی از طریق Netfix و مانندهای آن هستیم. به موازات شتاب در کاربردهای این‌گونه، شاهد ظهور بازی‌های آنلاین هوشمند و شگفت‌انگیزی مانند: حیات دوم (Second Life) و دنیای جنگ‌افزارها (World of Warcraft) و سرانجام در همین اواخر کاربردهای شبکه‌های اجتماعی مانند فیس‌بوک،

1- Login
2- Text chat

تویتر و نظایر آنها بوده‌ایم که سبب پیدایش شبکه‌های انسانی بر فراز شبکه‌هایی از مسیریاب‌ها و پیوندهای ارتباطی شده‌اند. بدیهی‌است آثاری از توقف و یا کُندی سرعت در خلاقیت‌هایی از این دست نه‌تنها مشاهده نمی‌شود بلکه هر روز با خلاقیت‌های جدیدی در این عرصه روبه‌رو می‌شویم و قطعاً از لحظه‌ی انتشار ویراست ششم این اثر تاکنون تغییرات و شگفتی‌های نوظهوری پا به عرصه‌ی فناوری کاربردهای اینترنتی گذارده‌اند. بدیهی است همگی آنها کاربردهای صوتی و تصویری را نیز با موفقیت به همراه خود دارند. از جمله این موارد می‌توان به تلفن اینترنتی، اشتراک ویدیویی و پخش زنده، رادیوهای اینترنتی و **رسانه‌های تصویری پروتکل اینترنت**[1] (IPTV) اشاره نمود. دستیابی گسترده به پیوندهای سریع و بدون سیم در مناطق مسکونی چشم‌انداز وسیعی را برای طراحی نرم‌افزارهای کاربردی جدید و هیجان‌آور در آینده نوید می‌دهد.

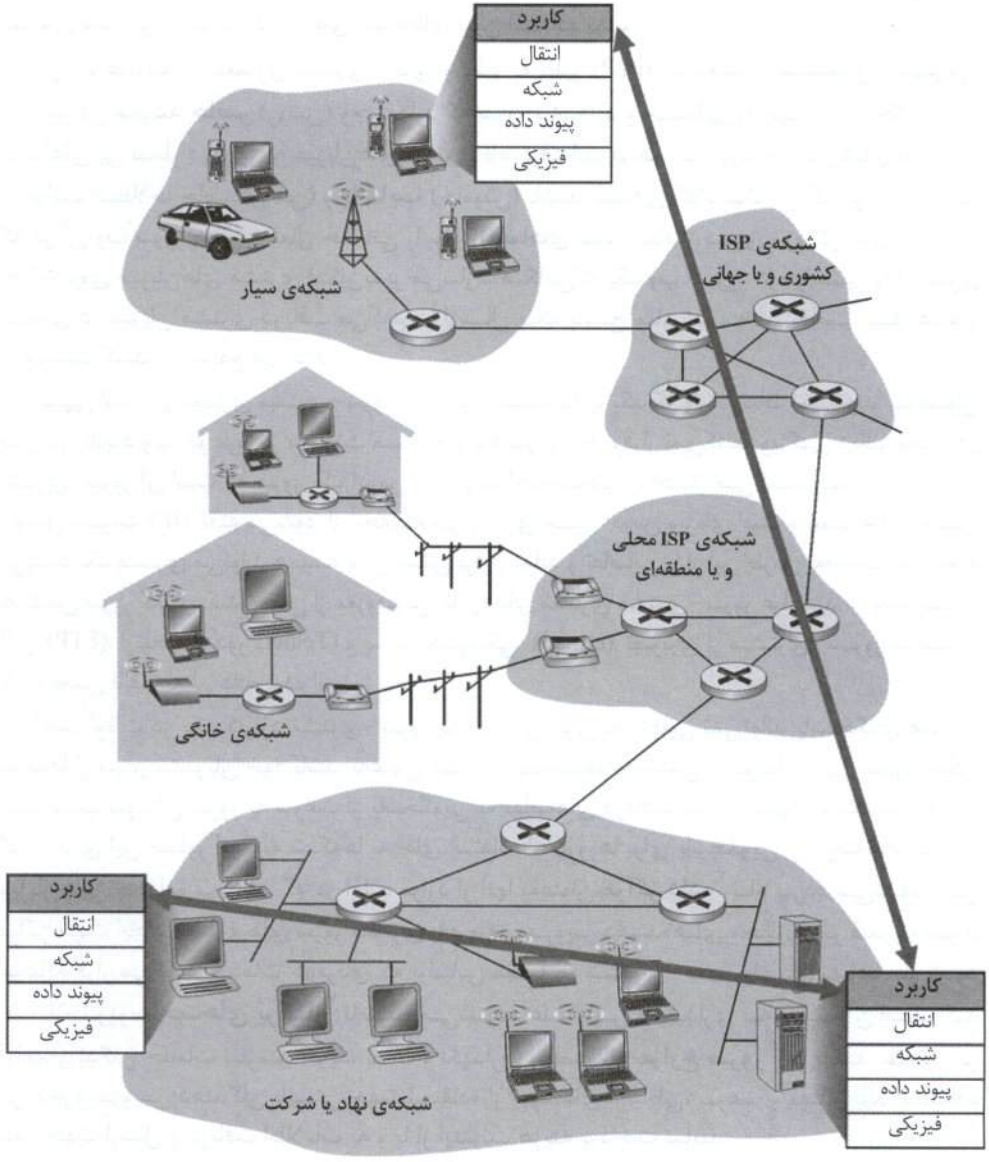
در این فصل اصول پیاده‌سازی همه‌ی این موارد بررسی خواهد شد. در آغاز اصول بنیادی لایه‌ی کاربرد شامل خدمات شبکه که مورد نیاز کاربردها، مشتری‌ها، سرورها، پردازشگر و واسط‌های لایه انتقال است، بررسی می‌شود. چندین نرم‌افزار کاربردی شبکه شامل وب، پست الکترونیکی، DNS، توزیع نظیر به نظیر (P2P) و نیز تلفن اینترنت P2P از جمله‌ی مواردی هستند که با جزئیات کامل تحلیل می‌شوند. رویکردهای توسعه این کاربردها از نقطه‌نظر TCP و UDP به‌ویژه **واسط‌های برنامه نویسی کاربردی**[2] (API) و نیز کاربردهای مشتری- سرور در زبان برنامه‌نویسی جاوا و پیتون از جمله موارد پوشش‌یافته در این فصل است. لایه‌ی کاربرد می‌تواند نقطه‌ی آغاز و مؤثری برای مطالعه‌ی پروتکل‌ها به‌شمار آید.

۲- ۱ اصول کاربردهای شبکه

فرض کنید به دنبال طراحی یک نرم‌افزار کاربردی می‌باشید که سرانجام آن تحولی مادی و معنوی را برای شما به ارمغان می‌آورد. هسته‌ی توسعه و نوآوری هر برنامه‌ی کاربردی شبکه، نگارش برنامه‌ای قابل اجرا بر روی سیستم‌های انتهایی مختلف می‌باشد که ارتباط آنها را با یکدیگر بر روی شبکه میسر می‌سازد. به‌عنوان مثال، در کاربرد وب دو برنامه‌ی متمایز هستند که با یکدیگر ارتباط برقرار می‌کنند. این دو عبارتند از **برنامه‌ی مرورگر**[3] که بر روی میزبان کاربر (کامپیوتر شخصی، کامپیوتر همراه، تلفن همراه و غیره) اجرا می‌شود و برنامه‌ی وب سرور که بر روی سرور میزبان وب در حال پردازش است. مثال دیگر سیستم اشتراک فایل P2P است که برنامه‌ای را در هر یک از میزبان‌هایی که در انجمن اشتراک فایل عضویت دارند، به اجرا در می‌آورد. در این شرایط برنامه‌ها در میزبان‌های مختلف همگی مشابه یکدیگر می‌باشند. هنگامی‌که یک طراح کاربرد جدیدی را نوآوری می‌کند، بایستی برنامه‌ای را بنویسد که بر روی چندین سیستم انتهایی قابل اجرا باشد. این نرم‌افزارها را می‌توان به زبان‌های مختلف برنامه‌نویسی نظیر C و جاوا و یا Python نوشت. آنچه مهم است برنامه‌ی نگارش شده نیازی به اجرا بر روی تجهیزات بنیادی شبکه مثل راهیاب‌های لایه پیوند و یا مسیریاب‌ها را ندارد. حتی اگر منظور نوشتن برنامه‌های کاربردی برای اینگونه تجهیزات باشد. همانگونه که در فصل اول و شکل ۱- ۲۴ نشان داده شد. تجهیزات هسته‌ی شبکه در لایه‌ی کاربرد عمل نمی‌کنند بلکه عملکرد آنها در لایه‌های پائین‌تر است. به‌ویژه در لایه‌ی شبکه و پائین‌تر از آن. این موضوع سبب محدود شدن

1- Internet Protocol Television
2- Application Programming Interface
3- Browser Programs

نرم‌افزارهای کاربردی به سیستم‌های انتهایی می‌شود. شکل ۲- ۱ توسعه و بکارگیری سریع و متنوع کاربردهای شبکه را نشان می‌دهد.



شکل ۲- ۱ ♦ ارتباط برای یک برنامه‌ی کاربردی شبکه بین لایه‌ی کاربرد و سیستم‌های انتهایی انجام می‌شود.

قبل از ورود به کُدنویسی نرم‌افزاری ابتدا بایستی طرح کاملی را برای برنامه‌ی کاربردی مورد نظر تعریف نمود. یادآوری این نکته ضروری است که معماری یک کاربرد کاملاً از معماری شبکه متمایز است. معماری شبکه عبارت است از ساختار پنج لایه‌اینترنت. از نقطه‌نظر یک طراح برنامه‌ی کاربردی

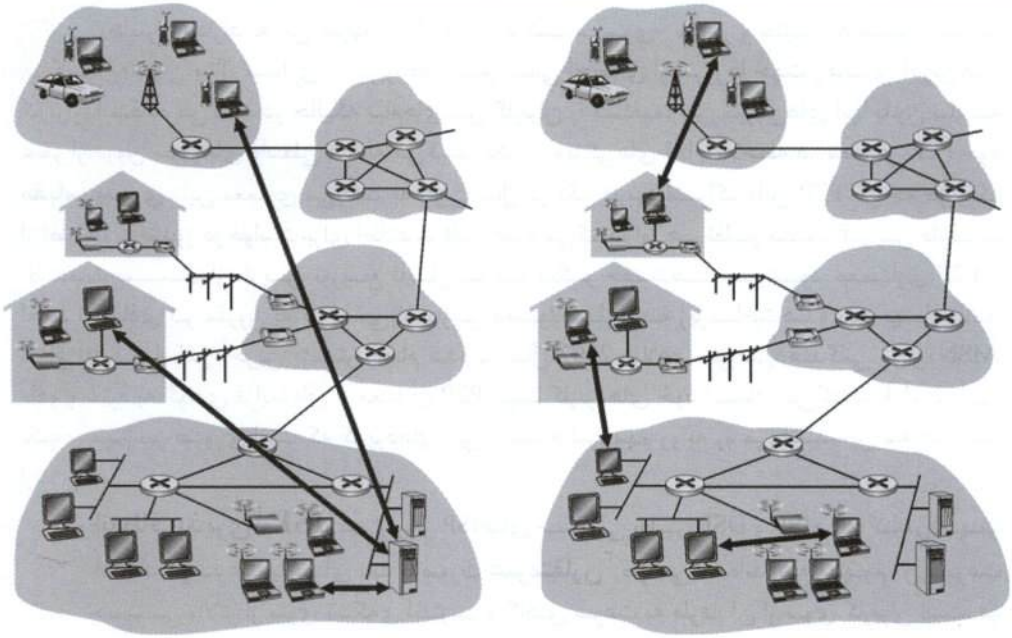
ساختار شبکه ثابت است و مجموعه‌ای از خدمات مشخص را برای یک کاربرد فراهم می‌آورد. **معماری کاربرد**[1] از طرف دیگر به‌وسیله طراح کاربرد تعریف می‌شود و به بیان چگونگی ساختار کاربرد بر روی انواع سیستم‌های انتهایی می‌پردازد. در گزینش معماری کاربرد، احتمال دارد طراح از یکی از دو نظریه‌ی معماری غالب در کاربردهای شبکه‌های نوین استفاده نماید.

این دو عبارتند از: معماری مشتری- سرور و نظیر به نظیر (P2P). در معماری **مشتری- سرور**[2] یک میزبان همیشه **حاضر** (روشن) وجود دارد که سرور نام دارد و خدماتی را جهت پاسخگویی به تقاضاهای بی شمار دیگری که میزبان‌های مشتری نام گرفته‌اند، فراهم می‌آورد. میزبان‌های مشتری می‌توانند اصطلاحاً حاضر (روشن) و یا **غایب** (خاموش) باشند. نمونه‌ی کلاسیک آن کاربرد وب است که در آن وب سرور همیشه فعال خدماتی را برای تقاضاهای صادر شده از سوی مرورگرهای در حال اجرا بر روی میزبان‌های مشتری امکان‌پذیر می‌سازد. هنگامی‌که یک وب سرور درخواستی را از سوی عنصری در میزبان مشتری دریافت می‌کند با ارسال یک پاسخ مناسب به آن تقاضا، نیاز عنصر درخواست کننده را مرتفع می‌سازد.

بدیهی است در معماری مشتری- سرور، مشتری‌ها مستقیماً با یکدیگر وارد ارتباط نمی‌شوند. به‌عنوان مثال در کاربرد وب، دو مرورگر به‌صورت مستقیم با یکدیگر ارتباط برقرار نمی‌کنند. ویژگی دیگر معماری مشتری- سرور آن است که سرور یک آدرس ثابت و شناخته شده‌ای را اختیار می‌کند، که به آن آدرس پروتکل اینترنت (IP) گفته می‌شود. از آنجائیکه سرور دارای چنین آدرس ویژه‌ای است و همیشه نیز روشن می‌باشد یک مشتری می‌تواند همیشه با آن تماس برقرار نماید و تقاضایی را از این طریق به‌صورت یک بسته به آدرس سرور گسیل کند. بعضی از معروفترین کاربردهای معماری مشتری- سرور عبارتند از: وب، انتقال فایل (FTP)، ارتباط راه دور (Telnet) و پست الکترونیکی (e-mail). تصویری از مشتری- سرور در شکل ۲- ۲ بخش (الف) نشان داده شده است.

اغلب اوقات در یک کاربرد مشتری- سرور یک میزبان سرور به تنهایی نمی‌تواند پاسخگوی همه‌ی تقاضاها از سوی مشتریان خود باشد. به‌عنوان مثال در یک شبکه اجتماعی با طرفداران بی شمار، ممکن است سبب شود آن سرور به سرعت از پاسخگویی به تقاضاهای دریافت شده از سوی مشتریان عاجز گردد. برای این منظور اینگونه شبکه‌ها به جای استفاده از سرورها برای پاسخگویی از خوشه‌های بزرگ میزبان‌هایی استفاده می‌کنند که در اغلب موارد از آنها به‌عنوان **مراکز داده**[3] نام برده می‌شود. این مراکز جهت ایجاد معماری‌های سرور مشتری به‌صورت سرویس‌دهنده‌های مجازی پر قدرت مورد استفاده قرار می‌گیرد. خدمات کاربردی که براساس معماری مشتری- سرور بنیان نهاده می‌شود اصطلاحاً **زیرساخت‌های پُرکار**[4] نامیده می‌شوند. علت این نامگذاری به سبب آن است که فراهم‌آورندگان خدمات نیازمند خرید، نصب و نگهداری به اصطلاح **مزارع سرور**[5] می‌باشند. علاوه بر این موارد سرویس دهندگان بایستی جهت استفاده از پیوندهای ارتباطی، سرعت و پهنای باند استفاده شده جهت ارسال و دریافت اطلاعات به و یا از اینترنت هزینه پرداخت نمایند.

1- Application architecture
2- Client-server architecture
3- Data Centers
4- Infrastructure intensive
5- Server farms



(ب) معماری نظیر به نظیر (الف) معماری مشتری - سرور

شکل ۲- ۲ ♦ (الف) معماری مشتری- سرور. (ب) معماری نظیر به نظیر

خدمات اینترنتی مشهور مانند موتورهای جستجوی گوگل، شبکه‌های اجتماعی مانند: **فضای من**[1]، **فیس بوک**[2]، تجارت از طریق اینترنت نظیر آمازن و ایی- بی، پست الکترونیکی بر پایه‌ی وب همانند **پست یاهو**[3] و خدمات اشتراک فایل‌های تصویری نظیر یوتیوپ و اسکایپ همگی از جمله زیرساختهای پُرکار و هزینه بر می‌باشند.

در معماری **نظیربه‌نظیر**[4] استفاده و یا وابستگی به سرورهای زیرساخت به حداقل خود می‌رسد. در این معماری نرم‌افزارهای کاربرد حداکثر بهره را از توان زوج میزبان‌هایی می‌برند که مستقیماً مرتبط با یکدیگر می‌باشند. در اصطلاح شبکه‌این میزبان‌ها را **جفت**[5] یکدیگر می‌نامند. این جفت‌ها معمولاً در مالکیت سرویس‌دهنده قرار ندارند بلکه ابزارهای ارتباطی مثل کامپیوترهای شخصی و همراه هستند که از سوی کاربران خود کنترل می‌شوند. بسیاری از این جفت‌ها در خانه‌ها، دانشگاه‌ها و مراکز اداری قرار دارند. از آنجائیکه این جفت‌ها جهت ارتباط با یکدیگر از درون یک یا چند سرور ویژه عبور نمی‌کنند به آنها نظیر به نظیر گفته می‌شود. اینگونه کاربردها عبارتند از پروتکل توزیع فایل مانند **بیت ترنت**[6]، اشتراک فایل همانند eMule و LimWire، تلفن اینترنتی نظیر **اسکایپ**[7] و رسانه‌ی تصویری IPTV شکل ۲- ۲ بخش (ب) معماری P2P را نشان می‌دهد. قبلاً اشاره گردید بعضی از کاربردها معماری

1- My Space
2- Facebook
3- Yahoo Mail
4- P2P architecture
5- Pairs
6- Bit torrnet
7- Skype

ترکیبی (هایبرید) دارند به این مفهوم که از هر دو ترکیب مشتری- سرور و نظیر به نظیر استفاده می‌کنند. به‌عنوان مثال بسیاری از کاربردهای پیام رسانی لحظه‌ای از سرورها جهت رهگیری آدرس‌های کاربران استفاده می‌کنند در حالیکه پیام‌های بین کاربران را مستقیماً بین میزبان‌های آنها بدون نیاز به عبور از درون سرورهای واسطی مبادله می‌کنند. یکی از ویژگی‌های قابل ملاحظه در معماری P2P خود **مقیاس‌پذیری**[1] این معماری می‌باشد. به‌عنوان مثال در یک کاربرد اشتراک فایل P2P اگر چه هر یک از نظیرها به صدور درخواست برای اطلاعات فایل اقدام می‌کنند، اما هر نظیر سبب افزایش ظرفیت خدمات سیستم از طریق توزیع فایل‌ها به دیگر جفت‌ها می‌شود. معماری P2P از لحاظ اقتصادی نیز مقرون به صرفه می‌باشد زیرا معمولاً نیازی به زیرساخت‌ها و نیز پهنای باند سرور ندارد. براساس آخرین پژوهش انجام شده در سال 2007 میلادی سرویس‌دهندگانی مثل MSN، یاهو و غیره به گونه‌ی فزاینده‌ای از معماری P2P جهت کاربردهای خود استفاده می‌کنند. یادآوری این نکته‌ی مهم نیز ضروری است که کاربردهای آتی با سه چالش مهم رو به رو می‌باشند، این سه عبارتند از:

۱- **انعطاف‌پذیری ISP**[2]: بسیاری از ISPهای مسکونی نظیر DSL و ISPهای کابلی جهت استفاده از سرعت یا پهنای باند به‌صورت **غیرمتقارن**[3] طراحی شده‌اند که مفهوم آن سرعت دسترسی بالاتر از سوی شبکه‌ی اینترنت و کاهش سرعت به طرف آن از سوی کاربران است. در کاربردهای توزیع فایل و یا پخش تصاویر زنده به‌صورت P2P ترافیک به سمت اینترنت از سوی سرورها به طرف کاربران خانگی هدایت می‌شود. نتیجه‌ی آن فشار بار قابل ملاحظه بر روی ISPها خواهد بود. بنابراین کاربردهای آتی P2P بایستی به گونه‌ای طراحی شوند که سبب کاهش بار بر روی ISPها و در نتیجه انعطاف‌پذیری آنها گردند.

۲- **امنیت:** به دلیل ماهیت توزیع پذیری بسیار بالا و طبیعت آزاد کاربردهای P2P، این کاربردها می‌توانند چالش مهمی برای امنیت شبکه به‌شمار آیند.

۳- **سودمندی:** موفقیت آتی کاربردهای P2P بستگی زیاد به متقاعد نمودن کاربران از لحاظ حجم و میزان قابل استفاده از سرعت، امکانات ذخیره‌سازی و منابع پردازش برای کاربردها دارد. این نکته به مفهوم طراحی به همراه بازدهی بالا می‌باشد.

۲-۲ محاوره‌ی پردازش‌ها

قبل از واقعیت بخشیدن به برنامه‌های کاربردی شبکه ابتدا بایستی درک کاملی از چگونگی ارتباط و محاوره‌ی برنامه‌های در حال اجرا بر روی چندین سیستم انتهایی به دست آورد. در ادبیات سیستم‌های عامل برنامه‌ها با یکدیگر محاوره نمی‌کنند بلکه **پردازش‌ها**[4] هستند که در واقع عمل گفتگو و ارتباط با یکدیگر را انجام می‌دهند. تصویری که از یک پردازش می‌توان در ذهن خلق نمود عبارت از برنامه‌ای است که در درون و بین یک سیستم انتهایی در حال اجرا می‌باشد. هنگامی‌که پردازش‌ها بر روی یک سیستم در حال اجرا می‌باشند آنها می‌توانند با یکدیگر و نیز با دیگر پردازش‌های مابین خود ارتباط

1- Self Scalability
2- ISP friendly
3- Asymmetrical
4- Processes

برقرار کنند و جهت انجام این عمل از قوانین و فرآیندهایی که سیستم عامل آن میزبان تعیین می‌کند، استفاده نمایند. البته در این کتاب ما در پی آشکارسازی این نکته که پردازش‌ها در میزبان‌های مشابه چگونه با یکدیگر ارتباط برقرار می‌کنند، نمی باشیم. بلکه به دنبال چگونگی اجرای پردازش‌ها در میزبان‌های گوناگون و با سیستم عامل‌های مختلف از نقطه‌نظر ارتباط با یکدیگر هستیم.

پردازش‌های موجود بر روی دو سیستم انتهایی مختلف، ارتباط با یکدیگر را از طریق مبادله‌ی **پیام‌ها**[1] در فراسوی شبکه‌های کامپیوتری انجام می‌دهند. یک پردازش فرستنده، پیام‌هایی را خلق و سپس با وارد نمودن به شبکه آنها را ارسال می‌کند و در مقابل یک پردازش گیرنده این پیام‌ها را دریافت و احتمالاً پاسخی را با ارسال یک پیام به فرستنده می‌دهد.

شکل ۲- ۱ ارتباط پردازش‌ها با یکدیگر را از طریق استفاده از لایه‌ی کاربرد در پنج لایه‌ی پشته‌ی پروتکلی نشان می‌دهد.

پردازش‌های مشتری- سرور به مفهوم آن است که معمولاً یک کاربرد شبکه شامل جفتی از پردازش‌ها می‌باشد که پیام‌هایی را از طریق شبکه بین خود مبادله می‌کنند. به‌عنوان مثال در کاربرد وب یک پردازش مشتری پیام‌هایی را با یک پردازش وب سرور مبادله می‌کند. در یک سیستم اشتراک فایل P2P یک فایل از یک پردازش در یکی از نظیرها به پردازشی در نظیر دیگر منتقل می‌شود. برای هر جفت پردازش در حال تعامل یکی از دو پردازش دارای نشانه‌ی مشتری (client) و پردازش دیگر دارای برچسب سرور می‌گردد. در وب یک برنامه‌ی مرورگر پردازش مشتری و یک وب سرور مانند گوگل یک پردازش سرور می‌باشد. در فرآیند P2P نظیری که در حال دریافت اطلاعات فایل است برچسب مشتری و همتای آن که در حال بارگذاری اطلاعات یک فایل است نشانه‌ی سرور را به خود اختصاص می‌دهد. بدیهی است در بعضی از کاربردها مثل اشتراک فایل P2P یک پردازش می‌تواند هر دو نقش مشتری و سرور را نیز داشته باشد، زیرا که یک پردازش در یک سیستم اشتراک فایل P2P می‌تواند اطلاعات فایل‌ها را هم بارگذاری و هم دریافت نماید. در چارچوب هر فرآیند ارتباطی بین یک جفت پردازش هنوز هم برچسب یک پردازش مشتری و دیگری سرور خواهد بود. بنابراین پردازش‌های مشتری- سرور را از این به بعد به‌صورت زیر تعریف می‌کنیم:

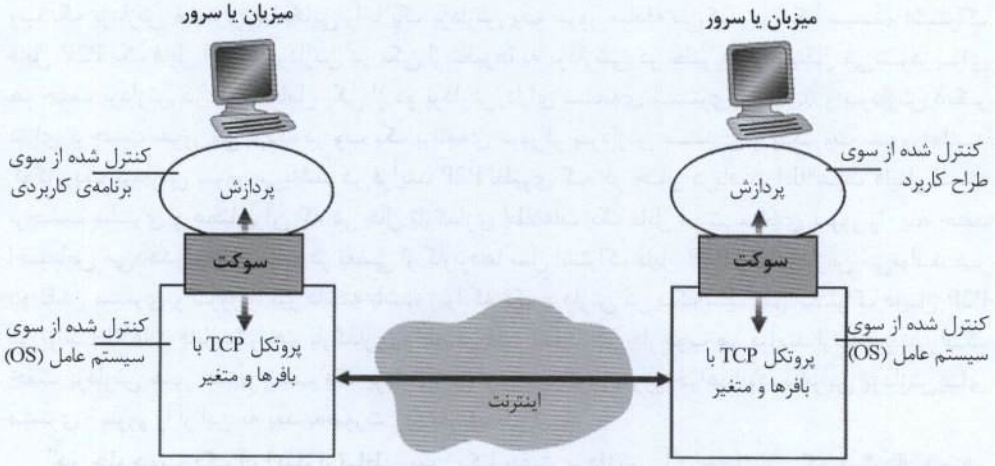
"در چارچوب یک فرآیند ارتباطی بین یک جفت پردازش، آن پردازشی که پیش قدم در ارتباط است دارای برچسب مشتری و پردازشی که در حال انتظار جهت اتصال و آغاز فرآیند مبادله است نشان سرور را به خود می‌گیرد."

در وب پردازش‌های مرورگر وب سرور با یکدیگر تماس برقرار می‌کنند. بنابراین پردازش مرورگر مشتری و پردازش وب سرور یک سرویس‌دهنده (سرور) به‌شمار می‌آید. در سیستم اشتراک فایل P2P هنگامی‌که نظیر A درخواست ارسال یک فایل مشخص را از نظیر B می‌کند، آنگاه A مشتری و B یک سرور است. بدیهی است در چارچوب یک فرآیند ارتباطی مشخص این نامگذاری‌ها مفهوم خود را آشکار می‌سازد. موضوع بعدی در این بخش بررسی واسط بین پردازش و شبکه‌های کامپیوتری است.

براساس مطالب عنوان شده در آغاز این فصل بسیاری از کاربردها شامل جفت پردازش‌های ارتباطی می‌باشند. این جفت پردازش‌ها هستند که در هر جفت سیستم انتهایی نسبت به ارسال پیام‌ها به یکدیگر اقدام می‌کنند. هر پیام ارسالی از یک پردازش به پردازش دیگر بایستی از درون شبکه‌ی خود عبور نماید. یک پردازش پیام‌های ارسالی و یا دریافتی را از طریق یک نرم‌افزار واسط شبکه که اصطلاحاً

1- Messages

به آن **سوکت**[1] گفته می‌شود از درون شبکه عبور می‌دهد. برای درک بهتر موضوع به این مثال توجه نمائید: در ذهن خود یک پردازش را همانند یک خانه یا آپارتمان تصور کنید و در ورودی آنرا یک سوکت. هنگامی‌که یک پردازش می‌خواهد پیامی را به سوی پردازش دیگری به میزبانی غیر از خود ارسال نماید او پیام‌های خود را از طریق دَر ورودی و خروجی آپارتمان یا خانه (سوکت) خارج می‌نماید. البته این فرآیند ارسال فرض را بر آن گذارده است که یک زیرساخت انتقال در طرف دیگر در ورودی و خروجی وجود دارد که این پیام را به در ورودی و خروجی آپارتمان پردازش مقصد تحویل می‌دهد. هنگامی‌که پیام وارد میزبان مقصد می‌شود پیام از درون سوکت (در ورودی و خروجی) پردازش دریافت کننده عبور نموده و برنامه‌ی پردازش دریافت بر روی آن پیام عمل می‌نماید. شکل ۲- ۳ ارتباط بین دو پردازش از طریق سوکت‌ها بر روی شبکه‌ی اینترنت را نشان می‌دهد. در این شکل فرض بر آن است که پروتکل انتقال استفاده شده به‌وسیله‌ی پردازش همان پروتکل TCP اینترنت می‌باشد. همانگونه که شکل ۲- ۳ نشان می‌دهد یک سوکت واسط بین لایه‌ی کاربرد و لایه‌ی انتقال در بین یک میزبان قرار دارد.



شکل ۲- ۳ ♦ پردازش‌های کاربرد، سوکت‌ها و پروتکل انتقال

این فرآیند را واسط برنامه‌نویسی کاربردی (API) می‌نامند. این واسط مابین یک کاربرد و شبکه قرار می‌گیرد زیرا که سوکت یک واسط برنامه‌نویسی است که از طریق آن کاربردهای شبکه طراحی و ساخته می‌شوند. بنابراین طراح هر برنامه‌ی کاربردی کنترل کامل را بر روی هر عنصری در سمت لایه‌ی کاربرد سوکت اعمال می‌کند. اما کنترل او بر روی لایه‌ی انتقال در سمت دیگر سوکت بسیار ناچیز است. در حقیقت تنها کنترلی که طراح یک کاربرد بر روی سمت لایه‌ی انتقال دارد عبارت از: (۱) حق انتخاب پروتکل و (۲) تعریف و تعیین چند عنصر وابسته به لایه‌ی انتقال مانند حداکثر اندازه‌ی حجم بافر و حداکثر اندازه‌ی طول هر قطعه می‌باشد.

هنگامی‌که یک طراح پروتکل انتقال را در صورت اختیار گزینش نمود. آنگاه برنامه‌ی کاربردی براساس استفاده از خدمات لایه‌ی انتقال که آن پروتکل فراهم می‌آورد، طراحی و ساخته می‌شود.

1- Socket

۲-۳ خدمات انتقال در دسترس کاربردها

با یادآوری این نکته که یک سوکت، واسط بین پردازش کاربردی و پروتکل لایه‌ی انتقال است نتیجه‌گیری می‌شود که کاربرد در طرف ارسال، پیام‌ها را به درون سوکت گسیل می‌دارد و در سمت دیگر همان سوکت پروتکل لایه انتقال مسئولیت تحویل پیام‌ها به سوکت دریافت را برعهده می‌گیرد. بسیاری از شبکه‌های کامپیوتری و از جمله اینترنت دارای بیش از یک پروتکل در لایه انتقال می‌باشند. بنابراین هنگامی‌که طراحی یک کاربرد در حال انجام است بایستی نسبت به گزینش یکی از این پروتکل‌ها در لایه انتقال تصمیم‌گیری کرد. چگونه این تصمیم‌گیری یا به عبارت دیگر گزینش انجام می‌شود؟ به احتمال زیاد یک طراح ابتدا به مطالعه‌ی خدماتی که پروتکل‌های لایه انتقال فراهم می‌آورند، پرداخته و سپس یکی از آنها که بهترین سازگاری را از لحاظ نیازهای آن برنامه کاربردی دارد گزینش و به این مرحله از طراحی پایان می‌دهد. این وضعیت را می‌توان مشابه عملیات گزینش ترن یا هواپیما برای مسافرت بین دو شهر تصور نمود. مسافر یکی از این دو را با توجه به اینکه هر یک خدمات متفاوتی را ارائه می‌دهند، بایستی گزینش نماید. به‌عنوان مثال ترن به دلیل توقف بین ایستگاه‌ها زمان طولانی‌تر را نسبت به مسافرت هوایی در پی دارد. در ارتباط با خدمات لایه‌ی انتقال برای کاربردها می‌توان به چهار مورد اساسی اشاره نمود: **انتقال مطمئن داده‌ها**[1]، **نهفتگی**[2] (بازدهی)، **زمانبندی**[3] و **امنیت**[4].

- **انتقال مطمئن داده‌ها**

در فصل اول اشاره گردید بسته‌ها در بین یک شبکه ممکن است تلف شوند. به‌عنوان مثال یک بسته می‌تواند سبب سرریزی بافری در یک روتر شود و یا آنکه از سوی یک میزبان یا مسیریاب پس از آنکه یک یا چند بیت آن آلوده گردید حذف گردد. برای بسیاری از کاربردها همانند پست الکترونیکی، انتقال فایل، دستیابی راه دور به میزبان، انتقال اسناد وب و کاربردهای تجاری اتلاف داده‌ها زیان جبران ناپذیری را به همراه دارد به‌ویژه در ارتباط با آخرین مورد که زیان می‌تواند متوجه هر دو طرف معامله یعنی مشتری و یا بانک شود. بنابراین برای پشتیبانی این خدمات، ضمانت‌های لازم بایستی فراهم شود تا آنکه برنامه‌ی کاربردی در یک طرف داده‌ها را به گونه‌ای مطلوب ارسال و کاربرد طرف مقابل آنها را به طور کامل دریافت نماید. اگر یک پروتکل چنین خدمات مطمئنی را برای تحویل داده‌ها فراهم آورد آن پروتکل برخوردار از امکانات انتقال مطمئن داده‌ها می‌باشد. یکی از مهمترین خدماتی که یک پروتکل لایه انتقال می‌تواند تعریف کند عبارت از یک کاربرد پردازش به پردازش انتقال مطمئن داده‌ها می‌باشد. هنگامی‌که یک پروتکل انتقال چنین امکاناتی را فراهم کند آنگاه پردازش فرستنده کافی است داده‌ها را به سوکت با اطمینان کامل از اینکه آنها بدون خطا تحویل پردازش دریافت کننده می‌شوند، تحویل نماید. هنگامی‌که یک پروتکل لایه‌ی انتقال چنین خدماتی را فراهم نمی‌آورد این امکان وجود دارد که داده‌ی ارسالی توسط پردازش فرستنده هرگز به گیرنده تحویل نشود. این موضوع ممکن است برای آن دسته از کاربردهایی که تا حدودی تحمل اتلاف بسته‌ها را دارند قابل قبول باشد. نمونه‌های این نوع نرم‌افزارهای کاربردی عبارتند از: کاربردهای چند رسانه‌ای مثل خدمات صوتی و تصویری

1- Reliable transfer of data
2- Throughput
3- Timing
4- Security

بی‌درنگ و یا سرویس ذخیره‌ی صوت و تصویر که اتلاف بخشی از داده‌ها برای آنها قابل تحمل است. در کاربردهای چند رسانه‌ای اتلاف داده‌ها ممکن است سبب یک پرش یا بریدگی کوتاه به هنگام پخش صدا و یا تصویر گردد که چندان تأثیری در مفهوم کلی اطلاعات ندارد.

- **نهفتگی یا بازدهی**

در فصل اول اصول بازدهی یا نهفتگی قابل واگذاری از سوی شبکه‌های کامپیوتری معرفی گردید. در چارچوب یک فرآیند ارتباط بین دو پردازش در مسیر یک شبکه، نهفتگی عبارتست از سرعتی که پردازش در یک طرف می‌تواند بیت‌ها را تحویل پردازش طرف مقابل نماید. از آنجائیکه پردازش‌های دیگر نیز از پهنای باند در دسترس در طول مسیر شبکه به گونه‌ی مشترک استفاده می‌نمایند و نیز این فرآیندها حالت ثابتی را از لحاظ آمد و شد ندارند. بنابراین بازدهی در دسترس می‌تواند در هر فاصله از زمان متغیر باشد. این ملاحظات سبب می‌شود که سرویس دیگر پروتکل لایه‌ی انتقال نیز مشخص گردد. آن سرویس عبارت است از ضمانت بازدهی یا همان نهفتگی در دسترس در یک سرعت یا پهنای باند ویژه. با چنین امکاناتی یک کاربرد می‌تواند یک بازدهی ضمانت شده‌ی r بیت بر ثانیه را تقاضا نماید و پروتکل لایه‌ی انتقال نیز این اطمینان را ایجاد می‌کند که این بازدهی همیشه به میزان حداقل r بیت بر ثانیه خواهد بود. چنین سرویس بازدهی ضمانت شده‌ای علاقه مندان بسیاری را در بین کاربردها به همراه خواهد داشت. به‌عنوان مثال اگر یک کاربرد تلفن اینترنتی رمزگذاری صدا را در سرعت 32Kbps انجام دهد، آنگاه نیازمند ارسال داده‌ها در شبکه و نیز تحویل آنها به کاربرد دریافت کننده با همین سرعت می‌باشد. اگر پروتکل انتقال قادر به تأمین چنین بازدهی نباشد، کاربرد بایستی رمزگذاری صدا را در سرعت پائین‌تر انجام دهد و در نتیجه بازدهی دریافتی را در مقیاسی حفظ نماید که رمزگذاری در سرعت پائین امکان‌پذیر باشد در غیر اینصورت، بایستی انجام این فرآیند را فراموش نماید زیرا دریافت نیمی از بازدهی مورد نیاز عملاً در کاربرد تلفن اینترنتی بی فایده است. کاربردهای نیازمند بازدهی را در واژگان شبکه کاربردهای **حساس در برابر پهنای باند**[1] می‌نامند. بسیاری از کاربردهای چند رسانه‌ای حساس در برابر پهنای باند می‌باشند. بعضی از این کاربردها ممکن است از روش‌های رمزگذاری با قابلیت سازگاری استفاده نمایند که نتیجه‌ی آن انجام عملیات رمزگذاری با سرعت سازگار با بازدهی در دسترس می‌باشد. بدیهی است کاربردهای حساس در برابر پهنای باند نسبت به بازدهی شبکه بسیار وابسته می‌باشند، امّا در مقابل **کاربردهای انعطاف‌پذیر**[2] می‌توانند از هر اندازه بازدهی که شبکه می‌تواند در اختیار آنها قرار دهد، بهره گیرند. به عبارت دیگر این بهره‌گیری می‌تواند در یک محدوده‌ی حداقل و یا حداکثر متغیر باشد. پست الکترونیکی، انتقال فایل و انتقال از طریق وب از جمله‌ی کاربردهای انعطاف‌پذیر به‌شمار می‌آیند. بدیهی است هرچه بازدهی بیشتر باشد نتیجه‌ی حاصل مطلوبتر است.

- **زمانبندی**

یک پروتکل لایه‌ی انتقال می‌تواند زمانبندی ضمانت شده‌ای را فراهم آورد. همانند بازدهی ضمانت شده، زمانبندی مطمئن در قالب روش‌ها و چارچوب‌های مختلف امکان‌پذیر است. نمونه‌ای از این ضمانت ممکن است به اینگونه باشد که هر بیت ارسالی از سوی فرستنده به درون سوکت حداکثر پس از گذشت زمان تأخیری معادل 100 میلی ثانیه به سوکت دریافت کننده وارد گردد. چنین خدماتی در

1- Bandwidth Sensitive applications
2- Elastic applications

بین کاربردهای زمان واقعی مانند تلفن اینترنتی، محیط‌های مجازی کنفرانس‌های تلفنی و بازیهای تیمی همزمان علاقه‌مندان بسیاری دارند. همگی این موارد نیازمند رعایت یک زمانبندی مؤثر برای تحویل داده‌ها می‌باشند. تأخیر طولانی در سرویس تلفن اینترنتی سبب توقف‌های غیر طبیعی در طول مکالمه خواهد شد. در بازی‌های تیمی یا محیط‌های محاوره‌ی مجازی وقوع تأخیرهای طولانی بین انجام عملیات و مشاهده‌ی پاسخ آنها از سوی محیط مانند بازیکنان دیگر و یا در یک ارتباط انتها به انتها، کاربرد را از لحاظ عملی در برابر چشمان کاربران غیر واقعی جلوه می‌دهد. برای کاربردهای غیر زمان واقعی تأخیرهای کوتاه همواره بر تأخیرهای طولانی ترجیح داده می‌شوند اگرچه هیچگونه محدودیتی از لحاظ تأخیرهای انتها به انتها اعمال نگردد.

- **امنیت**

یک پروتکل لایه‌ی انتقال می‌تواند یک یا چند امکان امنیتی را برای یک کاربرد فراهم آورد. به‌عنوان مثال در یک میزبان فرستنده یک پروتکل انتقال می‌تواند همه‌ی داده‌ها را از طریق برنامه‌ی پردازش فرستنده رمزگذاری نماید و در میزبان گیرنده پروتکل لایه‌ی انتقال داده‌های دریافتی را قبل از تحویل به پردازش دریافت کننده رمزگشایی نماید. ارائه‌ی چنین خدماتی احساس اطمینان را بین پردازش‌های فرستنده و گیرنده ایجاد می‌نماید. اگرچه ممکن است داده‌ها به گونه‌ای بین پردازش‌های فرستنده و گیرنده از طریق شخص ثالث قابل رویت باشند. یک پروتکل لایه‌ی انتقال علاوه بر انجام خدمات محرمانه‌سازی داده‌ها، عملیاتی چون رعایت یکپارچگی داده‌ها و اعتبارسنجی نقطه‌ی انتها را نیز می‌تواند امکان‌پذیر نماید.

۲- ۴ خدمات انتقال در اینترنت

تا اینجا خدمات انتقال که یک شبکه‌ی کامپیوتری می‌تواند به‌صورت کلی تأمین نماید بررسی گردید. از این به بعد انواع ابزارهای پشتیبانی کاربردها که اینترنت می‌تواند آنها را فراهم نماید، بررسی می‌شوند. شبکه‌ی اینترنت و از دید کلی تر شبکه‌های TCP/IP دو پروتکل انتقال را در دسترس کاربردها قرار می‌دهد. این دو عبارتند از UDP و TCP. هنگامی‌که طراح، یک برنامه‌ی کاربردی جدید را برای اینترنت خلق می‌کند یکی از نخستین تصمیم‌گیری‌های وی استفاده از UDP و یا TCP در آن برنامه می‌باشد. هر یک از این دو پروتکل مجموعه‌ای از خدمات مختلف را برای فعال‌سازی کاربردها ارایه می‌دهند. شکل ۲- ۴ نیازهای خدماتی برخی از کاربردها را نشان می‌دهد.

کاربرد	اتلاف داده	پهنای باند	حساس در برابر زمان
انتقال فایل	بدون اتلاف	انعطاف‌پذیر	خیر
پست الکترونیکی (ایمیل)	بدون اتلاف	انعطاف‌پذیر	خیر
اسناد وب	بدون اتلاف	انعطاف‌پذیر (چند kpbs)	خیر
تلفن اینترنت یا کنفرانس تصویری	تحمل‌پذیر در برابر اتلاف	صدا: چند kbps تا 1mbps تصویر: 10kbsp تا 5Mbps	آری: صدها میلی‌ثانیه
صدا و تصویر ذخیره‌سازی شده	تحمل‌پذیر در برابر اتلاف	مشابه بالا	آری: چند ثانیه
بازی‌های تعامل گرار	تحمل‌پذیر در برابر اتلاف	چند kbps تا 10kpbs	آری: صدها میلی ثانیه
پیام بی‌درنگ	بدون اتلاف	انعطاف‌پذیر	آری و خیر

شکل ۲- ۴ ♦ نیازهای کاربردهای منتخب شبکه

- **خدمات TCP**

الگوی خدمات TCP شامل یک سرویس اتصال‌گرا با انتقال مطمئن داده‌ها می‌باشد. هنگامی‌که یک کاربرد از طریق TCP فعال می‌شود به‌عنوان پروتکل انتقال، آن کاربرد هر دو سرویس را از TCP دریافت می‌کند. سرویس اتصال‌گرا به مفهوم آن است که TCP با استفاده از اطلاعات کنترل لایه‌ی انتقال مشتری و سرور به چگونگی مبادله‌ی اطلاعات بین این دو پی می‌برد. این آگاهی قبل از آنکه پیام‌های سطح کاربرد آغاز به ارسال اطلاعات نمایند روی می‌دهد. این مرحله را اصطلاحاً **دست‌دهی** می‌نامند و جهت آگاه‌سازی مشتری و سرور و ایجاد فرصت کافی برای آنها جهت آماده شدن ارسال و دریافت بسته‌ها می‌باشد. پس از مرحله‌ی دست‌دهی یک ارتباط TCP بین سوکت‌های دو پردازش ایجاد می‌شود. این ارتباط به‌صورت دو طرفه‌ی همزمان می‌باشد و در نتیجه دو پردازش می‌توانند پیام‌ها را از طریق آن همزمان بین یکدیگر مبادله نمایند. هنگامی‌که یک کاربرد به ارسال پیام‌ها خاتمه می‌دهد او بایستی این ارتباط را قطع (پاره) نماید. این سرویس اصطلاحاً **اتصال‌گرا** به جای **ارتباط** نامیده می‌شود زیرا ارتباط دو پردازش دارای استحکام بسیار ضعیفی است. در فصل سوم سرویس اتصال‌گرا و چگونگی پیاده‌سازی آن با جزئیات بیشتر بررسی می‌شود. از طرفی سرویس انتقال مطمئن داده‌ها به مفهوم آن است که دو پردازش در حال ارتباط با تکیه بر توان TCP در تحویل همه‌ی داده‌های ارسالی بدون خطا و با رعایت ترتیب و نظم مناسب با یکدیگر تعامل می‌کنند. هنگامی‌که یک طرف کاربرد جریانی از بایت‌ها را وارد یک سوکت می‌کند می‌تواند این اطمینان را داشته باشد که TCP همان جریان از بایت‌ها را به سوکت گیرنده تحویل می‌نماید بدون آنکه یکی از بایت‌ها تلف و یا نسخه برداری مجدد شده باشد. علاوه بر موارد بالا TCP مکانیزم‌های کنترل تراکم را نیز شامل می‌شود که یک سرویس سودمند برای تمام اینترنت به‌شمار می‌آید. سودمندی آن به‌صورت مستقیم شامل پردازش‌های در حال ارتباط می‌باشد. مکانیزم کنترل تراکم TCP سبب می‌شود هنگامی‌که بین فرستنده و گیرنده تراکم ایجاد می‌شود، هر دو طرف مشتری و سرور مجبور می‌شوند، سرعت خود را کاهش دهند. در فصل سوم ملاحظه خواهید نمود که چگونه کنترل تراکم TCP تلاش می‌کند هر ارتباط TCP را از لحاظ استفاده‌ی مشترک از پهنای باند شبکه محدود نماید. بدیهی است کاهش سرعت انتقال سبب زیان عمیق به کاربردهای بی‌درنگ همانند کاربردهای صوتی و تصویری خواهد شد زیرا آنها نیازمند حداقل نهفتگی ممکن می‌باشند. علاوه بر آن کاربردهای زمان واقعی در برابر اتلاف تحمل پذیری بیشتری دارند و نیازمند خدمات انتقال کاملاً مطمئن نمی‌باشند. به همین سبب طراحان کاربردهای بی‌درنگ اغلب ترجیح می‌دهند کاربردهای آنها بر روی UDP اجرا گردد تا TCP.

- **خدمات UDP**

پروتکل UDP از جمله پروتکل‌هایی است که از لحاظ عملکرد از نهایت سادگی برخوردار است زیرا حداقل امکانات را فراهم می‌آورد. این پروتکل غیر اتصال گراست بنابراین نیازی به دست‌دهی قبل از آنکه دو پردازش آغاز به ارتباط نمایند در خود احساس نمی‌کند. پروتکل UDP انتقال نامطمئن داده‌ها را در خود جای داده است. به عبارت دیگر هنگامی‌که یک پردازش پیامی را به درون سوکت UDP گسیل می‌دارد، هیچگونه ضمانتی از سوی UDP در جهت آنکه پیام‌ها هرگز به پردازش دریافت کننده خواهد رسید، فراهم نمی‌شود. علاوه بر آن پیام‌های وارد شده در پردازش دریافت کننده خارج از ترتیب می‌باشند. مکانیزم‌های کنترل تراکم در UDP تعریف نشده است بنابراین سمت ارسال UDP داده‌ها را می‌تواند با سرعت دلخواه به درون لایه‌ی پائین‌تر یعنی لایه‌ی شبکه گسیل کند. بدیهی است نهفتگی

انتها به انتها ممکن است کمتر از سرعت نامی UDP باشد زیرا تراکم شبکه و یا پیوندهای دیگر ممکن است سبب این محدودیت سرعت شوند. از آنجائیکه کاربردهای بی‌درنگ تحمل اتلاف بسته‌ها را دارند اما نیازمند یک حداقل سرعت مؤثر می‌باشند، طراحان کاربردهای زمان واقعی اجرای برنامه‌های کاربردی خود را بر روی UDP ترجیح می‌دهند تا آنکه با سرآیند بسته و مکانیزم‌های کنترل تراکم TCP مواجه شوند، از طرف دیگر از آنجائیکه ساختار بسیاری از **دیواره‌های آتش**[1] به گونه‌ای پیکربندی شده است که اکثر ترافیک‌های از نوع UDP را مسدود می‌کنند، طراحان به‌صورت فزاینده‌ای کاربردهای زمان واقعی و چند رسانه‌ای را از نقطه‌نظر اجرا بر روی TCP متمرکز می‌کنند. لازم به یادآوری است که هیچیک از دو پروتکل TCP و UDP امکانات **رمزگذاری**[2] را در درون خود ندارند. بنابراین داده‌هایی که از یک پردازش فرستنده عبور می‌کنند و وارد سوکت آن می‌شوند همان داده‌هایی هستند که از طریق شبکه عبور و سرانجام به پردازش مقصد می‌رسند. در نتیجه اگر به طور مثال پردازش ارسال کننده‌یک رمز عبور یا گذر واژه‌ای را به‌صورت متن شفاف یعنی رمز نشده به درون سوکت خود وارد نماید، این رمز عبور عیناً به همان صورت با عبور از همه‌ی پیوندهای شبکه به گیرنده خواهد رسید و این احتمال قوی وجود دارد که در بین راه رمز اشاره شده کشف و سپس رمزگشایی گردد. بنابراین به دلایل محرمانه نگه داشتن اسرار خصوصی و سایر موارد امنیتی این موضوع تبدیل به یک چالش حیاتی برای بسیاری از کاربردها شده است. به همین منظور جامعه‌ی نگران اینترنت، بهینه‌سازی هایی را برای TCP تکمیل نموده است که اصطلاحاً به آنها **لایه‌ی ایمن سوکت‌ها**[3] (SSL)گفته می‌شود. پروتکل ارتقاء یافته‌ی TCP با عامل SSL نه تنها تمام وظایف قبلی TCP را انجام می‌دهد بلکه خدمات امنیت پردازش به پردازش شامل رمزگذاری، حفظ یکپارچگی داده‌ها و اعتبارسنجی نقطه‌ی انتهایی را نیز انجام می‌دهد. تأکید بر این نکته ضروری است که SSL پروتکل سوم انتقال اینترنت در سطح TCP و UDP نیست بلکه مکمل TCP به‌شمار می‌آید. این عامل مکمل البته در لایه‌ی کاربرد پیاده‌سازی شده است.

به‌ویژه اگر یک کاربرد بخواهد از سرویس SSL استفاده نماید بایستی کُدهای برنامه‌نویسی SSL را در درون خود منظور نماید. این کُدها شامل توابع کتابخانه‌ای و کلاسهای پیشرفته‌ی موجود در SSL می‌باشد. بدیهی است هر دو طرف مشتری و سرور در یک نرم‌افزار کاربردی بایستی شامل این کُدها باشند. راهکار SSL، سوکت API ویژه‌ی خود را دارد که مشابه سوکت API پروتکل TCP سنتی است. هنگامی‌که یک کاربرد از SSL استفاده می‌کند، پردازش ارسال کننده داده‌ی خود را به‌صورت متن شفاف (رمز نشده) به سوکت SSL گسیل می‌کند و SSL ساکن در میزبان ارسال کننده داده را رمزگذاری و از درون سوکت TCP عبور می‌دهد. داده‌های رمزگذاری شده از طریق پیوندهای اینترنت پس از عبور به سوکت TCP پردازش گیرنده وارد می‌شوند. سوکت دریافت کننده داده‌های رمز شده را به SSL تحویل می‌دهد که در این نقطه مجدداً رمزگشایی می‌شوند. سرانجام SSL داده‌ها را که به‌صورت متن شفاف درآمده‌اند از طریق عبور از سوکت SSL به پردازش دریافت کننده تحویل می‌دهد. جزئیات بیشتری از SSL در فصل هشتم بررسی خواهد شد.

در ادامه‌ی مطالب این بخش خدماتی که از طریق پروتکل لایه‌ی انتقال اینترنت فراهم نمی‌شوند بررسی می‌گردد. قبلاً اشاره گردید سازماندهی خدمات امکان‌پذیر به‌وسیله‌ی لایه انتقال در چهار

1- Firewalls
2- Encryption
3- Secure Socket Layer

موضوع **انتقال مطمئن داده‌ها، بازدهی، زمانبندی** و **امنیت** خلاصه می‌شدند. کدامیک از این سرویس‌ها به‌وسیله TCP یا UDPفراهم می‌شود؟ در پاسخ به این پرسش ملاحظه گردید که TCP انتقال مطمئن داده‌ها را فراهم می‌آورد و از طریق SSL خدمات امنیت داده را نیز امکان‌پذیر می‌سازد. اما آنچه در مباحث گذشته مشخص نشد دو موضوع زمانبندی و بازدهی یا نهفتگی می‌باشد، خدماتی که از طریق پروتکل‌های انتقال اینترنت در حال حاضر فراهم نمی‌شوند. بنابراین بی‌درنگ این پرسش مطرح می‌شود که آیا کاربردهای حساس در برابر زمان مانند تلفن اینترنتی نمی‌توانند در حال حاضر از امکانات اینترنت بهره گیرند؟ پاسخ قاطع خیر است. شبکه‌ی اینترنت کاربردهای حساس در برابر زمان را سال‌هاست که میزبانی نموده است. این کاربردها از آن جهت نسبتاً به گونه‌ی مطلوب فعلی عمل می‌نمایند که طراحی آنها برای این منظور انجام شده است که بدون نیاز به این ضمانت‌ها عمل کنند. چندین راهکار طراحی از این دست را در فصل هفتم بررسی و به حُقه‌های آنها پی خواهیم برد. اما در هر حال هر طرح هوشمندی دچار محدودیت خواهد شد، آن هم در هنگامی‌که تأخیر از حدّ معینی عبور نماید و این اتفاقی است که در اینترنت عمومی روی می‌دهد. نتیجه آنکه اینترنت در حال حاضر در اغلب موارد خدمات رضایت بخشی را برای کاربردهای حساس در برابر زمان فراهم می‌آورد اما نمی‌تواند ضمانت هر زمانبندی و یا پهنای باندی را فراهم آورد. شکل ۲- ۵ نمونه هایی از پروتکل‌های انتقال را برای مشهورترین کاربردهای اینترنت نشان می‌دهد. ملاحظه می‌شود که کاربردهای پست الکترونیکی، دستیابی راه دور، وب گردی و انتقال فایل همگی از TCP استفاده می‌کنند. این کاربردها از آن جهت TCP را انتخاب نموده‌اند زیرا که TCP خدمات انتقال مطمئن داده‌ها را فراهم می‌آورد با این ضمانت که همه‌ی داده‌ها سرانجام به مقصد خود خواهند رسید. هم چنین از شکل ملاحظه می‌شود که تلفن اینترنتی به‌ویژه بر روی UDP اجرا می‌شود. هر طرف کاربرد تلفن اینترنتی نیازمند ارسال داده‌ها از درون شبکه با حداقل سرعت می‌باشند. شکل ۲- ۴ را ملاحظه نمائید. این نیاز از طریق UDP بهتر تأمین می‌شود تا TCP، هم چنین کاربردهای تلفن اینترنتی در برابر اتلاف بسته‌ها تحمل پذیر می‌باشند و نیازمند انتقال مطمئن داده‌ها هم نمی‌باشند که TCP آنها را فراهم می‌کند.

کاربرد	پروتکل لایه‌ی کاربرد	پروتکل انتقال مؤثر
پست الکترونیکی	پروتکل SMTP [RFC 5321]	TCP
دسترسی راه دور	پروتکل Telnet [RFC 854]	TCP
وب گردی	پروتکل HTTP [RFC 2616]	TCP
انتقال فایل	پروتکل FTP [RFC 959]	TCP
پخش جریان رسانه‌ای	پروتکل HTTP مانند یوتیوب و RTP	TCP یا UDP
تلفن اینترنتی	پروتکل RTP، SIP یا خصوصی اسکایپ	به‌ویژه UDP

شکل ۲- ۵ ♦ کاربردهای مشهور اینترنت، پروتکل‌های لایه‌ی کاربرد و انتقال آنها

پایان‌بخش این قسمت موضوع آدرسی دهی پردازش‌ها می‌باشد. تا به حال تمرکز مطالب بر روی خدمات انتقال بین دو پردازش در حال محاوره بود. اما چگونه یک پردازش مشخص می‌کند که با کدام پردازش می‌خواهد با استفاده از این سرویس‌ها ارتباط برقرار نماید؟ چگونه یک پردازش در حال اجرا بر روی یک میزبان شرکت NKE در تهران ایران مشخص می‌کند که تمایل دارد با پردازش مشخص بر روی یک میزبان در دانشگاه MIT در آمریکا ارتباط برقرار نماید؟ برای شناسایی پردازش دریافت کننده

دو قطعه اطلاعات بایستی مشخص گردد. این دو قطعه اطلاعات ویژه عبارتند از: (۱) نام یا آدرس میزبان و (۲) یک شناسه که پردازش دریافت‌کننده را در میزبان مقصد مشخص نماید. در شبکه‌ی اینترنت میزبان از طریق آدرس IP شناسایی می‌شود. فصل چهارم به جزئیات اینگونه آدرس‌ها می‌پردازد. آنچه در این مقطع مورد نیاز است آن است که یک آدرس IP یک کمیت 32 بیت است که ما می‌توانیم آنرا به‌عنوان یک شناسه‌ی منحصر یک میزبان تلقی نمائیم. البته همانگونه که در فصل ۴ ملاحظه خواهد شد بکارگیری گسترده از **ترجمه‌گرهای آدرس شبکه**[1] (NAT) به مفهوم آن است که در عمل یک آدرس با طول 32 بیت به تنهایی آدرس منحصر به فرد یک میزبان نمی‌باشد. علاوه بر نیاز به آگاهی آدرس میزبانی که یک پیام به سمت آن هدف گیری شده است، میزبان ارسال کننده بایستی به شناسایی پردازش در حال اجرا در مقصد نیز بپردازد. آگاهی از این اطلاعات از آن جهت مهم است که به طور کلی یک میزبان می‌تواند کاربردهای بسیاری را اجرا نماید. یک شماره **پورت**[2] مقصد این هدف را روشن می‌سازد. کاربردهای متداول و محبوب دارای شماره پورت‌های مشخص و ویژه می‌شوند. به‌عنوان مثال یک وب سرور از طریق شماره پورت 80 شناسایی می‌شود. یک سرور پست الکترونیکی که از پروتکل SMTP استفاده می‌کند دارای شماره پورت 25 است. فهرست همه‌ی شماره پورت‌های مشهور برای پروتکل‌های استاندارد اینترنت را می‌توان از طریق سامانه‌ی: http://www.iana.org جستجو نمود. هنگامی‌که یک طراح یک کاربرد جدید را خلق می‌کند به آن کاربرد بایستی یک شماره پورت جدید اختصاص یابد. مطالب مربوط به شماره پورت‌ها در فصل سوم بررسی شده است.

۲-۵ پروتکل‌های لایه کاربرد

آموختیم که پردازش‌های شبکه از طریق ارسال پیام‌ها به سوکت‌ها با یکدیگر ارتباط برقرار می‌کنند. اما ساختار این پیام‌ها چگونه است؟ مفهوم هر یک از میدانهای مختلف تعریف شده در پیام‌ها چیست؟ چه هنگام پردازش‌ها پیام ارسال می‌کنند؟ این پرسشها ما را به حقیقت پروتکل‌های لایه کاربرد راهنمایی می‌کنند. یک پروتکل لایه کاربرد به تعریف چگونگی مبادله‌ی پیام‌ها بین پردازش‌های مختلف در حال اجرا بر روی سیستم‌های انتهایی مختلف می‌پردازد. به‌ویژه آنکه یک پروتکل لایه‌ی کاربرد موضوعاتی از قبیل موارد ذیل را تعریف می‌کند:

- انواع پیام‌هایی که مبادله شده‌اند به‌عنوان مثال پیام‌های تقاضا و پاسخ.
- گرامر یا ترکیب ساختار هر پیام مثل میدان‌هایی که در پیام وجود دارند و یا چگونگی تعریف این میدان‌ها
- مفهوم میدان‌ها به عبارت دیگر معانی اطلاعات درون میدان‌ها
- قوانین مربوط به زمان و یا چگونگی ارسال پیام‌ها و پاسخ آنها

برخی از پروتکل‌های لایه کاربرد در مستندهای RFC[3] مشخص شده‌اند و در دسترس عموم قرار گرفته‌اند. به‌عنوان مثال پروتکل لایه کاربرد وب یا HTTP[4] به‌عنوان یک سند RFC در دسترس می‌باشد. اگر طراح یک مرورگر وب قوانین مربوط به این پروتکل را در RFC مربوطه دنبال نماید، مرورگری که وی طراحی نماید قادر خواهد بود صفحه‌های وب را از هر وب سرور مبتنی بر آن RFC

1- Network Address Translation
2- Port number
3- Request for Comment
4- Hyper text transfer protocol (RFC 2616)

بازیابی نماید. پروتکل‌های لایه کاربرد بی شماری نیز موجود می‌باشند که اختصاصی بوده و در اختیار عموم به‌صورت بین المللی قرار نمی‌گیرند. به‌عنوان مثال بسیاری از پروتکل مربوط به سیستم‌های اشتراک فایل P2P که از پروتکل‌های لایه کاربرد خصوصی است، استفاده کرده‌اند. لازم به یادآوری است که تمایز بین کاربردهای شبکه و پروتکل‌های لایه‌ی کاربرد بسیار با اهمیت است. یک پروتکل لایه کاربرد تنها یک قطعه از یک کاربر شبکه است. برای روشن شدن مطلب به مثالهای عنوان شده در ادامه‌ی این بحث دقت نمائید. سرویس وب یک کاربرد مشتری- سرور است که اجازه می‌دهد کاربران مستندهایی را از وب سرورها براساس تقاضا بدست آورند، در صورتی‌که کاربرد وب شامل بسیاری از عناصر و قطعه‌ها از جمله استاندارد قالب‌بندی اسناد یعنی HTML[1]، مرورگرهای وب مثل کاوشگر اینترنت مایکروسافت (ie) و فایرفاکس، وب سرورها مثل سرورهای مایکروسافت و آپاچی و یک پروتکل لایه کاربرد می‌شوند. پروتکل لایه کاربرد وب HTTP، قالب‌بندی و نظم پیام‌هایی که بایستی بین مرورگر و وب سرور مبادله شوند را تعریف می‌کند. بنابراین HTTP فقط یک قطعه است (اگرچه قطعه‌ی بسیار مهمی است) از کاربرد وب. نمونه‌ی دیگر یک کاربرد پست الکترونیکی اینترنت است که دارای بسیاری از عناصر شامل سرور پست نامه‌ها که درون صندوق‌های پستی کاربران را تعریف کرده است، نامه خوان‌ها که برای کاربران امکان خواندن و یا نگارش نامه‌ها و پیام‌ها را فراهم می‌آورند، یک استاندارد و تعریف ساختار پیام‌های پست الکترونیکی (e-mail) و سرانجام پروتکل‌های لایه کاربرد که به تعریف چگونگی مبادله‌ی پیام‌ها بین سرورها و نیز عبور پیام‌ها بین سرورها و نامه خوانها می‌پردازند. بدیهی است در این بخش آخر چگونگی تفسیر محتویات بخش‌های ویژه‌ای از پیام‌ها مثل سرانداز هر پیام الکترونیکی نیز تعریف شده است. پروتکل لایه کاربرد اصلی پست الکترونیکی SMTP[2] یا پروتکل ساده‌ی انتقال پیام می‌باشد. بنابراین پروتکل SMTP اگرچه قطعه‌ی بسیار مهم و حیاتی است اما تنها یک قطعه از کاربرد پست الکترونیکی به‌شمار می‌آید.

کاربردهای شبکه که در این کتاب پوشش‌یافته‌اند در ادامه‌ی این خلاصه مطلب معرفی می‌شوند. کاربردهای اینترنت در دسترس عموم و نیز خصوصی جدیدی تقریباً روزانه طراحی و خلق می‌شوند. بنابراین تمرکز این کتاب بر روی تعداد کوچکی از کاربردها می‌باشد که هم مهم هستند و هم فراگیر. در این فصل پنج کاربرد مهم که عبارتند از: وب، انتقال فایل، پست الکترونیکی، سرویس دفترچه راهنما و کاربردهای P2P بررسی خواهند شد. نخست به بررسی وب می‌پردازیم نه از آن جهت که کاربردی است فوق‌العاده محبوب بلکه به دلیل آنکه پروتکل لایه‌ی کاربرد آن یعنی HTTP بسیار ساده و شفاف از لحاظ درک می‌باشد. پس از آن کاربرد انتقال فایل به دلیل تفاوت جذاب آن با HTTP و تضادی که با آن دارد بررسی می‌شود. به دنبال آن کاربرد پست الکترونیکی یکی از تأثیرگذارترین کاربردها در اینترنت به دلیل پیچیدگی نسبتاً گسترده‌ی آن در مقایسه با وب تحلیل می‌شود. علت پیچیدگی این کاربرد به واسطه‌ی آن است که در عوض بهره گرفتن از یک پروتکل لایه‌ی کاربرد از چندین پروتکل استفاده می‌کند. پس از کاربرد پست الکترونیکی، سرویس محدوده نام (DNS) که در واقع سرویس دفترچه راهنما را فراهم می‌آورد و بسیاری از کاربران اینترنت در تعامل مستقیم با آن قرار ندارند، بررسی می‌شود. بدیهی است درخواست غیر مستقیم کاربران از DNS از طریق دیگر کاربردها مثل وب، انتقال فایل و پست الکترونیکی صورت می‌پذیرد. این سرویس یعنی DNS با ظرافت بسیار نشان می‌دهد چگونه عملکرد یک قطعه از هسته‌ی شبکه که همان ترجمه‌ی نام شبکه به آدرس شبکه

1- Hyper text mark up language
2- Simple mail transfer protocol (RFC5321)

است در یک لایه‌ی کاربرد اینترنت پیاده‌سازی می‌شود. سرانجام کاربردهای اشتراک فایل P2P که عبارتند از توزیع فایل، توزیع بانک‌های اطلاعاتی و تلفن IP تحلیل می‌شود.

۲-۶ وب و پروتکل انتقال فرامتن

تا اوایل دهه‌ی 1990 میلادی کاربرد اینترنت تنها منحصر به پژوهشگران، استادان دانشگاه‌ها و دانشجویان می‌شد، که از طریق یک رمز عبور و نام شناسه به میزبان‌های راه دور متصل می‌شدند و به انتقال فایل‌ها از میزبان‌های محلی به آنها و برعکس می‌پرداختند. بیشترین کاربرد در آن هنگام شامل ارسال و دریافت اخبار، بازیابی و ارسال پست‌های الکترونیکی می‌شد. اگرچه این کاربردها در آن هنگام و حتی اکنون نیز بسیار سودمند بودند اما اینترنت برای بسیاری از جوامع و مردمان دیگر ناشناخته بود. اما در همان اوایل دهه 1990 میلادی یک کاربرد مهم و جدیدی وارد صحنه شد و آن عبارت بود از:

وب گسترده‌ی جهانی (www). این پدیده نوین نخستین کاربرد اینترنت بود که نظر همگان را به خود معطوف نمود. این پدیده به سرعت دچار تغییر و تکامل گردید و این دگرگونی هنوز هم ادامه دارد. وب جهانی سبب شد مردمان در درون و بیرون از محیط‌های کاری خود بتوانند با یکدیگر تعامل برقرار کنند. این پدیده سبب شد اینترنت از حالت یکی از تعداد بی شمار شبکه‌ی داده تبدیل به تنها شبکه‌ی داده در سراسر گیتی شود. شاید آنچه که سبب جذب کاربران به وب شد، توانمندی آن در عملکرد به‌صورت پاسخگویی براساس تقاضا بود. کاربران آنچه را که مایل هستند دریافت می‌کنند و زمان آنرا نیز خود تعریف می‌کنند. بنابراین این رسانه‌ی اجتماعی همانند رسانه‌های رادیویی و تصویری نبود که کاربران را مجبور به دریافت اطلاعاتی نماید که گردانندگان آن اراده می‌کردند.

علاوه بر آنکه وب اطلاعات را براساس تقاضا در اختیار می‌گذارد، بسیاری از ویژگی‌های جذاب دیگری را برای مردمان به همراه دارد که آنها سخت به آن علاقه‌مند و در جهت حفظ آن کوشا می‌باشند. به طرز بسیار ساده‌ای یک فرد اطلاعاتی را از طریق وب در دسترس عموم جهانیان قرار می‌دهد. هر فردی می‌تواند ناشر باشد و این مهم را با کمترین هزینه انجام دهد. موتورهای جستجو و آبَرپیوندها امکان گردش در درون سامانه‌های وب را که شبیه یک اقیانوس است، فراهم می‌آورند. احساس بشر از طریق شبیه‌سازی‌های گرافیکی به ذوق در می‌آید. فرمها و عناصر کوچک اجرایی جاوا و انبوهی از تجهیزات ما را قادر به تعامل با صفحه‌ها و سامانه‌های متعدد می‌نمایند. هر روز بیشتر از روز قبل وب صورتی از واسط‌های دستیابی به اطلاعات ذخیره شده‌ی صوتی و تصویری در اینترنت را برای ما فراهم می‌آورد. امکانات چند رسانه‌ای که براساس تقاضا به آسانی در اختیار متقاضی قرار می‌گیرد.

پروتکل انتقال فرامتن HTTP که پروتکل لایه‌ی کاربرد وب جهانی یا همان اینترنت می‌باشد، در حقیقت هسته و قلب آن را تشکیل می‌دهد. این پروتکل به‌وسیله‌ی دو سند RFC به شماره‌های 1945 و 2616 تعریف شده است. پروتکل HTTP در دو برنامه پیاده‌سازی شده است: برنامه‌های مشتری و برنامه‌ی سرور. این دو برنامه به‌صورت یک جفت، هنگام اجرا بر روی سیستم‌های انتهایی مختلف از طریق مبادله‌ی پیام‌های HTTP با یکدیگر محاوره می‌کنند. پروتکل HTTP ساختار این پیام‌ها را تعریف می‌کند و علاوه بر آن چگونگی مبادله پیام‌ها بین سرور و مشتری را نیز تعریف می‌کند. قبل از بررسی جزئیات HTTP نخست بعضی از واژگان متداول در وب بررسی می‌شود.

یک صفحه یا **سند وب**[1] شامل اشیاء است. یک **شیئی**[2] عبارت است از یک فایل مانند یک فایل HTML، یک فایل تصویری JPEG، یک برنامه‌ی کوتاه Java و یا یک قطعه‌ی تصویری که از طریق یک **پیوند ارتباطی**[3] به نام URL دسترسی به آن امکان‌پذیر می‌شود. اغلب صفحه‌های وب شامل یک فایل بر پایه‌ی **زبان نشانه‌گذاری اَبَرمتن**[4] است که پیوست آن چندین شیئ ارجاعی نیز می‌باشد. به‌عنوان مثال اگر یک صفحه‌ی وب شامل متن HTML و پنج تصویر JPEG باشد، آنگاه صفحه وب دارای 6 شیء می‌باشد، به عبارتی صفحه‌ی پایه‌ی HTML به همراه ۵ تصویر. اصولاً فایل پایه‌ی HTML دیگر عناصر در صفحه را با آدرس پیوند ارتباطی اشیاء URL مورد اشاره قرار می‌دهد. هر آدرس URL دارای دو عنصر است که عبارتند از: نام سرور میزبان که عناصر را در خود جای داده است و نام مسیر اشیاء. به‌عنوان مثال آدرس URL زیر:

http://www.nke.com/computerdept/book.gif

دارای نام میزبان www.nke.com و نام مسیر computerdept/book.gif می‌باشد. از آنجائیکه مرورگرهای وب نظیر کاوشگر اینترنت و فایرفاکس، HTTP سمت مشتری را در چارچوب وب پیاده‌سازی می‌کنند بنابراین معمولاً از واژه‌های مرورگر و مشتری به گونه‌ی جایگزین یکدیگر استفاده می‌شود. سرورهای وب که معمولاً HTTP سمت سرور را پیاده‌سازی می‌کنند، اشیاء وب را که هر کدام از طریق یک URL قابل بازیابی است در درون خود جای داده‌اند. محبوب‌ترین وب سرورها **سرور اطلاعات اینترنت مایکروسافت** (MIIS) و سرور آپاچی (Apache) می‌باشند. آنچه که HTTP تعریف می‌کند عبارت است از آنکه چگونه مشتریان وب صفحه‌های وب را از وب سرورها تقاضا کنند و یا آنکه چگونه سرورها این صفحه‌ها را به مشتریان تحویل دهند. چگونگی تعامل بین مشتری و سرور با جزئیات بیشتر در ادامه‌ی مطالب بررسی خواهد شد، اما مفهوم کلی آن در شکل ۲- ۶ نشان داده شده است. هنگامی‌که یک کاربرد تقاضای یک صفحه‌ی وب را از طریق فشردن ابرپیوند آن می‌کند، مرورگر پیام‌های HTTP تقاضا را برای اشیاء در صفحه به سرور گسیل می‌دارد. سرور این تقاضاها را دریافت و با پیام‌های HTTP جواب که شامل اشیاء است به آنها پاسخ می‌دهد. پروتکل HTTP از پروتکل TCP به‌عنوان پروتکل انتقال به جای UDP استفاده می‌کند. مشتری HTTP ابتدا درخواست ارتباط با سرور مورد نظر را از طریق TCP می‌کند و هنگامی‌که این ارتباط برقرار گردید، پردازش‌های مرورگر و سرور از طریق واسط‌های سوکت خود ارتباط TCP را دستیابی می‌کنند. قبلاً در بخش ۲-۱ توضیح داده شد که در سمت مشتری واسط سوکت نقش دروازه‌ی ورودی و خروجی بین پردازش مشتری و ارتباط TCP را ایفا می‌کند در حالیکه در سمت سرور آن، دَری بین پردازش سرور و ارتباط TCP به‌شمار می‌آید. مشتری پیام‌های درخواست HTTP را به واسط سوکت خود می‌فرستد و پیام‌های پاسخ HTTP را نیز از واسط سوکت خود دریافت می‌کند. به گونه‌ی مشابه سرور HTTP پیام‌های تقاضا را از طریق واسط سوکت خود دریافت می‌کند و پیام‌های پاسخ را به واسط سوکت خود گُسیل می‌دارد.

1- Web page
2- Object
3- Uniform resource locator
4- Hyper text markup language



شکل ۲- ۶ ♦ رفتار تقاضا و پاسخ پروتکل HTTP

هنگامی‌که مشتری پیامی را به واسط سوکت خود ارسال می‌کند، آن پیام از کنترل مشتری خارج و در اختیار TCP قرار می‌گیرد. با یادآوری این نکته از بخش ۲-۱ که TCP یک سرویس مطمئن انتقال داده را به پروتکل HTTP ارایه می‌دهد، این به مفهوم آن است که هر پیام تقاضای HTTP ارسالی از سوی پردازش مشتری، سرانجام بدون هیچگونه تغییری به سرور می‌رسد. به شکل مشابه هر پیام ارسالی پاسخ HTTP از سوی سرور نیز در نهایت بدون تغییر به مشتری تحویل می‌گردد. در اینجا می‌توان به اهمیت معماری لایه‌ای و ویژگی‌های آن پی برد. پروتکل HTTP نیازی ندارد که نگران اتلاف داده‌ها یا چگونگی جزئیات بازیابی داده‌های تلف شده یا به نظم درآوردن داده‌ها در بین شبکه به‌وسیله‌ی TCP باشد. این دغدغه‌ها از وظایف TCP و پروتکل‌های لایه‌های زیرین مجموعه‌ی پروتکلی می‌باشد. نکته‌ی مهم آن است که سرور، فایل تقاضا شده را بدون ذخیره‌سازی وضعیت‌های اطلاعاتی مشتری به آن ارسال می‌نماید. اگر یک مشتری ویژه دو بار تقاضای عنصر مشابهی را در فواصل زمانی چند ثانیه بنماید، سرور به این تقاضا از طریق ارسال مجدد عنصر مورد نظر مشتری پاسخ می‌دهد. به عبارت دیگر مثل اینکه فراموش کرده باشد که چند لحظه‌ی پیش به تقاضا، یک بار پاسخ داده است. علت این موضوع آن است که سرور HTTP هیچگونه اطلاعاتی را از مشتری جمع آوری نکرده است. به همین منظور به HTTP یک پروتکل **بدون موقعیت**[1] گفته می‌شود. تأکید بر این نکته نیز ضروری است که وب از معماری کاربرد سرور- مشتری به گونه‌ای که در بخش ۲- ۱ توضیح داده شد استفاده می‌کند. یک وب سرور همیشه فعال (روشن) است و دارای یک آدرس ثابت است و خدمات آن از طریق میلیون‌ها مرورگر مختلف قابل دستیابی است.

در بسیاری از کاربردهای اینترنت محاوره‌ی بین مشتری و سرور برای مدت زمانی نسبتاً طولانی صورت می‌پذیرد. این محاوره به‌صورت صدور مجموعه‌ای از تقاضاها از سوی مشتری و سپس پاسخ آنها از جانب سرور به‌صورت یک به یک صورت می‌پذیرد. بستگی به کاربرد و چگونگی پیاده‌سازی یک محاوره روند آن به‌صورت متناوب و یا پیوسته در یک فاصله‌ی زمانی انجام می‌شود. هنگامی‌که این محاوره مشتری- سرور از طریق TCP در حال انجام است طراح یک کاربرد مجبور به انجام یک تصمیم‌گیری مهم می‌شود. آن تصمیم عبارتست از آنکه آیا هر جفت تقاضا و پاسخ بایستی از طریق یک ارتباط مجزای TCP صورت پذیرد و یا آنکه همگی تقاضاها و به دنبال آن پاسخ‌ها بایستی از طریق یک

1- Stateless Protocol

ارتباط مشابه TCP انجام گیرد؟ در صورت رویکرد نخست کاربرد را اصطلاحاً **ارتباط ناپایا**[1] می‌نامند در صورتی‌که در حالت دوم آن کاربرد یک ارتباط **پایا** یا **ماندگار** نامیده می‌شود. برای درک عمیق موضوع از لحاظ طراحی، نقاط قوت و ضعف ارتباط پایا در چارچوب یک مورد ویژه بررسی می‌شود و آن کاربرد پروتکل HTTP است که می‌تواند از هر دو ارتباط ماندگار و ناپایا استفاده نماید. اگرچه روش پیش‌فرض پروتکل HTTP استفاده از ارتباط ماندگار است اما سرورها و مشتری‌های HTTP را می‌توان به‌صورت ارتباط‌های غیرپایا نیز تعریف نمود.

- **ارتباط‌های غیر ماندگار با HTTP**

برای آشکار شدن موضوع مراحل انتقال یک صفحه‌ی وب از سرور به یک مشتری برای حالت ارتباط ناپایا بررسی می‌شود. اگر فرض شود صفحه‌ی مورد نظر شامل یک فایل پایه‌ی HTML به همراه 10 تصویر JPEG می‌باشد و همگی 11 عنصر در یک سرور مشابه جاسازی شده‌اند و علاوه بر آن آدرسی که فایل پایه‌ی HTML از آن استفاده می‌کند به‌صورت ذیل باشد:

http://www.someschool.edu/somedepartment/home.index

آنگاه وقایعی که روی می‌دهد عبارتند از:

۱- پردازش HTTP مشتری پیش قدم برای یک ارتباط TCP با سرور می‌شود این ارتباط از طریق بخش www.someschool.edu از آدرس بر روی شماره پورت 80 انجام می‌شود که شماره‌ی پیش‌فرض پروتکل HTTP است و برای ارتباط TCP یک سوکت در مشتری و دیگری در سرور تأمین شده است.

۲- پروتکل HTTP مشتری یک پیام تقاضای HTTP را از طریق سوکت مربوطه به سرور ارسال می‌کند. پیام تقاضا شامل نام مسیر است که در این مثال: /somedept/home.indes می‌باشد.

۳- پردازش سرور HTTP پیام تقاضا را از سوکت خود دریافت و شیئ آن را که همان: /somedept/home.indes می‌باشد از طریق حافظه‌ی RAM و یا مشابه آن بازیابی نموده و آن را در پیام پاسخ HTTP کپسول‌بندی و سپس از طریق سوکت خود به سوی مشتری گسیل می‌دارد.

۴- پردازش سرور HTTP سپس از TCP می‌خواهد که ارتباط TCP را قطع نماید، اما TCP تا زمان حصول اطمینان از اینکه مشتری پیام پاسخ را بدون هیچگونه تغییری سالم دریافت نموده است به این ارتباط پایان نمی‌دهد.

۵- مشتری HTTP پیام پاسخ را دریافت می‌کند. ارتباط TCP پایان می‌پذیرد. پیام عنصر کپسول‌بندی شده را به‌عنوان یک فایل HTTP تعریف می‌کند و مشتری فایل را از پیام جدا نموده و سپس به مطالعه‌ی فایل HTML می‌پردازد. نتیجه‌ی آن یافتن ارجاع‌ها به 10 عنصر JPEG می‌باشد.

۶- مجدداً چهار مرحله‌ی نخست برای هر ارجاع به عنصر JPEG تکرار می‌شود. به عبارت دیگر در این مثال حلقه‌ای با 10 بار تکرار.

هنگامی‌که مرورگر صفحه‌ی وب را دریافت نمود، آنرا برای کاربر به نمایش در می‌آورد. دو مرورگر

1- Non persistent connections

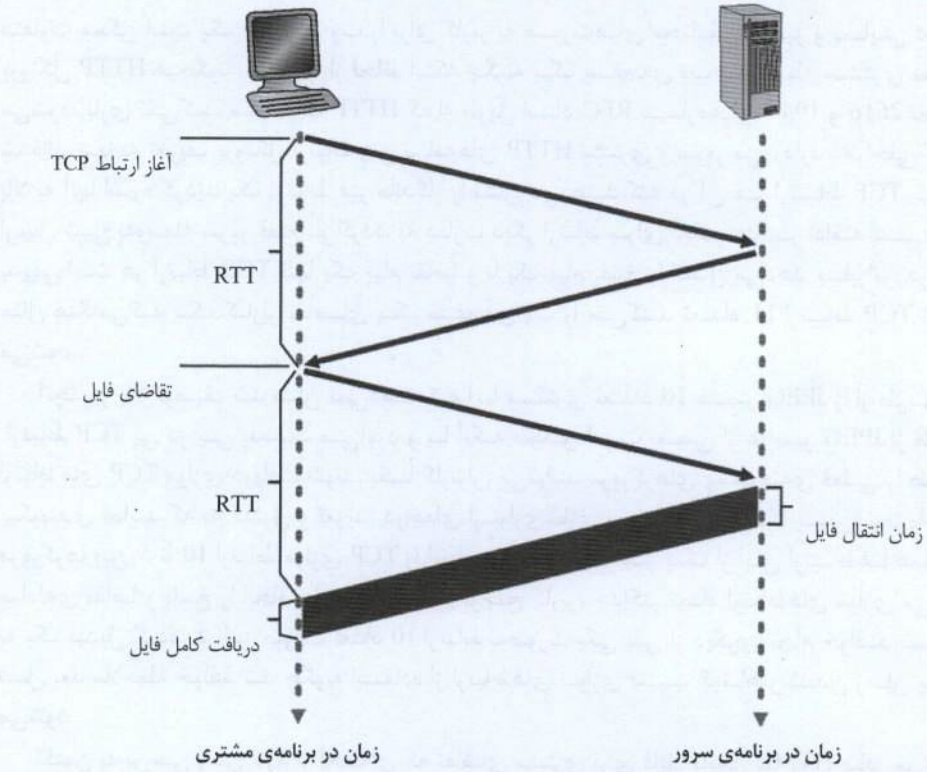
متفاوت ممکن است یک صفحه‌ی وب را برای کاربر به صورت‌های مختلف تفسیر و نمایش دهند. پروتکل HTTP هیچگونه نقشی را از لحاظ اینکه چگونه یک صفحه‌ی وب به‌وسیله مشتری تفسیر می‌شود، بازی نمی‌کند. مشخصات HTTP که از طریق اسناد RFC شماره‌های 1945 و 2616 تعریف شده‌اند تنها به تعریف پروتکل ارتباط بین برنامه‌های HTTP مشتری و سرور می‌پردازند. مراحلی که در بالا به آنها اشاره گردید یک ارتباط غیر ماندگار را نشان می‌دهند که در آن هر ارتباط TCP پس از ارسال شیئ به‌وسیله سرور قطع می‌گردد. به عبارت دیگر ارتباط برای دیگر عناصر ادامه نمی یابد. بدیهی است هر ارتباط TCP تنها یک پیام تقاضا و یا یک پیام پاسخ را انتقال می‌دهد. بنابراین در این مثال هنگامی‌که یک کاربر تقاضای یک صفحه‌ی وب را می‌کند تعداد 11 ارتباط TCP تولید می‌شود.

آنچه در بالا توصیف شد نشان نمی‌دهد که آیا مشتری تعداد 10 عنصر JPEG را از طریق 10 ارتباط TCP پی در پی بدست می‌آورد و یا آنکه ممکن است بعضی از عناصر JPEG از طریق ارتباط‌های TCP موازی دریافت شوند. یقیناً کاربران می‌توانند مرورگرهای پیشرفته‌ی فعلی را طوری پیکربندی نمایند که به کنترل و تعریف درجه‌ای از توازی‌سازی بپردازد. در حالت پیش‌فرض اغلب مرورگرها بین 5 تا 10 ارتباط موازی TCP را انجام می‌دهند که هر یک از این ارتباط‌ها عملیات مبادله‌ی تقاضا و پاسخ را انجام می‌دهد. براساس ترجیح کاربر، حداکثر تعداد ارتباط‌های موازی می‌تواند به یک تبدیل گردد. در این صورت تعداد 10 ارتباط به‌صورت یکی پس از دیگری انجام خواهد شد. در فصل بعد ملاحظه خواهد شد چگونه استفاده از ارتباط‌های توازی سبب کوتاهتر شدن زمان پاسخ می‌شود.

اکنون به بررسی زمان لازم از لحظه‌ای که تقاضای مشتری برای فایل پایه‌ی HTML صادر می‌شود تا هنگامی‌که تمامی آن فایل در اختیار آن قرار می‌گیرد می‌پردازیم. برای این منظور زمان دوره‌ی گردش (RTT) تعریف می‌شود که عبارت است از زمانی که یک بسته‌ی کوچک از مشتری به سمت سرور و سپس از طریق آن به مشتری باز می‌گردد. زمان RTT شامل: تأخیر انتشار بسته، تأخیر صف بسته در بین مسیریاب‌های واسط و راهیابها و سرانجام تأخیر پردازش بسته می‌شود. اکنون حالتی را در نظر بگیرید که یک کاربر بر روی یک ابرپیوند کلیک می‌کند. شکل ۲- ۷ نشان می‌دهد که این عمل سبب می‌شود مرورگر یک ارتباط TCP را بین خود و وب سرور تقاضا نماید. این تقاضا شامل یک دست‌دهی سه جانبه به‌صورت زیر می‌باشد:

- مشتری یک قطعه‌ی کوتاه TCP را به سرور ارسال می‌کند.
- سرور دریافت این قطعه را با ارسال یک قطعه کوتاه TCP تأئید می‌کند.
- مشتری دریافت پاسخ سرور را تأئید می‌نماید.

دو بخش نخست این دست‌دهی سه جانبه یک RTT به طول می‌انجامد. پس از کامل شدن این دو بخش مشتری پیام تقاضای HTTP را که با بخش سوم دست‌دهی سه‌جانبه ترکیب کرده است به ارتباط TCP ارسال می‌کند. هنگامی‌که پیام تقاضا به سرور وارد می‌شود، آنگاه فایل HTML از سوی سرور به ارتباط TCP گسیل می‌شود. بدیهی است این تقاضا و پاسخ زمان RTT دیگری را مصرف می‌کند.



شکل ۲- ۷ ♦ محاسبه‌ی تخمینی زمان لازم برای تقاضا و پاسخ یک فایل

نتیجه آنکه تقریباً مجموع زمان لازم برابر است با حاصل جمع 2RTT و زمان انتقال در سرور فایل HTML.

• ارتباط‌های پایا با HTTP

ارتباط‌های ناپایا از نقطه ضعف‌هایی برخوردار است که می‌تواند در کیفیت آن تأثیرگذار باشد. این نارسایی‌ها عبارتند از: نخست، یک ارتباط جدید بایستی برای هر درخواست عنصر ایجاد و نگهداری شود. بنابراین برای هر یک از این ارتباط‌ها، بافرهای TCP بایستی مشخص گردند و متغیرهای TCP نیز بایستی در هر دو طرف مشتری و سرور نگهداری شوند. رعایت این موضوع سبب بار اضافی بر روی وب سرور می‌شود، زیرا این احتمال وجود دارد که سرور بایستی به تقاضاهای صادر شده از سوی صدها مشتری مختلف در هر لحظه پاسخ دهد. دوم همانگونه که قبلاً توضیح داده شد، هر عنصر دچار یک تأخیر تحویل شامل 2RTT می‌شود که یکی برای ایجاد و نگهداری ارتباط TCP است و دیگری برای تقاضا و دریافت عنصر می‌باشد.

در ارتباط‌های پایا یک سرور پس از ارسال پاسخ، ارتباط TCP را باز نگاه می‌دارد، بنابراین تقاضاها و پاسخ‌های مربوط به آنها در بین همان مشتری و سرور می‌تواند از طریق ارتباط TCP ایجاد شده در قبل صورت پذیرد. یک صفحه‌ی وب همانند مثال قبل که شامل فایل پایه‌ی HTML به همراه 10 عنصر می‌شد، اکنون از طریق یک ارتباط TCP از نوع ماندگار قابل انتقال است. علاوه بر این موارد، چندین صفحه‌ی وب که بر روی یک سرور ساکن می‌باشند از طریق آن قابل انتقال به مشتری مشابه از

طریق یک ارتباط پایای TCP می‌باشند. تقاضا برای اشیاء می‌تواند به‌صورت پی در پی صورت پذیرد بدون آنکه نیازی برای پاسخگویی به تقاضاهای در جریان لازم باشد. به عبارت دیگر نوعی روش **خط لوله**[1] پدیدار می‌شود. سرور HTTP معمولاً ارتباطی را که برای مدت زمان معینی بدون استفاده مانده است قطع می‌کند. زمان انتظار برای قطع، انعطاف پذیر و قابل برنامه‌ریزی است. هنگامی که یک سرور تقاضاهای پی در پی دریافت می‌کند، عناصر موضوع تقاضا را نیز یکی پس از دیگری ارسال می‌کند. روش پیش‌فرض پروتکل HTTP استفاده از ارتباط پایا با روش خط لوله است.

۲-۷ قالب‌بندی پیام HTTP

سند RFC شماره‌ی 2616 قالب‌بندی پیام‌های HTTP را نیز تعریف می‌کند. دو نوع پیام HTTP، یکی پیام‌های تقاضا و دیگری پاسخ به ترتیب زیر بررسی می‌شوند:

پیام تقاضای HTTP

در ذیل یک پیام درخواست HTTP به‌صورت نمونه آورده شده است:

```
GET /Somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: Close
User-agent: Mozilla/5.0
Accept-language: fr
```

با مطالعه‌ی این مثال ساده آگاهی قابل ملاحظه‌ای را می‌توان بدست آورد. نخستین نکته‌ی قابل توجه آن است که این پیام براساس کُدهای ساده در قالب یک **متن آسکی**[2] نوشته شده است. به عبارت دیگر زبان محاوره‌ی معمولی بین افراد، دوم آنکه ملاحظه می‌شود پیام شامل پنج خط می‌باشد. هر یک از سطرها از طریق فشرده شدن کلید « بازگشت » به سطر بعد رفته است. مفهوم این عبارت آن است که به دنبال هر خط از متن فرمان رفتن به خط بعد قرار دارد و سرانجام آنکه برای آخرین خط پیام کلید بازگشت دو مرتبه فشرده شده است. اگرچه این پیام درخواست ویژه دارای 5 سطر است، اما یک پیام می‌تواند شامل تعداد زیادی خط نیز باشد. حتی می‌تواند شامل یک سطر باشد ! اکنون به تحلیل این مثال ساده از لحاظ مفهوم آن می‌پردازیم. خط نخست یک پیام تقاضای HTTP است که اصطلاحاً سطر تقاضا نام دارد. سطرهایی که به دنبال این خط می‌آیند در ادبیات شبکه **خطوط سرانداز**[3] نام نهاده‌اند. خط تقاضا حامل سه محدوده‌یا میدان است:

(۱) محدوده‌ی روش، (۲) میدان آدرس URL و (۳) میدان نوع و یا نسخه‌ی HTTP.

محدوده‌ی روش ارزش‌های مختلفی را اختیار می‌کند این روش‌ها عبارتند از: GET، POST، HEAD، PUT و DELETE. اکثر پیام‌های تقاضای HTTP از روش GET استفاده می‌کنند. از این روش هنگامی استفاده می‌شود که مرورگر تقاضای یک شیئ را می‌کند که شناسه‌ی آن در میدان آدرس URL تعریف شده است. در این مثال مرورگر عنصر: somedir/page.html/ را تقاضا کرده است، مفهوم محدوده‌ی نسخه نیز آشکار است در این مثال مرورگر از نسخه‌ی HTTP/1.1 استفاده کرده است.

1- Pipelining
2- ASCII
3- Header Lines

اکنون سطرهای مربوط به سرپیام بررسی می‌شوند. عنوان:

Host:www.someSchool.edu

بیان کننده‌ی میزبانی است که شیئ در آن ساکن است. ممکن است تصور شود که این سطر لازم نیست، چرا که یک ارتباط TCP در درون میزبان لحاظ شده است. در بخش ۲- ۹ ملاحظه خواهد شد که اطلاعات ارایه شده از سوی سرپیام میزبان به‌وسیله **سرورهای جانشین**[1] لازم می‌باشند. با افزودن سطر Connection:Close مرورگر به سرور می‌گوید مایل نیست از ارتباط پایا استفاده نماید، به عبارت دیگر از سرور می‌خواهد که پس از ارسال عنصر ارتباط را قطع نماید. سطر بعدی یعنی User agent:Mozilla/4.0 مشخص می‌کند که از چه عامل کاربری استفاده می‌کند به عبارت دیگر این سطر نوع مرورگر استفاده شده برای صدور تقاضا به سرور را تعریف می‌کند. در این مثال مرورگر استفاده شده مُوزیلا نسخه 4 (Mozilla/4.0) می‌باشد. این سرپیام از آن جهت مفید است که سرور را قادر به ارسال نسخه‌های مختلف از یک عنصر مشابه به دیگر عامل‌های کاربری و یا مرورگرها می‌نماید. هر یک از نسخه‌ها به‌وسیله‌یک URL مشابه دستیابی می‌شوند. سرانجام سطر آخر دارای سرپیام Accept language:fr است، که ترجیح کاربر در دریافت عنصر به زبان یا نسخه‌ی فرانسوی آن می‌باشد. در صورتی‌که این نسخه از عنصر در سرور موجود نباشد آنگاه سرور نسخه پیش‌فرض خود را ارسال خواهد نمود. سطر آخر یکی از ده‌ها عنوان قابل دسترسی در HTTP می‌باشد. شکل ۲- ۸ قالب‌بندی کلی یک پیام تقاضا را نشان می‌دهد. ملاحظه می‌شود که پس از خطوط عناوین و کلیدهای بازگشت اضافه شده به آنها یک چارچوب بدنه نیز وجود دارد. در روش GET این چارچوب بدنه خالی است اما هنگامی‌که از POST استفاده می‌شود این چارچوب نیز تعریف می‌گردد. یک مشتری HTTP معمولاً از روش POST برای تکمیل یک فرم استفاده می‌کند. به‌عنوان مثال، هنگامی‌که کاربر واژه‌ای را برای جستجو وارد یک موتور جستجو می‌کند. با یک پیام POST هنوز هم کاربر تقاضای یک صفحه‌ی وب را از سرور می‌کند اما محتویات ویژه‌ی درون آن صفحه‌ی وب بستگی دارد به آنچه کاربر در محدوده‌های فرم وارد نموده است. اگر ارزش میدان POST باشد آنگاه چارچوب بدنه حامل اطلاعاتی می‌شود که کاربر در میدان‌های فرم وارد کرده است. بدیهی است تقاضایی که در ارتباط با یک فرم شکل می‌گیرد ممکن است لزوماً از POST استفاده ننماید. در اغلب مواقع فرم‌های HTML از روش GET استفاده می‌کنند و داده‌های وارد شده در محدوده‌های فرم در آدرس URL تقاضا شده قرار می‌گیرند. به‌عنوان مثال اگر یک فرم از GET با دو محدوده استفاده کند که یک میدان دارای ارزش monkeys و میدان دوم حامل bananas باشد آنگاه URL دارای ساختار زیر خواهد بود:

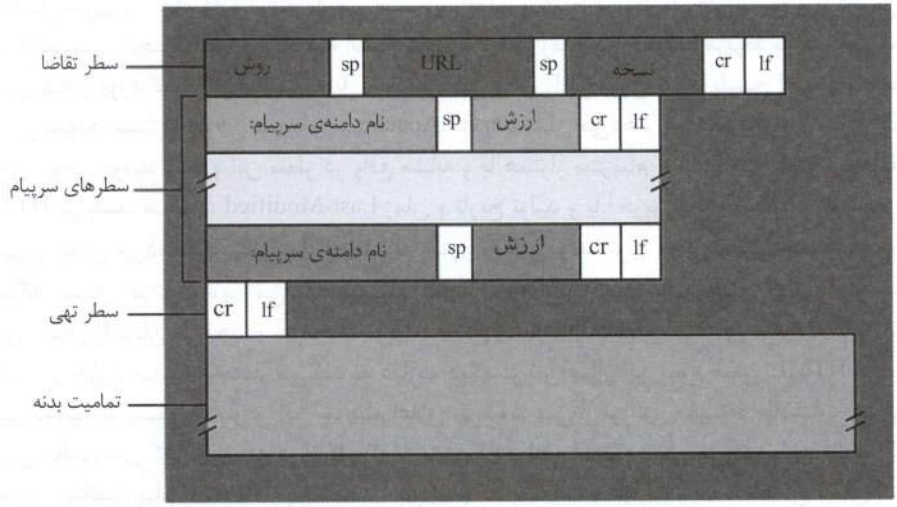
```
www.somesite.com/animalsearch?monkeys & bananas
```

شما ممکن است در وب گردی روزانه‌ی خود به آدرس‌های URL طولانی چون مثال بالا برخورد کرده باشید.

روش HEAD از لحاظ بیان مشابه GET می‌باشد. هنگامی‌که یک سرور درخواستی را با روش HEAD دریافت می‌کند پاسخ آنرا با یک پیام HTTP بدون شیئ می‌دهد. طراحان برنامه‌های کاربردی معمولاً از روش HEAD برای شناسایی **عیب‌های**[2] نرم‌افزاری استفاده می‌کنند. روش PUT نیز غالباً در هماهنگی با ابزار نشر وب به‌کار گرفته می‌شود. این روش به کاربر امکان می‌دهد که یک عنصر را در

1- Proxy Servers
2- Debugging

یک مسیر ویژه (مانند دفترچه راهنما) و یا یک وب سرور خاص بارگذاری نماید. از روش PUT کاربردهایی که می‌خواهند اشیاء را در یک وب سرور بارگذاری نمایند نیز استفاده می‌کنند. سرانجام روش DELETE که به یک کاربر یا کاربرد اجازه می‌دهد به حذف یک عنصر از یک وب سرور بپردازد.



شکل ۲- ۸ ♦ قالب‌بندی کلی یک پیام تقاضای

پیام پاسخ

در ذیل نمونه‌ای از یک پیام پاسخ HTTP آمده است. این پیام می‌تواند در حقیقت پاسخ مثال بالا باشد که به‌عنوان پیام تقاضا ارسال گردیده بود:

```
HTTP/1.1 200 OK
Connection: Close
Date: Sat, 07 Jul 2012 12:00:15 GMT
Server: Apache/1.3.0 (Centos)
Last-Modified: Sun, 6 May 2012 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

با کمی دقت به متن پیام پاسخ که از سوی سرور ارسال گردیده است. ملاحظه می‌شود که این پیام دارای سه بخش است: (۱) یک **خط وضعیت**[1]، (۲) تعداد 6 خط سرپیام و (۳) چارچوب بدنه. در حقیقت بخش اصلی این پیام چارچوب آن می‌باشد که شامل عنصر درخواست شده است که با خط: .. data data data data data. به نمایش درآمده است. خط وضعیت خود از سه محدوده تشکیل شده است: نسخه‌ی پروتکل، یک کُد وضعیت و اعلام وضعیت پیام مربوطه. در این مثال خط وضعیت بیان کننده‌ی اطلاعاتی از قبیل آنکه سرور از پروتکل HTTP/1.1 استفاده می‌کند و همه چیز در وضعیت

1- Status Line

مطلوب (OK) قرار دارد. مفهوم آن است که سرور اطلاعات مورد نیاز درخواست را یافته است و در حال ارسال عنصر است. اکنون اگر به خطوط سرپیام توجه شود ملاحظه می‌شود سرور از خط: Connection:Close استفاده نموده تا به مشتری اعلام نماید که پس از ارسال پیام ارتباط TCP را قطع می‌کند. سرپیام ...Date:Sat بیان کننده زمان و تاریخی است که پروتکل HTTP پیام پاسخ و ارسال آن از طریق سرور را انجام داده است. بدیهی است این زمان و تاریخ مربوط به خلق شیئ و یا آخرین مرتبه که عنصر مورد تجدید نظر قرار گرفته است، نمی‌باشد. این زمان در حقیقت مربوط به هنگامی است که سرور عنصر مورد نظر را از سیستم فایل خود بازیابی و آن را در درون پیام پاسخ لحاظ و همراه آن ارسال نموده است. از طرفی سرپیام: ....Server:Apache نشان می‌دهد که پیام پاسخ از سوی یک وب سرور آپاچی تولید شده و این سطر در واقع مشابه و یا همتراز سرپیام: User-agent: در پیام تقاضای HTTP می‌باشد. سرپیام: Last-Modified زمان و تاریخ تولید و یا آخرین تجدید نظر انجام شده بر روی شیئ را نشان می‌دهد. این عنوان از لحاظ نهان‌سازی عنصر در مشتری محلی و نیز **سرورهای نهان شبکه**[1] بسیار حیاتی و مهم می‌باشد. سرپیام: Content-Length تعداد بایت‌های موجود در شیئ در حال ارسال را نشان می‌دهد. و سرانجام سرپیام: Content-Type: بیان کننده‌ی نوع اطلاعاتی است که عنصر در چارچوب بدنه اختیار می‌کند به عبارت دیگر در این مثال این نوع متن HTML است. نوع شیئ به‌صورت رسمی از طریق این سرپیام اعلان می‌شود. پس از بررسی مفهوم اطلاعات پیام پاسخ اکنون قالب‌بندی کلی پیام که در شکل ۲- ۹ نشان داده شده است بررسی می‌شود. این قالب کلی در حقیقت نگاشت پیام پاسخ بالا می‌باشد. در این پیام کُد وضعیت و عبارت وابسته به آن بیان کننده‌ی نتیجه‌ی یک تقاضا می‌باشد. بعضی از کُدهای متداول و عبارت‌های وابسته به آنها، در ذیل آمده است:

- کُد :200 OK به مفهوم آن است که تقاضا موفق است و اطلاعات در قالب پیام پاسخ در حال ارسال می‌باشد.
- کُد :301 Moved Permanently یعنی عنصر مورد نظر به‌صورت دایمی جابجا گردیده است (نقل مکان) و آدرس URL جدید آن در سرپیام Location: در پیام پاسخ مشخص شده است. بدیهی است نرم‌افزار مشتری به‌صورت خودکار URL جدید را بازیابی می‌کند.
- کُد :400 Bad Request به مفهوم یک رمز خطای عام (ژنریک) می‌باشد و نشان می‌دهد که پیام تقاضا از سوی سرور قابل تفسیر نمی‌باشد.
- کُد :404 Not Found این پیام نشان می‌دهد که مستند مورد نظر در سرور وجود ندارد.
- کُد :505 HTTP Version Not Supported به این معنی است که نسخه‌ی پروتکل مورد تقاضا به‌وسیله‌ی سرور پشتیبانی نمی‌گردد.

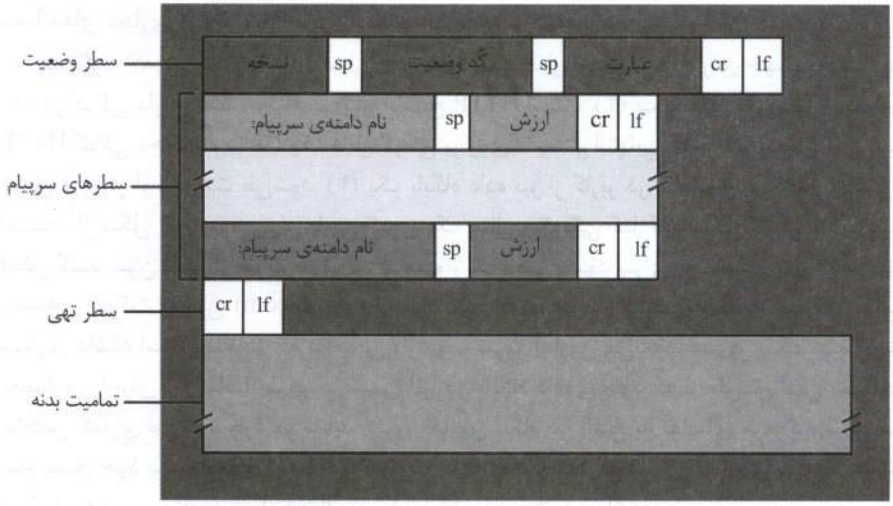
چنانچه مایل باشید یک پیام پاسخ HTTP را در قالب واقعی آن ملاحظه کنید، ابتدا به وب سرور مورد علاقه‌ی خود از طریق Telnet متصل شوید، آنگاه یک سطر پیام تقاضا برای عنصری که در آن سرور قرار دارد، وارد نمائید. به‌عنوان مثال در مقابل فرمان بی‌درنگ (C:\) عبارت‌های ذیل را وارد نمائید:

```
telnet cis.poly.edu 80

Get/~ross/ HTTP/1.1
Host: cis.poly.edu
```

1- Cache Servers (proxy servers)

پس از نگارش آخرین سطر، کلید بازگشت را دو بار فشار دهید. این پیام سبب می‌شود که یک ارتباط TCP در پورت 80 متعلق به میزبان Cis.Poly.edu گشوده شده و سپس پیام تقاضای HTTP ارسال گردد. پاسخ این درخواست فایل پایه‌ی HTML مربوط به صفحه‌ی شخصی دکتر راس می‌باشد. اگر ترجیح می‌دهید سطرهای مربوط به پیام HTTP را بدون شیئ دریافت کنید، کافی است واژه‌ی GET را با HEAD جایگزین نمائید. سرانجام عبارت: /ross~/ را با /banana~/ جایگزین نمائید و مشاهده نمائید که چه پاسخی را دریافت می‌کنید.



شکل ۲- ۹ ♦ قالب کلی یک پیام پاسخ

در این بخش سرپیام‌های قابل استفاده در بین پیام‌های تقاضا و پاسخ HTTP بررسی گردید. مستند پروتکل HTTP به تعریف تعداد بی شماری از اینگونه سرپیام‌ها می‌پردازد که می‌توان از طریق مرورگرها، وب سرورها و سرورهای نهان شبکه وارد نمود. در پایان این بخش دو پرسش دیگر نیز بررسی می‌گردد. نخست آنکه یک مرورگر چگونه در مورد اینکه کدام سطر عناوین بایستی در یک **پیام تقاضا** وارد شود، تصمیم‌گیری می‌کند؟ و دوم آنکه یک وب سرور چگونه در مورد اینکه کدام سرپیام بایستی در یک **پیام پاسخ** لحاظ گردد، تصمیم‌گیری می‌کند؟ یک مرورگر سرپیام‌ها را به‌صورت تابعی از نسخه و نوع خود تولید می‌کند به عبارت دیگر یک مرورگر HTTP/1.0 قادر به تولید سرپیام‌های از نوع 1.1 نمی‌باشد. نکته‌ی دیگر آنکه براساس نوع پیکربندی کاربر مرورگر زبان ترجیحی و یا آنکه آیا مرورگر در حالت فعلی خود یک عنصر از رده خارج از لحاظ زمان و یا نوع را در نهان دارد، تصمیم‌گیری انجام می‌دهد. وب سرورها نیز به گونه‌ی مشابه عمل می‌کنند، زیرا انواع سرورهای متفاوت از لحاظ نسخه، پیکربندی و طراحی وجود دارند که همگی آنها بر روی گزینش و لحاظ نمودن سرپیام‌ها در پاسخ تأثیرگذار می‌باشند.

۲- ۸ تعامل کاربر و سرور: cookies

در مطالب بالا اشاره گردید که یک سرور HTTP بدون وضعیت یا موقعیت می‌باشد. این موضوع سبب

می‌شود سرورها ساده تر شده و در نتیجه امکان طراحی وب سرورهایی فراهم می‌آید که می‌توانند به‌صورت همزمان هزاران ارتباط TCP را با کارایی بسیار بالا تشکیل دهند. همواره برای یک سامانه‌ی وب شناسایی کاربران ممکن است به دلایل مختلف ضروری به نظر آید. این دلایل می‌تواند شامل مواردی از قبیل تمایل سرور به محدود نمودن دستیابی کاربران باشد و یا آنکه می‌خواهد محتویات اطلاعات خود را تابعی از شناسه‌ی یک کاربر نماید. برای همین منظور پروتکل HTTP از **کوکی‌ها**[1] استفاده می‌کند. این عناصر به‌وسیله مستند RFC شماره‌ی 2956 تعریف شده‌اند. کوکی‌ها برای سامانه‌های وب این امکان را فراهم می‌آورند که آنها بتوانند کاربران را رهگیری کنند. اکثریت سامانه‌های تجاری وب امروزه از این امکانات استفاده می‌کنند.

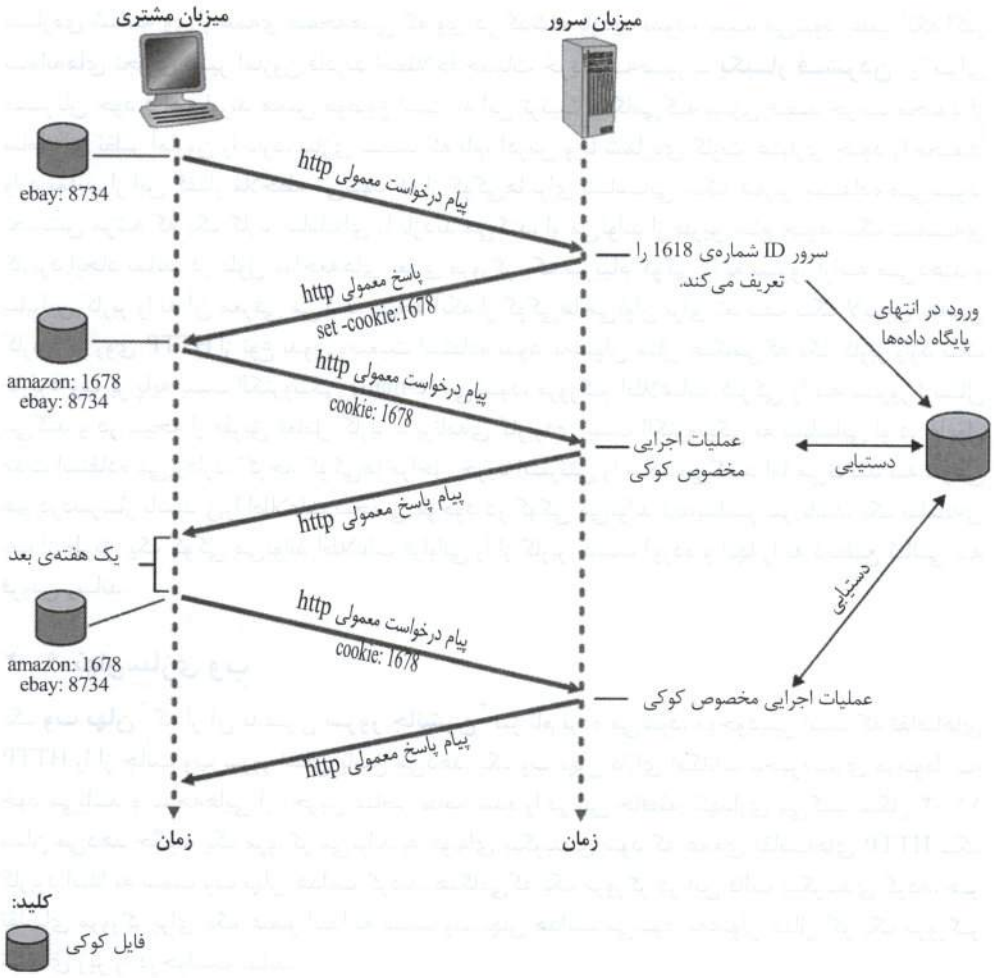
همانگونه که در شکل ۲- ۱۰ نشان داده شده است فناوری کوکی‌ها دارای چهار بخش می‌باشد. (۱) هر کوکی دارای یک سرپیام در پیام پاسخ HTTP است. (۲) یک سرپیام نیز در پیام تقاضای HTTP کوکی وجود دارد. (۳) یک فایل کوکی بر روی سیستم انتهایی کاربر قرار داده می‌شود که از طریق مرورگر او مدیریت می‌شود. (۴) یک پایگاه داده نیز از کاربر در سامانه وب نگهداری می‌شود. با استفاده از شکل ۲- ۱۰ می‌توان با پی گیری یک مثال چگونگی کارکرد یک کوکی را بررسی نمود. فرض کنید بیژن نامی از طریق مرورگر کامپیوتر شخصی خود برای نخستین بار از طریق وب به سامانه‌ی شرکت آمازون (Amazon.com) وارد شده است. فرض کنید وی قبلاً نیز از سامانه eBay دیداری داشته است. هنگامی‌که تقاضایی وارد وب سرور آمازون می‌شود، سرور یک شماره شناسه‌ی انحصاری را برای آن تقاضا تعریف و سپس آنرا در پایگاه داده‌ی خود که از طریق این شماره شناسه شاخص گذاری می‌شود وارد می‌نماید. سرور آمازون آنگاه در پاسخ به تقاضای مرورگر بیژن به همراه پیام پاسخ خود یک قطعه‌ی HTTP به صورت ; Set-Cookie: که شامل شماره‌ی شناسه است را ارسال می‌کند. به‌عنوان مثال سرپیام ارسالی ممکن است به‌صورت زیر باشد:

```
Set-Cookie: 1678
```

هنگامی‌که مرورگر بیژن پیام پاسخ HTTP را دریافت می‌کند. سرپیام اشاره شده را مشاهده و سپس مرورگر این عنوان را در قالب یک سطر به فایل کوکی خود که آنرا مدیریت می‌کند، اضافه می‌نماید. این خط اضافه شده شامل نام میزبان سرور و شماره‌ی شناسه در سرپیام: Set-Cookie می‌باشد. لازم به توضیح است فایل کوکی در حال حاضر دارای یک ورودی برای eBay می‌باشد، زیرا بیژن در گذشته از این سامانه نیز بازدیدی داشته است. هنگامی‌که او از درون صفحه‌های سامانه‌ی آمازون عبور می‌کند، هر مرتبه که بیژن یک صفحه‌ی وب را تقاضا می‌کند، مرورگر او ابتدا با فایل کوکی خود مشاوره می‌کند و شماره‌ی شناسایی او را برای این سامانه اخذ و سپس آن را در سرپیام کوکی که شامل شماره شناسه در پیام تقاضای HTTP است وارد می‌کند. به عبارت دیگر هر یک از تقاضاهای HTTP او به سرور آمازون شامل سرپیامی به‌صورت زیر است:

```
Cookie: 1678
```

1- Cookeis



شکل ۲- ۱۰ ♦ رهگیری وضعیت کاربر با استفاده از کوکی

به این‌گونه است که سرور آمازون قادر به رهگیری فعالیت‌های بیژن در سامانه مربوطه می‌شود. اگرچه سامانه‌ی وب آمازون لزوماً از نام بیژن و یا دقیقاً صفحه‌ای که کاربر 1678 بازدید نموده است و یا حتی زمان و ترتیب مراجعه‌ی وی آگاهی ندارد. آمازون از کوکی‌ها جهت گردش کار خدمات فروش خود استفاده می‌کند. به عبارت دیگر سابقه‌ای را از اقلامی که بیژن قصد خرید آنها را داشته است، نگهداری می‌کند. این موضوع به‌ویژه هنگام خرید واقعی جهت پرداخت جمعی اقلام خریداری شده در پایان یک مراجعه به سامانه، ضروری و سودمند می‌باشد. اگر بیژن پس از یک هفته یا بیشتر مجدداً بازدیدی را از سامانه‌ی آمازون داشته باشد، مرورگر او باز هم سرپیام Cookie:1678 را در پیام‌های تقاضای او وارد می‌نماید. آمازون هم چنین ممکن است محصولاتی را براساس صفحه‌های وب که بیژن در گذشته بازدید کرده است به وی پیشنهاد نماید. حال اگر او با سامانه‌ی آمازون از طریق وارد کردن مشخصات خود و نیز پست الکترونیکی، آدرس پستی و حتی شماره‌ی کارت اعتباری خود ثبت نام

نماید، آنگاه آمازون این اطلاعات را در پایگاه داده‌های خود وارد نموده و در نتیجه نام بیژن وابسته به شماره‌ی شناسه وی و همه‌ی صفحه‌هایی که وی در گذشته بازدید نموده است، می‌شود. علت آنکه اکثر سامانه‌های تجاری نظیر آمازون قادرند اصطلاحاً خدمات خرید به‌صورت **یکبار فشردن**[1] را برای مشتریان خود فراهم اورند همین موضوع است. به این ترتیب هنگامی‌که بیژن قصد خرید مجدد از سامانه‌ای نظیر آمازون را دارد، نیازی نیست که نام، آدرس و یا شماره‌ی کارت اعتباری خود را مجدداً وارد نماید. از این گفتار ملاحظه می‌شود که از کوکی‌ها برای شناسایی یک کاربر استفاده می‌شود. نخستین مرتبه که یک کاربر سامانه‌ای را بازدید می‌کند، او می‌تواند از طریق نام خود یک شناسه‌ی کاربرد ایجاد نماید. در طول مراجعه‌های بعدی مرورگر یک سرپیام کوکی را به سرور ارایه می‌دهد و بنابراین کاربر را به آن معرفی می‌کند. نتیجه آنکه از کوکی‌ها می‌توان برای تعریف یک لایه‌ی تعامل کاربرد بر روی HTTP از نوع بدون وضعیت استفاده نمود. به‌عنوان مثال هنگامی‌که یک کاربر وارد یک کاربُرد وب بر پایه پست الکترونیکی (e-mail) می‌شود، مرورگر اطلاعات کوکی را به سرور ارسال می‌کند و در نتیجه از طریق تعامل کاربر با برنامه‌ی کاربردی پست الکترونیکی به شناسایی او در طول مدت استفاده می‌پردازد. اگرچه کوکی‌ها مراحل خرید اینترنتی را ساده می‌کنند اما می‌توانند به نوعی هم دردسرساز باشند زیرا اطلاعات شخصی موجود در کوکی می‌تواند تهدیدآمیز نیز باشد. یک سامانه‌ی وب از طریق یک کوکی می‌تواند اطلاعات فراوانی را از کاربر بدست آورده و آنها را به ذینفع ثالثی به فروش رساند.

۲- ۹ نهان‌سازی وب

یک **وب نهان**[2] که از آن به‌عنوان **سرور جانشین**[3] نیز نام برده می‌شود، موجودیتی است که تقاضاهای HTTP را از جانب وب سرور اصلی پاسخ می‌دهد. یک وب نهان دارای امکانات ذخیره‌سازی مربوط به خود می‌باشد و نسخه‌هایی از آخرین عناصر تقاضا شده را در این حافظه نگهداری می‌کند. شکل ۲- ۱۱ نشان می‌دهد چگونه یک مرورگر می‌تواند به گونه‌ای پیکربندی شود که همه‌ی تقاضاهای HTTP یک کاربرد ابتدا به سمت وب نهان هدایت گردند. هنگامی‌که یک مرورگر در این قالب پیکربندی گردد، هر تقاضای مرورگر برای یک عنصر ابتدا به سمت وب نهان هدایت می‌شود. به‌عنوان مثال اگر یک مرورگر تقاضای زیر را درخواست نماید:

```
http://www.someschool.edu/compus.gif
```

آنچه در اثر این تقاضا رخ می‌دهد عبارت است از:

۱- مرورگر یک ارتباط TCP به وب نهان ایجاد می‌کند و یک تقاضای HTTP برای عنصر به وب نهان ارسال می‌کند.

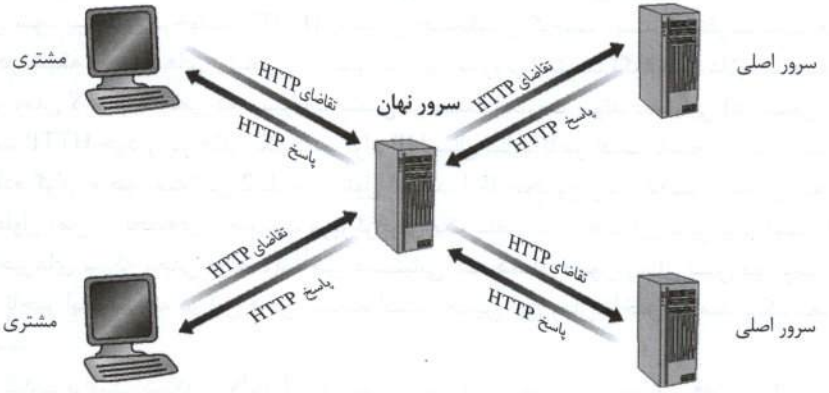
۲- وب نهان احتمال وجود نسخه‌ای از عنصر تقاضا شده را در حافظه‌ی محلی خود جستجو نموده و در صورت وجود، آن عنصر را به‌عنوان یک پیام پاسخ HTTP به مرورگر انتقال می‌دهد.

۳- اگر وب نهان حامل عنصر در حافظه‌ی خود نباشد یک ارتباط TCP با سرور اصلی می‌گشاید. به عبارت دیگر گشودن این ارتباط با www.SomeSchool.edu می‌باشد و سپس یک تقاضای

1- One Click Shopping
2- Cache web
3- Proxy Server

HTTP را برای آن عنصر به ارتباط TCP بین سرورهای نهان و اصلی گسیل می‌دارد. پس از دریافت تقاضا سرور اصلی شیئ مورد نظر را به وب نهان ارسال می‌کند.

۴- هنگامی‌که وب نهان عنصر را دریافت می‌کند، نسخه‌ای از آن را در حافظه‌ی محلی خود بارگذاری می‌کند و یک نسخه را از طریق یک پیام پاسخ HTTP به مرورگر با استفاده از ارتباط TCP موجود بین مرورگر و وب نهان گُسیل می‌نماید.



شکل ۲- ۱۱ ♦ تقاضای مشتری برای یک عنصر از طریق وب نهان

یادآوری این نکته ضروری است که یک وب نهان همزمان هر دو نقش سرور و مشتری را ایفا می‌کند. هنگامی‌که تقاضاهایی را از مرورگر دریافت می‌کند و پاسخ‌ها را به آن ارسال می‌نماید در نقش سرور ظاهر می‌شود و هنگامی‌که تقاضاها را ارسال و یا پاسخ‌ها را از سرور اصلی دریافت می‌کند عملکرد یک مشتری را دارد. معمولاً وب نهان به‌وسیله‌ی یک ISP خریداری و نصب می‌شود. به‌عنوان مثال یک دانشگاه ممکن است یک وب نهان را در شبکه‌ی مجتمع خود نصب نموده و همه‌ی مرورگرهای مجتمع را به سمت وب نهان هدایت کند. از طرفی یک ISP مسکونی مهم نظیر AOL[1] ممکن است یک یا چندین وب نهان را در شبکه‌ی خود نصب و مرورگرهای خود را از پیش جهت هدایت به سمت این سرورها پیکربندی نماید. نهان‌سازی وب به دو دلیل در اینترنت به‌کار گرفته شده است. نخستین علت آن است که یک وب نهان می‌تواند سبب کاهش قابل ملاحظه‌ی زمان پاسخگویی به تقاضاهای مشتریان باشد، به‌ویژه اگر حداکثر سرعت قابل تأمین مشتری و سرور اصلی به مراتب کمتر از حداکثر توان پهنای باند بین مشتری و وب نهان باشد. اگر در اکثر مواقع همانگونه که انتظار می‌رود یک ارتباط سریع بین مشتری و وب نهان موجود باشد و وب نهان نیز در حافظه‌ی خود عنصر تقاضا شده را داشته باشد، آنگاه وب نهان قادر است که شیئ را به سرعت در اختیار مشتری قرار دهد. علت دوم آن است که وب‌های نهان می‌توانند سبب کاهش چشمگیر ترافیک بر روی پیوندهای دستیابی یک نهاد به اینترنت باشند. با کاهش حجم ترافیک، شبکه‌ی یک شرکت یا دانشگاه نیازی به ارتقاء مرتب پهنای باند خود نخواهد داشت و بنابراین سبب کاهش هزینه برای آن نهاد خواهد شد. علاوه بر همه‌ی این موارد وب‌های نهان می‌توانند کاهش چشمگیر ترافیک در اینترنت را به گونه‌ی کلی سبب شوند که نتیجه‌ی آن افزایش کارایی همه‌ی کاربردها را به ارمغان می‌آورد.

1- American on Line

برای آنکه بتوان درک عمیقی از فواید وب‌های نهان بدست آورد به مثالی در چارچوب شکل ۲- ۱۲ اکتفا می‌شود. این شکل دو شبکه را نشان می‌دهد. شبکه‌ی یک نهاد و شبکه‌ی عمومی اینترنت. نهاد مورد اشاره در شکل یک شبکه‌ی محلی پرسرعت به همراه یک مسیریاب است که به روتر دیگری در اینترنت از طریق یک پیوند 15 Mbps متصل شده است. سرورهای اصلی در سراسر گیتی توزیع شده‌اند و به شبکه‌ی اینترنت پیوند خورده‌اند. فرض کنید میانگین اندازه‌ی حجم بیت هر شیئ یک مگابیت است و میانگین نرخ تقاضاها از سوی مرورگرهای نهاد به سرورهای اصلی برابر 15 تقاضا در ثانیه باشد. اگر فرض شود پیام‌های درخواست HTTP به میزان چشمگیری کوچک باشند که تقریباً سبب هیچگونه ترافیک در شبکه یا پیوندهای ارتباطی از مسیریاب نهاد به روتر اینترنت نگردند. و علاوه بر همه‌ی این موارد اگر زمان لازم از هنگامی‌که مسیریاب متصل در سمت اینترنت پیوند دستیابی (در شکل ۲- ۱۲) درخواست HTTP خود را در قالب یک داده گرام IP ارسال نماید تا دریافت پاسخ آن در قالب تعداد زیادی داده گرام به طور میانگین 2 ثانیه به طول انجامد. آنگاه مجموع زمان عکس العمل که عبارت است از طول زمان از لحظه‌ی درخواست مرورگر برای یک عنصر تا دریافت آن شیئ برابر است با حاصل جمع: تأخیرهای شبکه محلی (LAN)، تأخیر دستیابی که همان طول زمان بین دو روتر است و سرانجام تأخیر اینترنت که در این فرض 2 ثانیه است. تخمین این زمان تأخیر نیازمند یک محاسبه‌ی ساده است:

ابتدا شدت ترافیک شبکه‌ی LAN محاسبه می‌شود. این شدت بنا بر مطالب فصل یک به‌صورت زیر محاسبه می‌شود:

$$\text{شدت ترافیک LAN} = \text{تعداد تقاضا در ثانیه} \times \frac{\text{اندازه‌ی هر تقاضا}}{\text{سرعت پیوند تقاضا}}$$

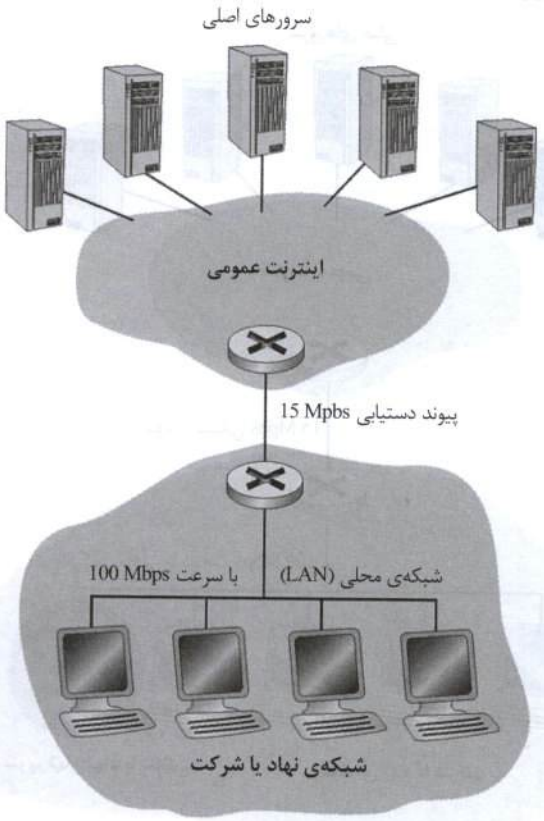
$$\text{شدت ترافیک LAN} = 15 \times \frac{1\ \text{Mbit}}{100\ \text{Mbps}} = 0.15$$

از طرفی شدت ترافیک پیوند دستیابی یعنی از مسیریاب اینترنت به روتر نهاد نیز برابر است با:

$$\text{شدت پیوند دستیابی} = 15 \times \frac{1\ \text{Mbit}}{15\ \text{Mbps}} = 1$$

شدت ترافیک 0.15 بر روی یک LAN به مفهوم آن است که حداکثر ده‌ها میلی ثانیه تأخیر را می‌توان انتظار داشت و بنابراین از تأخیر شبکه‌ی محلی می‌توان صرف نظر نمود. اما همانگونه که در فصل یک توضیح داده شد، هر چقدر شدت ترافیک به مقدار 1 نزدیک گردد مانند آنچه در پیوند دستیابی شکل ۲- ۱۲ رخ داده است، تأخیر بر روی پیوند بسیار طولانی و خارج از کنترل رشد می‌یابد. در نتیجه میانگین زمان پاسخ برای تقاضاها به سمت دقیقه‌ها رشد خواهد کرد که البته برای کاربران یک نهاد مانند دانشگاه و یا شرکت تجاری غیر قابل قبول است.

آنچه مسلم است آن است که بایستی تدبیری مطلوب به‌کار گرفته شود. یک راهکار ممکن افزایش نرخ دستیابی از 15Mbit به مثلاً 100Mbit می‌باشد. این افزایش سبب می‌شود شدت ترافیک بر روی پیوند دستیابی به مقدار 0.15 کاهش پیدا کند که نتیجه‌ی آن تأخیر ناچیز بین دو مسیریاب می‌باشد. در این حالت کل زمان پاسخ تقریباً به 2 ثانیه می‌رسد. یعنی برابر زمان تأخیر اینترنت. اما این راهکار مفهوم دیگری را نیز در پی دارد و آن این است که نهاد بایستی پیوند دستیابی خود را از 15Mbps به 100Mbps ارتقاء دهد که نتیجه‌ی آن افزایش هزینه را به دنبال می‌آورد.



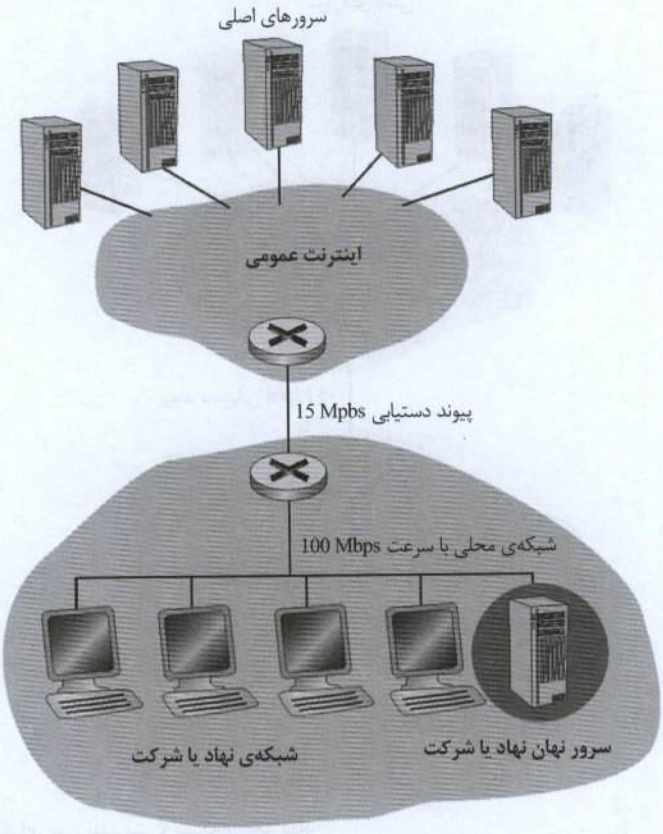
شکل ۲- ۱۲ ♦ تنگنای بین اینترنت و شبکه‌ی یک نهاد

اکنون راهکار جایگزین دیگری را در نظر بگیرید. در این راه حل از ارتقاء پیوند دستیابی خبری نیست اما در عوض یک وب نهان در شبکه‌ی آن نهاد نصب می‌گردد. شکل ۲- ۱۳ این راهکار را نشان می‌دهد. در عمل نرخ **برخورد موفق**[1] که عبارتست از درصد تعداد تقاضای پاسخ داده شده به‌وسیله‌ی وب نهان محدوده‌ای بین 0.2 تا 0.7 دارد. اگر فرض شود وب نهان نرخ برخورد موفق معادل 0.4 را برای نهاد در پی داشته باشد. از آنجائیکه برنامه‌ی مشتری و وب نهان از طریق یک LAN با سرعت یکسان به یکدیگر متصل شده‌اند در نتیجه 40 درصد تقاضاهای بی‌درنگ در یک طول زمانی 10 میلی ثانیه از سوی وب نهان پاسخگویی می‌شوند. اما 60 درصد باقیمانده بایستی پاسخ رضایت‌بخش خود را از سرور اصلی دریافت نمایند. عبور فقط 60 درصد از عناصر تقاضا شده از درون پیوند دستیابی سبب می‌شود شدت ترافیک بر روی پیوند از مقدار 1 به 0.6 کاهش پیدا کند. در عمل شدت ترافیک کمتر از 0.8 نتیجه‌اش تأخیر بسیار ناچیز است. این مقدار تأخیر برابر ده‌ها میلی‌ثانیه بر روی یک پیوند 15Mbps در مقایسه با تأخیر 2 ثانیه‌ی اینترنت بسیار ناچیز می‌باشد. با توجه به همه‌ی موارد اشاره شده میانگین زمان تأخیر در این حالت برابر است با:

$$\text{میانگین زمان تأخیر} = 0.4 \times 0.01\text{s} + 0.6 \times 2.01\text{s}$$

1- Hit rate

میانگین زمان تأخیر = 1.21s



شکل ۲-۱۳ ♦ افزودن یک وب نهان به شبکه‌ی یک نهاد

ملاحظه می‌شود این زمان اندکی بیش از 1.2 ثانیه است. بنابراین راهکار دوم زمان عکس العمل کوتاهتری را نسبت به راهکار اول به همراه دارد و نیازی به ارتقاء پیوند دستیابی به اینترنت نیز نمی‌باشد. بدیهی است هزینه‌ی خرید و نصب وب نهان هنوز بر دوش نهاد است اما این هزینه چندان چشمگیر نیست زیرا بسیاری از سرورهای نهان با استفاده از نرم‌افزارهای **محدوده‌ی عمومی**[1] که بر روی PCهای معمولی قابل اجرا می‌باشند، پیاده‌سازی و پیکربندی می‌شوند.

۲-۱۰ سرپیام GET شرطی

اگرچه نهان‌سازی سبب کاهش زمان لازم برای مبادله‌ی اطلاعات بین برنامه‌های مشتری و سرور می‌شود اما در هر حال چالش جدیدی را نیز به همراه دارد. نسخه برداری از یک عنصر ساکن در وب نهان خود سبب تحمیل تأخیر می‌شود. به عبارت دیگر شیئ ساکن در وب سرور ممکن است تا لحظه‌ی نسخه برداری آن به‌وسیله‌ی برنامه‌ی مشتری دستخوش تغییر و یا تجدید نظر شده باشد. از طرفی پروتکل HTTP دارای مکانیزمی است که به یک وب نهان اجازه می‌دهد تا عنصر را از لحاظ به روز بودن ارزیابی نماید این خود راهکاری برای چالش نام برده می‌باشد. این مکانیزم اصطلاحاً GET

1- Public domain Software

شرطی[1] نامیده می‌شود. یک پیام درخواست HTTP در حالی GET شرطی تلقی می‌شود که: (۱) پیام تقاضا از روش GET استفاده نماید و (۲) پیام حامل سرپیام: If-Modified-Since: باشد. برای نشان دادن عملکرد عبارت GET شرطی به مثالی که در ادامه می‌آید توجه نمائید: نخست فرض کنید برنامه‌ی سرور جانشین از سوی مرورگر متقاضی پیام تقاضایی را به برنامه‌ی وب سرور ارسال می‌کند:

```
GET/fruit/kiwi.gif HTTP/1.1
Host:www.exotiquecusine.com
```

دوم فرض کنید وب سرور پیام پاسخ را به همراه شیئ تقاضا شده به وب نهان گسیل می‌دارد:

```
HTTP/1.1 200 OK
Date: Sat, 7 Jul 2012 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Wed, 4 Jul 2012 09:23:24
Content-Type: image/gif

(data data data data data ...)
```

وب نهان عنصر را به مرورگر متقاضی ارسال می‌کند همچنین شیئ را نیز در حافظه‌ی محلی خود ذخیره می‌کند. از همه مهمتر آنکه وب نهان آخرین نسخه‌ی تجدید نظر شده‌ی عنصر را به همراه شیئ اصلی ذخیره می‌کند. سوم اگر یک هفته بعد برنامه‌ی مرورگر دیگری همان عنصر را از طریق وب نهان تقاضا نماید و شیئ مورد نظر هنوز در حافظه‌اش باشد، از آنجائیکه ممکن است عنصر در طول هفته‌ی گذشته دستخوش تجدید نظر در وب سرور شده باشد آنگاه وب نهان ارزیابی به روز بودن آن را از طریق صدور فرمان GET شرطی انجام می‌دهد. آنچه وب نهان به‌عنوان عبارت GET شرطی می‌فرستد به‌صورت زیر است:

```
GET/fruit/Kiwi.gif HTTP/1.1
Host: www.exotiquecusine.com
If-Modified-Since: Wed, 4 Jul 2012 09:23:24
```

ملاحظه می‌شود ارزش سرپیام: If-Modified-Since عیناً مشابه ارزش عبارت سرپیام Last-Modified: که سرور یک هفته‌ی پیش ارسال نموده است، می‌باشد. عبارت GET شرطی به سرور می‌گوید نسبت به ارسال عنصر در صورتی اقدام نماید که از آخرین تاریخ مشخص شده در آن تجدید نظر یا تغییری صورت پذیرفته باشد. فرض کنید شیئ مورد نظر نسبت به آخرین تاریخ یعنی 4 ژوئیه‌ی 2009 ساعت 09:23:24 هیچگونه تغییری نکرده باشد آنگاه در ارزیابی چهارم، وب سرور پیام پاسخ را به وب نهان در قالب ذیل ارسال می‌کند:

```
HTTP/1.1 304 Not Modified
Date: Sat, 14 Jul 2012 15:39:29
Server: Apache/1.3.0 (Unix)

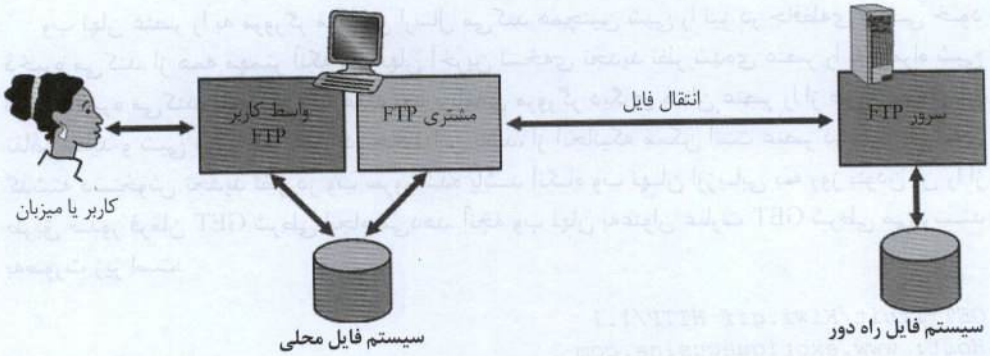
(empty entity body)
```

1- Conditional GET

ملاحظه می‌شود که در پاسخ عبارت GET شرطی باز هم وب سرور اقدام به گسیل پیام پاسخ می‌کند، اما همراه آن پیام، عنصر مورد تقاضا ارسال نمی‌گردد. نتیجه‌ی همراه نمودن شیئ مورد تقاضا سبب اتلاف پهنای باند و افزایش زمان مبادله بین کاربر و سرور می‌شود به‌ویژه اگر عنصر از لحاظ حجم بزرگ باشد. توجه به این نکته ضروری است که آخرین پیام دارای خط وضعیت: 304 Not Modified می‌باشد که در حقیقت مفهوم آن ارسال نسخه‌ی عنصر نهان از سرور جانشین به مرورگر متقاضی می‌باشد. بحث مربوط به پروتکل HTTP یعنی نخستین پروتکل اینترنت یا به عبارت دیگر پروتکل لایه‌ی کاربرد به انتها می‌رسد.

۲- ۱۱ انتقال فایل FTP

در تعامل انتقال فایل کاربر از طریق یک میزبان (میزبان محلی) مایل است فایل‌ها را به یک میزبان راه دور انتقال و یا از آن دریافت نماید. معمولاً برای آنکه کاربر بتواند به منبع فایل‌ها دسترسی پیدا کند مجبور است یک شناسه‌ی کاربری و رمز عبور را ارائه دهد. پس از فراهم‌آوری اطلاعات اعتبارسنجی، کاربر مجاز به انتقال در دو سوی ارتباط می‌شود. همانگونه که شکل ۲- ۱۴ نشان می‌دهد. کاربر تعامل با FTP را از طریق یک عامل کاربری FTP انجام می‌دهد.

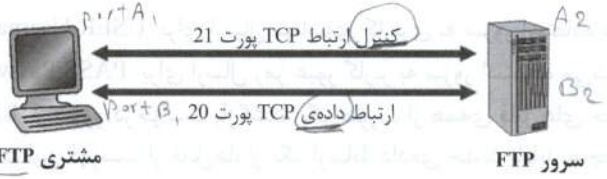


شکل ۲- ۱۴ ♦ پروتکل انتقال فایل (FTP) فایل‌ها را بین سیستم‌های فایل محلی و راه دور جابجا می‌کند

برنامه‌ی کاربرد ابتدا **نام میزبان**[1] مورد نظر را به طرف مقابل معرفی می‌کند. با انجام این عمل برنامه‌ی مشتری پروتکل انتقال فایل (FTP) در میزبان محلی یک ارتباط TCP را با پردازش سرور FTP در میزبان طرف مقابل برقرار می‌نماید. کاربر آنگاه شناسه‌ی کاربری و رمز عبور را از طریق ارتباط TCP به‌عنوان بخشی از فرمان‌های FTP ارسال می‌دارد. هنگامی‌که سرور اعتبارسنجی مورد نیاز را انجام دهد برنامه‌ی کاربر اقدام به نسخه برداری از یک یا چند فایل ذخیره شده از درون سیستم فایل راه دور می‌کند و آن نسخه‌ها را به سیستم فایل محلی خود انتقال می‌دهد. بدیهی است عکس این عمل نیز امکان‌پذیر است. پروتکل‌های HTTP و FTP هر دو عملیات انتقال فایل را انجام می‌دهند و از این لحاظ مشخصات مشترک زیادی با یکدیگر دارند. یکی از این موارد مشترک عبارت است از آنکه: هر دو در بالای TCP اجرا می‌شوند. اما به هر حال دو پروتکل لایه کاربرد دارای اختلاف‌های فراوانی نیز می‌باشند. یکی از مهم‌ترین آنها ملاحظه‌ی این موضوع است که FTP از دو ارتباط موازی TCP برای

1- Host name

انتقال فایل استفاده می‌کند. این دو ارتباط عبارتند از (۱) **یک ارتباط کنترل**[1] و (۲) **ارتباط داده**[2]. از ارتباط کنترل جهت ارسال اطلاعات کنترل بین دو میزبان استفاده می‌شود این اطلاعات شامل: شناسایی کاربر، رمز عبور، فرمان‌های تغییر فهرست در میزبان راه دور و فرمان‌های "PUT" و "get" جهت ارسال و دریافت فایل‌ها می‌باشند. از سوی دیگر ارتباط داده جهت گسیل یک فایل می‌باشد. از آنجائیکه FTP از یک ارتباط کنترل مجزا استفاده می‌کند اصطلاحاً گفته می‌شود FTP اطلاعات کنترل خود را **خارج از محدوده**[3] ارسال می‌کند. در فصل ۷ ملاحظه خواهد شد که از پروتکل جریان بی‌درنگ (RTSP) علاوه بر استفاده جهت کنترل انتقال پوسته‌ی رسانه‌ای مانند صدا و تصویر برای ارسال اطلاعات کنترل خارج از محدوده نیز استفاده می‌شود. لازم به یادآوری است پروتکل HTTP برای ارسال سرپیام‌های تقاضا و پاسخ از همان ارتباط TCP مشابه که انتقال فایل را نیز انجام می‌دهد، استفاده می‌نماید. به همین دلیل پروتکل HTTP از لحاظ چگونگی ارسال اطلاعات کنترل **درون محدوده**[4] نامیده می‌شود. در بخش‌های بعدی ملاحظه می‌گردد که پروتکل SMTP که موتور اصلی پست الکترونیکی است اطلاعات کنترل را به‌صورت درون محدوده انتقال می‌دهد. شکل ۲- ۱۵ ارتباط‌های کنترل و داده را در پروتکل انتقال فایل (FTP) نشان می‌دهد.



شکل ۲- ۱۵ ♦ ارتباط‌های کنترل و داده در پروتکل انتقال فایل

هنگامی‌که کاربر یک تعامل FTP را آغاز می‌کند. پروتکل FTP در سمت برنامه‌ی مشتری (کاربر) ابتدا یک ارتباط کنترل TCP را به سوی سرور (میزبان راه دور) از طریق پورت شماره‌ی 21 در سرور ایجاد می‌کند. آنگاه FTP در سمت مشتری شناسه‌ی کاربری و رمز عبور را از طریق این ارتباط کنترل ارسال می‌کند. علاوه بر آن FTP مشتری فرمان‌های مربوط به تغییر **فهرست**[5] راه دور را نیز از طریق همین ارتباط گسیل می‌کند. هنگامی‌که برنامه‌ی سمت سرور فرمان انتقال فایل را از طریق ارتباط کنترل دریافت نماید (این فرمان می‌تواند به طرف میزبان راه دور و یا از طرف آن باشد)، کاربرد سمت سرور یک ارتباط داده‌ی TCP را به سوی مشتری ایجاد می‌کند. پروتکل FTP دقیقاً یک فایل را از طریق این ارتباط داده ارسال و سپس آن را قطع می‌کند. اگر در طول همان تعامل کاربر انتقال دیگری را لازم داشته باشد آنگاه پروتکل FTP یک ارتباط داده‌ی دیگر را می‌گشاید. بنابراین در FTP ارتباط کنترل در سراسر زمان تعامل کاربر گشوده باقی می‌ماند. در صورتی‌که برای هر انتقال فایل یک ارتباط داده‌ی جدید بایستی در طول تعامل تعریف شود. این به مفهوم آن است که ارتباط داده غیر ماندگار است.

1- Control Connection
2- Data Connection
3- Out of band
4- In band
5- Directory

در سراسر عملیات تعامل، سرور FTP بایستی اطلاعات **موقعیت**[1] کاربر را نگهداری نماید. به‌ویژه سرور بایستی بین ارتباط کنترل با یک حساب کاربری مشخص وابستگی‌های لازم را تعریف نموده و سابقه‌ی فهرست جاری کاربر را به هنگام حرکت او در درخت فهرست میزبان راه دور رهگیری کند. نگهداری اثر پای اطلاعات وضعیت برای هر کاربر در حال تعامل محدودیت‌های چشمگیری را از لحاظ تعداد تعامل هایی که FTP به طور همزمان قادر به انجام آن می‌باشد، سبب می‌گردد. لازم به یادآوری است قبلاً توضیح داده شد که پروتکل HTTP بدون موقعیت می‌باشد به عبارت دیگر نیازی به رهگیری وضعیت هیچ یک از کاربران نمی‌باشد.

پایان‌بخش این مطلب بررسی فرمان‌ها و پاسخ‌های متداول پروتکل انتقال فایل (FTP) می‌باشد. فرمان‌ها از مشتری به سرور و پاسخ‌ها از سرور به مشتری همگی از طریق ارتباط کنترل در قالب کُدهای 7 بیت اسکی ارسال می‌شوند. بنابراین مشابه فرمان‌های HTTP، فرمان‌های پروتکل FTP از طریق افراد (طرف ثالث) قابل خواندن می‌باشند. برای ایجاد تمایز بین فرمان‌ها از کلید **بازگشت نورد**[2] در انتهای هر یک استفاده می‌شود. هر فرمان FTP شامل تعداد چهار حرف بزرگ اسکی می‌باشد که برخی از آنها نیز شناسه‌های اختیاری دارند. تعدادی از فرمان‌های متداول FTP در ذیل آمده است:

- فرمان :USER Username برای ارسال شناسه‌ی کاربری به سرور استفاده می‌شود.
- فرمان:PASS Password برای ارسال رمز عبور کاربر به سرور استفاده می‌شود.
- فرمان :LIST از سرور درخواست بازگشت یک صورت از همه‌ی فایل‌های جاری در فهرست راه دور را می‌کند. این فهرست از فایل‌ها، از یک ارتباط داده‌ی جدید ناپایا به جای ارتباط کنترل TCP استفاده می‌کند.
- فرمان :RETR filename جهت بازیابی (get) یک فایل از فهرست جاری میزبان راه دور به‌کار گرفته می‌شود. این فرمان سبب می‌شود میزبان راه دور یک ارتباط داده‌ی TCP را ایجاد و فایل تقاضا شده را از طریق آن ارسال نماید.
- فرمان :STOR filename جهت ذخیره‌ی (Put) یک فایل در فهرست جاری سرور راه دور استفاده می‌شود.

بدیهی است یک رابطه به گونه‌ی یک به یک بین فرمان‌های صادر شده از سوی کاربر و FTP از طریق ارتباط کنترل TCP وجود دارد. به دنبال هر فرمان یک پاسخ می‌باشد که از سوی سرور به مشتری گسیل می‌شود. پاسخ‌ها همگی در قالب ارزشهای عددی سه رقمی می‌باشند. همراه آنها یک پیام اختیاری است. از لحاظ ساختاری آنها مشابه کُد وضعیت و عبارت مربوط به آن در خط وضعیت پیام پاسخ HTTP می‌باشند. برخی از این فرمان‌ها و پاسخ‌ها در ذیل آمده است:

- `331 Username Ok, Password required`
- `125 Data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error Writing file`

خوانندگان علاقه‌مند به فرمان‌های FTP و پاسخ‌های آنها ترغیب به مطالعه‌ی مستند RFC شماره‌ی 959 [RFC959] می‌شوند.

1- State
2- Carriage return

از اینجا تا ۱۳۶

۲-۱۲ پست الکترونیکی در اینترنت

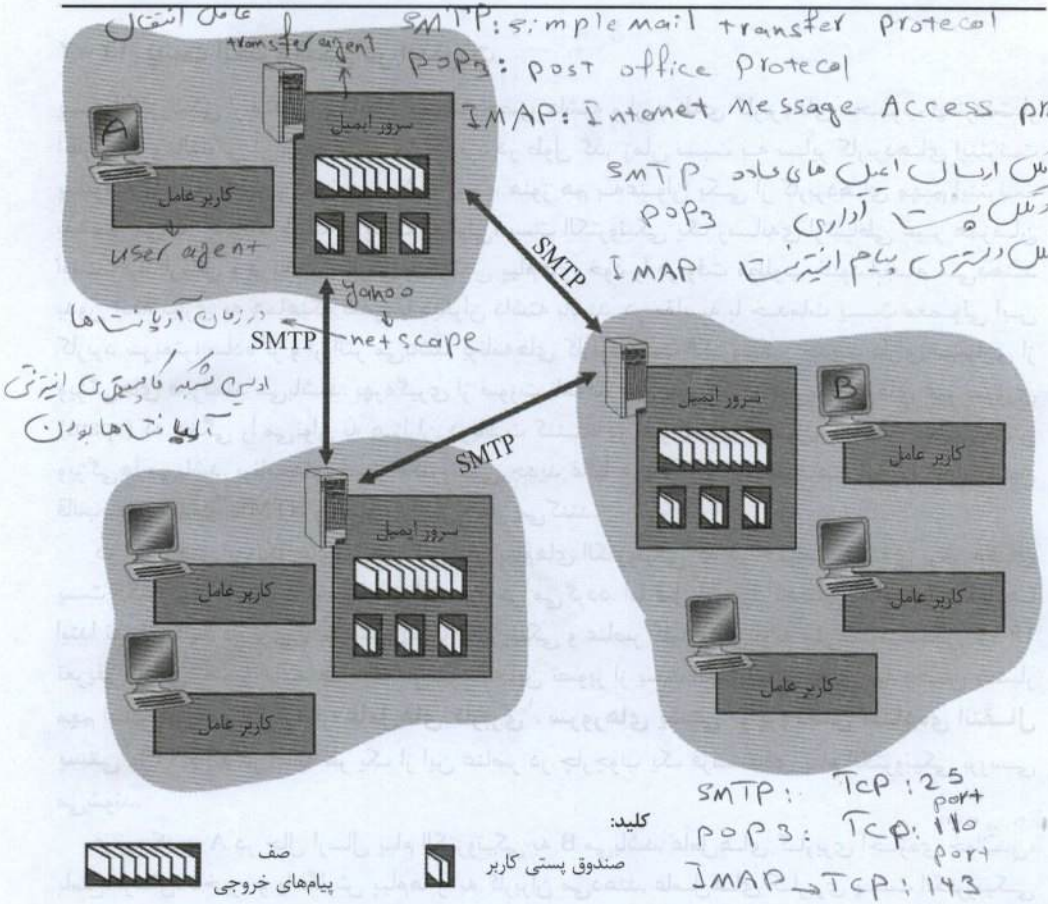
پست الکترونیکی از ابتدای پیدایش اینترنت حضور داشته و از جمله‌ی کاربردهای محبوب اینترنت از آغاز رشد و بالندگی آن بوده است. این کاربرد در طول گذر زمان نسبت به سایر کاربردهای اینترنت پیشرفته تر، پیچیده‌تر و قدرتمندتر شده است. هنوز هم به‌عنوان یکی از کاربردهای مهم اینترنت به‌شمار می‌آید. همانند شبکه‌ی پُست معمولی، پست الکترونیکی یک رسانه‌ی ارتباطی غیر همزمان است. افراد ارسال و دریافت، نگارش و خواندن پیام‌های خود را در وقت مطلوب خود انجام می‌دهند بدون آنکه نیازی به هماهنگ شدن با دیگران داشته باشند. در مقایسه با خدمات پست معمولی این کاربرد سریعتر، ساده تر و ارزانتر می‌باشد. برنامه‌های کاربرد پست الکترونیکی جدید دارای بسیاری از ویژگی‌های قدرتمند می‌باشند. بهره‌گیری از صورت نامه‌ها، پیام‌های الکترونیکی و پیام‌های غیر ضروری (spam) که همگی را می‌توان به هزاران دریافت کننده در هر لحظه ارسال نمود از جمله‌ی این ویژگی‌ها می‌باشد. برنامه‌های پست الکترونیکی جدید غالباً همراه خود پیوست‌ها، ابرپیوندها، متون قالب‌بندی شده‌ی HTML و تصاویر را نیز حمل می‌کنند.

در این بخش پروتکل‌های لایه‌ی کاربردی پیام‌های الکترونیکی که در حقیقت هسته‌ی برنامه‌های پست الکترونیکی اینترنت به‌شمار می‌آیند بررسی می‌گردد. اما قبل از ورود تحلیلی به این پروتکل‌ها ابتدا تصویر گسترده تری از سیستم پست الکترونیکی و عناصر کلیدی آن ارائه می‌شود. شکل ۲-۱۶ تعریفی از این تصویر است. ملاحظه می‌شود که این تصویر از پست الکترونیکی دارای سه بخش بسیار مهم است. این سه عبارتند از: **عامل‌های کاربری**[1]، **سرورهای پستی**[2] و **پروتکل ساده‌ی انتقال پستی**[3] (SMTP). در ادامه هر یک از این عناصر در چارچوب یک فرستنده‌ی پیام الکترونیکی بررسی می‌شوند.

فرض کنید A در حال ارسال پیام الکترونیکی به B می‌باشد. عامل‌های کاربری اجازه‌ی خواندن، پاسخ، ارسال، ذخیره و یا نگارش پیام‌ها را به کاربران می‌دهند. عامل‌های کاربری پست الکترونیکی اصطلاحاً **نامه‌خوان‌ها**[4] نامیده می‌شوند اگرچه در این اثر از این عبارت استفاده نمی‌شود. هنگامی‌که A به نگارش نامه‌ی خود پایان می‌دهد، عامل کاربری او پیام را به سرور پستی او منتقل می‌کند.

در آنجا پیام به صف پیام‌های خروجی انتقال و در آن ذخیره می‌شود. هنگامی‌که B مایل به خواندن یک پیام است، عامل کاربری B پیام را از صندوق پستی وی واقع در سرور پستی او بازیابی می‌کند. در اواخر دهه‌ی 1990 میلادی عامل **واسط کاربر گرافیکی**[5] (GUI) تبدیل به یکی از مشهورترین عامل‌های کاربردی شد زیرا امکان رویت و نگارش پیام‌های چند رسانه‌ای را در اختیار کاربران قرار می‌داد. در حال حاضر برنامه کاربردی **مایکروسافت**[6] (OE)، پست آپل[7] و مرغ توفان موزیلا[8] (MT) از جمله معروفترین عامل‌های کاربر GUI به‌شمار می‌آیند.

1- User agents
2- Mail servers
3- Simple Mail Transfer Protocol
4- Mail readers
5- Graphical user interface
6- Outlook express
7- Apple mail
8- Mozilla Thunderbird



شکل ۲- ۱۶ ♦ تصویر سطح پیشرفته‌ی سیستم پست الکترونیکی اینترنت

بدیهی است واسط‌های کاربری بی شمار دیگری که بر مبنای متن عمل می‌کنند مانند: پست (mail)، صنوبر (Pine) و نارون (elm) نیز وجود دارند. البته واسط‌های بر پایه‌ی وب نیز در سوی دیگر این ماجرا قرار می‌گیرند که هسته‌ی زیرساخت پست الکترونیکی به‌شمار می‌آیند. این هسته همان سرورهای پست الکترونیکی هستند. هر دریافت کننده‌ای مانند B دارای یک صندوق پستی در یکی از این سرورها می‌باشد. صندوق پستی پیام‌های ارسالی به خود را مدیریت و نگهداری می‌کند. یک پیام در آغاز سفر خود از عامل کاربری فرستنده از درون برنامه‌ی سرور فرستنده عبور و سپس به برنامه‌ی سرور گیرنده می‌رسد. آن پیام در صندوق پستی گیرنده ذخیره و مدیریت می‌شود. هنگامی‌که به‌عنوان مثال B مایل به دستیابی نامه‌ی خود از صندوق پستی می‌باشد، برنامه‌ی سرور مربوطه از طریق صندوق پستی او اطلاعات اعتبارسنجی شامل شناسه‌ی کاربری و رمز عبور را بررسی نموده و سپس به وی اجازه‌ی دستیابی به اطلاعات را می‌دهد. اگر برنامه‌ی سرور A از طرف دیگر نتواند نامه را به برنامه‌ی سرور B تحویل نماید، سرور A پیام را در یک صف پیام نگهداری می‌کند و تلاش می‌کند که در فرصت بعدی آن را انتقال دهد. تلاش مجدد برای ارسال معمولاً هر 30 دقیقه یکبار رخ می‌دهد و چنانچه پس از گذشت چند روز تلاش‌ها موفقیت‌آمیز نباشد برنامه‌ی سرور، پیام را از صف حذف و A را با ارسال یک پیام الکترونیکی از موضوع آگاه می‌کند.

پروتکل SMTP پروتکل لایه‌ی کاربرد اصلی برای سرویس در اینترنت می‌باشد. این پروتکل از سرویس انتقال مطمئن داده در TCP استفاده می‌کند. مانند اغلب پروتکل‌های لایه کاربرد، SMTP دارای دو سمت می‌باشد: (۱) یک سوی مشتری که بر روی سرور پستی ارسال کننده اجرا می‌گردد و (۲) سوی سرور که بر روی برنامه‌ی سرور پستی دریافت کننده اجرا می‌شود. هر دو برنامه‌ی مشتری و سرور SMTP بر روی هر یک از برنامه‌های سرور پستی پیاده‌سازی می‌شوند. هنگامی‌که یک برنامه‌ی سرور پستی نامه‌هایی را به برنامه‌ی سرور پستی طرف مقابل ارسال می‌کند نقش یک برنامه‌ی مشتری SMTP را دارد و هنگامی‌که پیامی را دریافت می‌کند به‌عنوان برنامه‌ی سرور SMTP شناخته می‌شود.

- **SMTP**

جزئیات پروتکل انتقال ساده‌ی پست الکترونیکی (SMTP) در سند RFC 5321 مشخص شده است. این پروتکل هسته‌ی اصلی این کاربرد اینترنتی است. همانگونه که قبلاً اشاره شد SMTP به انتقال پیام‌ها بین برنامه‌های سرور پُستی فرستنده و گیرنده می‌پردازد. این پروتکل از قدمت بیشتری نسبت به HTTP برخوردار است و پیشینه‌ی آن به سال 1982 میلادی و حتی عقب تر باز می‌گردد. اگرچه SMTP از کیفیت‌های بی شمار و شگفت‌انگیزی بهره‌مند است که این شهرت در اینترنت قابل لمس می‌باشد اما در هر حال یک فناوری به جا مانده از گذشته است. زیرا شامل برخی از ویژگی‌های منسوخ نیز می‌باشد. به‌عنوان مثال تعریف محدودیت 7 بیت کُدهای اسکی برای بدنه‌ی انواع پیام‌های پست الکترونیکی از جمله‌ی این موارد می‌باشد، که تنها شامل سرپیام‌ها نمی‌گردد. تأثیر این محدودیت در اوایل دهه‌ی 1980 میلادی بیش از گذشته محسوس شد، زیرا در آن هنگام کمبود ظرفیت انتقال (سرعت) به شدت اهمیت خود را نشان داد چرا که قبل از آن از فایل‌های پست الکترونیکی با حجم وسیع و با پیوست‌های تصاویر پر حجم، صدا و یا ویدیو استفاده نشده بود. اما امروز در عصر کاربردهای چند رسانه‌ای محدودیت 7 بیت کُدهای اسکی کمی ناخوشایند و دور از ذهن به نظر می‌رسد. زیرا داده‌های چند رسانه‌ای باینری بایستی به‌صورت کُدهای اسکی رمزگذاری و سپس از طریق SMTP ارسال گردند. در طرف مقابل پیام اسکی بایستی مجدداً به باینری تبدیل گردد. این در حالی است که در بخش ۲- ۶ گفته شد پروتکل HTTP نیازمند رمزگذاری داده‌های چند رسانه‌ای به کُدهای اسکی قبل از انتقال نمی‌باشد. برای نشان دادن عملکرد بنیادی SMTP به مثال ذیل توجه نمائید: فرض کنید A و B مایل به تعامل یک پیام ساده‌ی اسکی می‌باشند:

۱- از طریق عامل کاربری پست الکترونیکی A درخواست ارسال پیام می‌نماید. این تقاضا را با فراهم‌آوری آدرس پست الکترونیکی B مانند B@someschool.edu، نگارش یک پیام و فرمان ارسال صادر می‌کند.

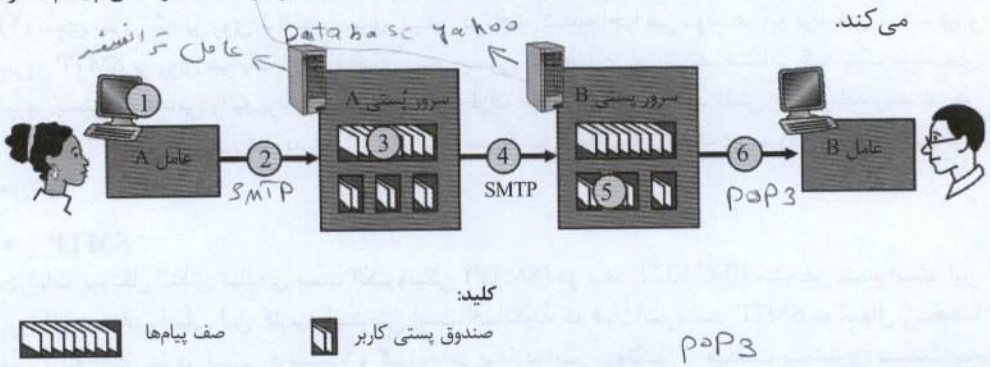
۲- عامل کاربری A پیام را به برنامه‌ی سرور پستی او گسیل می‌دارد و آن پیام در صف پیام قرار می‌گیرد.

۳- پروتکل SMTP سمت برنامه‌ی مشتری که بر روی سرور پستی A در حال اجرا می‌باشد، پیام را در صف مشاهده و در نتیجه یک ارتباط TCP را با یکی از پروتکل‌های SMTP در حال اجرا بر روی برنامه‌ی سرور پستی B برقرار می‌کند.

۴- پس از انجام عملیات دست‌دهی SMTP، پروتکل SMTP مشتری پیام A را وارد ارتباط TCP می‌کند.

۵- برنامه‌ی سرور پستی B در پروتکل SMTP سمت سرور، پیام را دریافت و آنرا از طریق برنامه‌ی سرور پستی B وارد صندوق پستی او می‌کند.

۶- آنگاه B از طریق عامل کاربری خود در یک زمان مناسب درخواست خواندن پیام A را می‌کند.



شکل ۲- ۱۷ ♦ تعامل پیام الکترونیکی بین A و B

شکل ۲- ۱۷ این ماجرا را به تصویر کشیده است. آنچه حائز اهمیت است ملاحظه‌ی این نکته است که پروتکل SMTP معمولاً از برنامه‌های سرور پستی میانی جهت ارسال پیام الکترونیکی استفاده نمی‌کند، حتی اگر دو سرور پستی در دو گوشه‌ی گیتی قرار گرفته باشند. اگر سرور A در توکیو و سرور B در تورنتو باشد، ارتباط TCP یک تعامل مستقیم بین برنامه‌ی سرورهای توکیو و تورنتو خواهد بود. به‌ویژه اگر سرور پستی B دچار اشکال شود پیام الکترونیکی در سرور پستی A باقی می‌ماند و منتظر یک تلاش جدید برای ارسال می‌شود به عبارت دیگر پیام در یک سرور پستی میانی نگهداری نمی‌شود. اکنون وضعیتی را در نظر بگیرید که در آن چگونگی انتقال یک پیام از یک سرور پستی ارسال کننده به یک سرور پستی دریافت کننده با جزئیات عمیق تر هدف می‌باشد. با اندکی تأمل می‌توان دریافت که پروتکل SMTP از بسیاری جهات شبیه به پروتکل تعامل چهره به چهره بین دو انسان است. نخست SMTP مشتری که در میزبان برنامه‌ی سرور پست فرستنده است یک ارتباط TCP را بر روی پورت شماره‌ی 25 در سرور SMTP و در میزبان دریافت کننده می‌گشاید. اگر سرور پستی میزبان دریافت فعال نباشد (دچار عیب باشد)، سرور مشتری در زمان دیگری تلاش خود را انجام می‌دهد. هنگامی‌که این ارتباط برقرار شد برنامه‌های سرور و مشتری به انجام نوعی دست‌دهی لایه‌ی کاربرد اقدام می‌کنند. به همان صورت که معمولاً دو نفر به هنگام تعادل اطلاعات بین خود ابتدا به معرفی یکدیگر می‌پردازند. بنابراین سرورها و مشتری‌های پروتکل SMTP قبل از انتقال اطلاعات خود را به یکدیگر معرفی می‌کنند. در طول انجام عملیات دست‌دهی این SMTP است که آدرس پست الکترونیکی فرستنده‌ای را که به تولید پیام مبادرت ورزیده است از طریق برنامه‌ی مشتری آن معرفی می‌کند و البته همین عمل را برای معرفی آدرس پست الکترونیکی گیرنده انجام می‌دهد. هنگامی‌که SMTP مشتری و سرور عملیات معارفه را انجام دادند آنگاه پیام از سوی برنامه‌ی مشتری ارسال می‌گردد. بنابراین پروتکل SMTP می‌تواند با تکیه بر سرویس انتقال مطمئن داده که TCP آنرا فراهم می‌آورد پیام‌های الکترونیکی را بدون خطا از سرور پستی خود دریافت نماید. برنامه‌ی مشتری این فرآیند را از طریق همان ارتباط مشابه در صورت وجود پیام‌های دیگری جهت ارسال به سرور تکرار می‌کند در غیر این صورت از TCP می‌خواهد که ارتباط را قطع نماید.

اکنون به بررسی متن پیام مبادله شده بین یک مشتری SMTP بنام C و یک سرور SMTP با نام S می‌پردازیم. نام میزبان مشتری Crepes.fr و نام میزبان سرور hamburger.edu می‌باشد. هر یک از سطرهای اسکی با پیشوند :C عیناً عباراتی هستند که مشتری به درون سوکت TCP خود ارسال می‌کند. مشابه آن هر یک از خطوط با پیشوند :S دقیقاً عباراتی هستند که سرور به درون سوکت TCP خود گسیل می‌کند. متن محاوره‌ی ذیل بی‌درنگ پس از گشوده شدن ارتباط TCP صورت می‌پذیرد:

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <A@crepes.fr>
S: 250 A@crepes.fr... Sender Ok
C: RCPT TO: <B@hamburger.edu<
S: 250 B@hamburger.edu... Recipient Ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
S: How about pickles?
C: .
S: 250  Message accepted for delivery
C: QUIT
S: 221 hamburger.ed closing connection
```

در مثال بالا برنامه‌ی مشتری یک پیام به صورت:

```
Do you like ketchup? How about pickles?
```

را از طریق سرور پستی Crepes.fr به سمت برنامه‌ی سرور پستی hamburger.edu ارسال می‌کند. به‌عنوان بخشی از محاوره برنامه‌ی مشتری تعداد 5 فرمان HELO، MAIL، RCPT TO، DATA و QUIT را نیز صادر می‌کند. بدیهی است مفاهیم این فرمان‌ها واضح است و نیازی به توضیح ندارد. برنامه‌ی مشتری علاوه بر موارد اشاره شده یک سطر را نیز ارسال می‌کند که حامل " • " نقطه می‌باشد و از طریق ارسال آن پایان پیام را به سرور اعلام می‌کند. در ادبیات نگارش اسکی هر پیام بایستی با عبارت CRLF•CRLF[1] پایان پذیرد که مفهوم این عبارت فشردن کلید بازگشت نورد به همراه یک فاصله‌ی خطی است. برنامه‌ی سرور برای هر فرمان یک پاسخ صادر می‌کند. هر پاسخ به همراه خود دارای یک کُد پاسخ و توضیحات اختیاری بزبان انگلیسی می‌باشد. پروتکل SMTP از ارتباط‌های پایا استفاده می‌کند بنابراین اگر سرور پستی فرستنده چندین پیام برای ارسال به یک سرور گیرنده‌ی مشابه دارد، برنامه‌ی فرستنده می‌تواند همه‌ی پیام‌ها را از طریق همان ارتباط TCP گسیل نماید. برای هر پیام برنامه‌ی مشتری، فرآیندی را با یک عبارت جدید به‌صورت MAIL FROM: Crepes.fr که در انتهای پیام با یک نقطه‌ی (•) فاصله دار همراه می‌باشد، آغاز می‌کند و هنگامی‌که همگی پیام‌ها ارسال شدند فرمان QUIT را صادر می‌کند. برای درک بهتر موضوع پیشنهاد می‌گردد با استفاده از Telnet یک محاوره‌ی مستقیم با سرور SMTP را تجربه نمائید. جهت انجام این موضوع ابتدا فرمان زیر را در مقابل نشان بی‌درنگ (C:\) وارد نمائید.

```
telnet serverName 25
```

1- Carriage return line feed

در این عبارت Server Name نام سرور پست محلی است. هنگامی‌که این عمل را انجام دهید یک ارتباط TCP بین برنامه‌های میزبان محلی و سرور پستی برقرار می‌نمائید. پس از وارد نمودن این خط بلافاصله بایستی پاسخ 220 را از سرور دریافت نمائید. پس از فرمان‌های پنج گانه‌ی اشاره شده و دو فرمان CRLF•CRLF و QUIT را در وقت مناسب آن صادر کنید.

• مقایسه SMTP و HTTP

در آغاز مقایسه‌ی دو پروتکل ابتدا یادآوری این نکته ضروری است که هر دو پروتکل جهت انتقال فایل‌ها از یک میزبان به میزبان دیگر مورد استفاده قرار می‌گیرند. پروتکل HTTP فایل‌ها را از یک وب سرور به یک مشتری وب انتقال می‌دهد که معمولاً آنها را اشیاء می‌خوانند. برنامه‌ی مشتری وب نیز همان مرورگر است، در صورتی‌که پروتکل SMTP انتقال فایل‌ها که همان پیام‌های پست الکترونیکی هستند را از یک برنامه‌ی سرور به سرور پست دیگر انتقال می‌دهد. هنگامی‌که انتقال فایل‌ها در حال انجام است هر دو پروتکل HTTP ماندگار و SMTP از ارتباط‌های پایا استفاده می‌کنند. بنابراین هر دو دارای مشخصات مشابه یکدیگر می‌باشند. اما در هر حال تفاوت‌های مهمی نیز بین آنها وجود دارد. نخست آنکه HTTP اصطلاحاً یک **پروتکل استخراج**[1] است. از آن جهت که اطلاعات بر روی یک وب سرور بارگذاری می‌شود و کاربر با استفاده از HTTP آن اطلاعات را از برنامه‌ی سرور در وقت مناسب خود بیرون می‌کشد. به‌ویژه آنکه ارتباط TCP به‌وسیله‌ی یک ماشین (نرم‌افزار) که مایل به دریافت فایل است تشکیل می‌شود. از طرف دیگر SMTP یک **پروتکل تزریق**[2] می‌باشد، زیرا سرور پستی ارسال کننده، فایل را به درون سرور دریافت کننده تزریق می‌کند. به‌ویژه آنکه ارتباط TCP از طریق نرم‌افزار ارسال کننده ایجاد می‌شود.

تفاوت دوم که قبلاً نیز به آن اشاره شد، آن است که پروتکل SMTP هر پیام را که شامل بدنه‌ی آن نیز می‌شود مجبور می‌کند که در قالب کُدهای اسکی 7 بیت تشکیل شوند. اگر پیام شامل حروفی باشد که به‌صورت اسکی 7 بیت نیستند، مانند حروف فرانسوی همراه با لهجه و یا شامل داده‌های باینری مانند فایل‌های تصویری آنگاه پیام بایستی در قالب کُدهای اسکی 7 بیت رمزگذاری گردد. پروتکل HTTP چنین محدودیتی را تحمیل نمی‌کند. تفاوت سوم و در عین حال مهم بین این دو پروتکل آن است که تعامل با یک فایل حامل اطلاعات تصویری و متن چگونه صورت می‌پذیرد؟ در بخش ۲- ۶ آموختیم که پروتکل HTTP هر عنصر را در درون پیام پاسخ خود کپسول‌بندی می‌کند در حالیکه پست الکترونیکی اینترنت و پروتکل SMTP همگی اشیاء یک پیام را در درون آن به‌صورت یک مجموعه قرار می‌دهد. به عبارت دیگر آنها بخشی از پیام می‌شوند.

قالب‌بندی پیام‌های الکترونیکی

هنگامی‌که A نام‌های معمولی می‌نویسد و می‌خواهد از طریق شبکه‌ی پست تحویل B نماید، او ممکن است همه نوع اطلاعات جانبی را در سرآغاز نامه‌ی خود قرار دهد. این اطلاعات می‌تواند شامل مواردی مانند آدرس B، آدرس بازگشت A و تاریخ و غیره باشد. به گونه‌ی مشابه هنگامی‌که یک پیام پست الکترونیکی از فردی به فرد دیگر ارسال می‌گردد، یک سرپیام آغازین ممکن است قبل از بدنه‌ی پیام قرار گیرد. این اطلاعات در مجموعه‌ای از سرپیام‌ها که به‌وسیله‌ی سند RFC5322 تعریف شده‌اند

1- Pull Protocol
2- Push Protocol

ممکن است در سرانداز هر پیام قرار داده شود. سرپیام‌ها به همراه بدنه‌ی پیام از طریق یک سطر تهی (یعنی CRLF) در پی یکدیگر قرار می‌گیرند. در مستند شماره 5322 دقیقاً قالب‌بندی سطر عناوین و نیز تفسیر مفهومی آنها مشخص شده‌اند. مانند پروتکل HTTP هر خط سرپیام شامل یک متن متعارف است که واژگان کلیدی به دنبال یک " : " و یک ارزش در پی یکدیگر قرار گرفته‌اند. بعضی از واژگان کلیدی اجباری و برخی اختیاری می‌باشند. هر سرپیام بایستی دارای یک From: و یک To: باشد و یا ممکن است به‌صورت اختیاری سرانداز Subject: و نظایر آن را نیز شامل گردد. نکته‌ی مهم تفاوت آنها با فرمان‌های SMTP است که قبلاً مطالعه گردید. اگرچه آنها شامل برخی واژه‌های مشترک مانند From و To می‌باشند. فرمان‌ها در بخش قبل (SMTP) بخشی از پروتکل دست‌دهی SMTP به‌شمار می‌آمدند در صورتی‌که سرپیام‌ها جزیی از بدنه‌ی پیام می‌باشند. سرانداز یک پیام ممکن است به‌صورت ذیل باشد:

```
From: A@crepes.fr
To: B@hamburger.edu
Subject: Searching for the meaning of life.
```

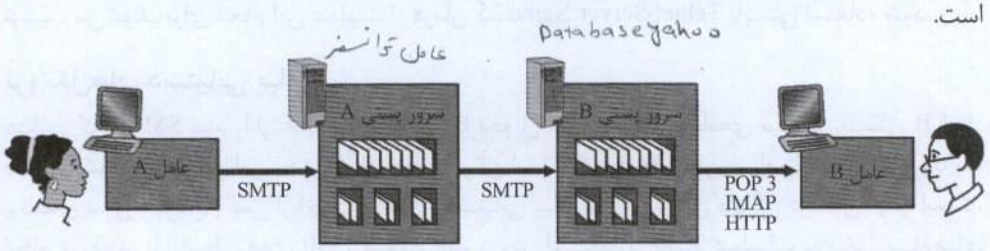
پس از سرپیام، یک خط تهی و سپس بدنه‌ی پیام در قالب اسکی قرار می‌گیرد. مجدداً خوانندگان به استفاده از برنامه‌ی Telnet برای ارسال یک پیام به همراه تعدادی سرپیام شامل Subject: و غیره ترغیب می‌شوند. برای انجام این عملیات از فرمان Telnet Server Name 25 بایستی استفاده شود.

پروتکل‌های دستیابی پیام

هنگامی‌که SMTP پیام را از A به سرور پستی B تحویل نمود. پیام در برنامه‌ی صندوق پستی B قرار داده می‌شود. در سراسر این بخش فرض مطلق بر آن است که B خواندن پیام را از طریق ورود به برنامه‌ی سرور میزبان و پس از اجرای برنامه‌ی دستیابی پیام که بر روی آن میزبان در حال اجرا است، انجام می‌دهد. تا اواسط دهه‌ی 1990 میلادی این روش استاندارد انجام عملیات بازیابی پیام‌های الکترونیکی بود. اما امروز دستیابی به پیام با استفاده از یک معماری مشتری- سرور صورت می‌پذیرد. یک کاربر پیام الکترونیکی خود را از طریق یک برنامه‌ی مشتری در حال اجرا در سیستم انتهایی خود می‌خواند. به‌عنوان مثال سیستم انتهایی می‌تواند یک PC و یا یک کامپیوتر همراه باشد. از طریق اجرای یک برنامه‌ی مشتری بر روی میزبان محلی کاربران می‌توانند از مجموعه‌ای از امکانات بهره گیرند که شامل پیام‌های چند رسانه‌ای و یا پیوست‌های متنوع می‌باشند.

با فرض آنکه B (دریافت کننده) عامل کاربری خود را بر روی میزبان محلی به حالت اجرا درآورد، طبیعی است که امکان قرار دادن یک سرور پیام را نیز در نظر داشته باشد. با این رویکرد برنامه‌ی سرور A قادر می‌شود مستقیماً با برنامه‌ی میزبان B محاوره نماید. اما یک چالش در این رویکرد وجود دارد. قبلاً اشاره گردید برنامه‌ی سرور پیام صندوق‌های پستی را مدیریت و برنامه‌های مشتری- سرور را در دو سوی SMTP اجرا می‌کند. اگر قرار باشد سرور پستی B بر روی میزبان محلی او سُکنی گزینند آنگاه کامپیوتر B همواره بایستی روشن و متصل به اینترنت باشد، تا قادر باشد پیام‌های جدید را که هر لحظه احتمال ورود آنها می‌رود، دریافت نماید. قبول این موضوع برای بسیاری از کاربران غیر عملی و دشوار می‌باشد. در صورتی‌که در عمل عامل کاربری بر روی میزبان محلی به اجرا در می‌آید و دستیابی به صندوق پستی یک کاربر با استفاده از یک برنامه‌ی سرور پست اشتراکی همیشه روشن انجام می‌گیرد. این سرور از سوی دیگر کاربران نیز دستیابی و معمولاً از طریق ISP کاربران نگهداری و مدیریت می‌شود. نمونه‌هایی از آن شرکت‌ها و دانشگاه‌ها می‌باشند.

اکنون مسیری که یک پیام الکترونیکی به هنگام ارسال از A به B طی می‌کند، بررسی می‌شود. قبلاً اشاره گردید که در نقطه‌ای از مسیر، پیام نیازمند ذخیره‌سازی بر روی سرور پستی B می‌باشد. این مهم می‌تواند به سادگی از طریق اجبار عامل کاربر به ارسال مستقیم پیام به سرور B محقق شود. پروتکل SMTP این توان‌مندی را در درون خود دارد. زیرا که طراحی این پروتکل بر مبنای تزریق پیام از یک میزبان به دیگری است. اما معمولاً عامل کاربری وارد تعامل مستقیم با سرور دریافت کننده نمی‌شود. در عوض همانگونه که شکل ۲- ۱۸ نشان می‌دهد، برنامه‌ی کاربر A با استفاده از SMTP پیام الکترونیکی او را به درون برنامه‌ی سرور پست A تزریق و سپس از طریق آن با استفاده از SMTP و به‌عنوان یک برنامه‌ی مشتری پیام را به سرور B تحویل می‌دهد. اکنون این پرسش مطرح می‌شود که دو مرحله‌ی اجرایی اشاره شده برای چیست؟ پاسخ می‌تواند به‌صورت ذیل باشد: نخست بدون عبور پیام A از طریق سرور آن عامل کاربری A منابع لازم جهت رسیدن به سرور پست در مقصد نامعلوم را ندارد و در نتیجه از این طریق A نخست پیام خویش را بر روی سرور متعلق به خود ذخیره و سپس سرور وی از طریق تلاشهای پی در پی مثلاً هر 30 دقیقه یکبار می‌تواند پیام را به سرور B تحویل نماید، تا آن هنگام که سرور B فعال شود. اگر سرور A دچار اشکال گردد آنگاه او بهانه‌ی کافی جهت شکایت از مدیر سیستم را در اختیار دارد. مستند RFC مربوط به پروتکل SMTP چگونگی استفاده از فرمان‌های SMTP جهت انتقال پیام از طریق چندین سرور SMTP را به طور کامل و گسترده تعریف نموده است.



شکل ۲- ۱۸ ♦ پروتکل‌های پیام الکترونیکی و عوامل ارتباطی آنها

دوم آنکه هنوز این معما وجود دارد، چگونه یک دریافت کننده مثل B که یک کاربر عامل بر روی میزبان محلی او در حال اجرا می‌باشد، پیام متعلق به خود را از سرور پست ISP دریافت می‌کند؟ یادآوری این نکته ضروری است که کاربر عامل B نمی‌تواند از SMTP برای بدست آوردن پیام خود استفاده نماید زیرا اخذ پیام یک عملیات استخراج است در حالیکه SMTP یک پروتکل تزریق است. راهکار این معما از طریق معرفی یک پروتکل ویژه‌ی دستیابی پیام میسّر خواهد بود. این پروتکل ویژه پیام‌ها را از سرور B به میزبان محلی او منتقل می‌کند.

در حال حاضر چندین پروتکل پُرآوازه دستیابی، شامل **پروتکل پست اداری نسخه‌ی 3**[1] (POP3)، **پروتکل دستیابی پیام اینترنت**[2] (IMAP) و البته HTTP وجود دارند. پروتکل SMTP براساس شکل ۲- ۱۸ انتقال پیام از کاربر عامل ارسال کننده را به سرور فرستنده انجام می‌دهد. یک پروتکل دستیابی پیام مانند POP3 از طرف دیگر انتقال پیام از برنامه‌ی سرور پست دریافت کننده را به کاربر عامل او انجام می‌دهد.

1- Post Office Protocol V3
2- Internet Mail Access Protocol

• پروتکل پست اداری (POP3)

پروتکل POP3 یک راهکار ساده‌ی دستیابی پیام است که در سند RFC1939 تعریف شده است از آنجائیکه این پروتکل بسیار ساده است عملکرد آن نیز دارای محدودیت هایی می‌باشد. آغاز فعالیت POP3 هنگامی است که کاربر عامل (مشتری) یک ارتباط TCP به سرور را بر روی پورت شماره‌ی 110 بازگشایی می‌کند. با برقراری ارتباط TCP این پروتکل از سه مرحله عبور می‌کند: (۱) **دریافت اختیارات**[1]، (۲) **مبادله**[2] و (۳) **به روز رسانی**. در مرحله‌ی نخست که دریافت اختیارات نام دارد کاربر عامل یک پنجره خالی شناسه‌ی کاربری را به همراه رمز عبور برای اعتبارسنجی کاربر ارسال می‌کند. در مرحله دوم تبادل صورت می‌گیرد به این مفهوم که کاربر عامل پیام‌ها را بازیابی می‌کند، همچنین در این مرحله مشتری پیام‌ها را جهت حذف نشان‌گذاری می‌کند و آمار پیام‌ها را بدست می‌آورد و یا نشان‌های حذف پیام را خارج می‌کند. در مرحله سوم به روز رسانی انجام می‌شود. پس از آنکه مشتری فرمان خروج (quit) را صادر می‌کند که مفهوم آن خاتمه‌ی POP3 است، سرور پیام‌های نشان گذاری شده را حذف می‌کند. در یک مبادله‌ی POP3 کاربر عامل فرمان‌ها را صادر و سرور به هر فرمان از طریق پاسخگویی واکنش نشان می‌دهد. دو واکنش معمولاً از سوی سرور روی می‌دهد: نخست واکنش +OK و به دنبال آن داده‌های سرور به مشتری است که سرور از آنها برای تأیید وضعیت پیام‌های قبلی استفاده می‌کند و دوم واکنش ERR- است که سرور آن را برای نشان دادن وجود خطا در پیام‌های قبلی به کار می‌برد. مرحله‌ی اختیارگیری نیز دو فرمان دارد: این دو فرمان عبارتند از:

User <username> و Pass <password> برای درک بهتر این دو فرمان بار دیگر می‌توان از برنامه‌ی Telnet استفاده نمود. با بکارگیری پورت شماره‌ی 110 می‌توان وارد سرور POP3 شد و دو فرمان عنوان شده را وارد نمود. اگر فرض شود mailServer نام سرور شما باشد آنگاه آنچه مشاهده خواهید نمود به‌صورت ذیل است:

```
telnet mailServer 110
+Ok POP3 Server ready
user B
+Ok
Pass hungry
+Ok user successfully logged on
```

اگر یک فرمان را اشتباه وارد نمائید، سرور POP3 با یک پیام ERR- واکنش نشان می‌دهد. اکنون فرآیند مبادله بررسی می‌شود. یک عامل کاربر که از POP3 استفاده می‌کند اغلب می‌تواند به‌وسیله‌ی کاربر پیکربندی شود تا از این طریق **دریافت و حذف**[3] و یا **دریافت و نگهداری**[4] را انجام دهد. زنجیره‌ی فرمان‌های صادر شده از طریق یک کاربر عامل POP3 بستگی دارد به اینکه کدامیک از دو نوع روش را کاربر عامل اجرا می‌کند. در روش دریافت و حذف کاربر عامل فرمان‌های list، retr و dele را وارد نماید. به‌عنوان مثال فرض کنید کاربری دارای دو پیام در صندوق پستی خود می‌باشد. در محاوره‌ای که در ذیل آمده است C: کاربر عامل و S: سرور پست است. بنابراین مبادله‌ی صورت گرفته شبیه به حالت ذیل است:

1- Authorization
2- Transaction
3- Download and delete
4- Download and Keep

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ....
S: ................
S: ......... blah)
S: .
C: dele 1
C: retr 2
S: (blah  blah ...
S: ............
S: ........ blah)
S: .
C: dele 2
C: quit
S:+Ok POP3 server signing off
```

توضیحات مثال بالا:

کاربر عامل ابتدا از سرور می‌خواهد طول هر پیام ذخیره شده را مشخص کند. مشتری سپس هر پیام را از سرور بازیابی و حذف می‌کند. بدیهی است بعد از هر مرحله‌ی دریافت اختیارات، کاربر عامل فقط از چهار فرمان استفاده می‌کند. این فرمان‌ها list، retr ، dele و quit هستند. قاعده‌ی تعریف این فرمان‌ها در RFC1939 تعریف شده است. پس از پردازش فرمان quit، سرور POP3 وارد مرحله سوم یعنی به‌روزرسانی می‌شود و پیام‌های 1 و 2 را از صندوق پستی حذف می‌کند. اشکالی که در این روش دریافت و حذف وجود دارد آن است که دریافت کننده (B) ممکن است سیار باشد. به عبارت دیگر از طریق چندین پردازشگر مانند کامپیوتر شخصی، کامپیوتر همراه، گوشی همراه و علاوه بر این موارد از خانه یا محل کار پیام‌های الکترونیکی خود را دستیابی نماید. در نتیجه روش دریافت و حذف، پیام‌های B را بر روی این پردازشگرهای چندگانه **تقسیم‌بندی**[1] (دیوارکشی) می‌کند. به‌ویژه آنکه اگر B نخست پیامی را از طریق کامپیوتر محل کار خود بخواند، آنگاه این امکان وجود ندارد که او مجدداً این پیام را از کامپیوتر منزل خود در زمان دیگری بخواند. از سوی دیگر در روش دریافت و نگهداری B می‌تواند پیام‌های خود را از طریق ماشین‌های مختلف به دفعات و از مکان‌های مختلف بازیابی نماید. در طول یک تعامل POP3 بین مشتری و سرور برخی از وضعیت‌های اطلاعاتی به‌ویژه رهگیری پیام‌های نشان‌گذاری شده برای حذف از سوی سرور POP3 نگهداری می‌شود. در هر حال سرور POP3 به حمل اینگونه اطلاعات در یک تعامل در حال انجام اقدام نمی‌کند. عدم بهره‌گیری از این امکانات سبب پیاده‌سازی ساده‌ی سرور POP3 می‌شود.

پروتکل دستیابی پیام اینترنت[2] (IMAP)

در پروتکل دستیابی POP3 ملاحظه شد هنگامی‌که B پیام خود را به میزبان محلی انتقال می‌نمود

1- Partition
2- Internet mail access protocol

می‌توانست به تعریف پوشه‌های پیام و جابجا نمودن پیام‌ها به این پوشه‌ها مبادرت نماید. پس همچنین می‌توانست پیام‌ها را حذف یا در بین پوشه‌ها جابجا نماید و یا آنکه به جستجوی آنها براساس نام فرستنده و یا موضوع بپردازد. اما این الگو یعنی قرار گرفتن پوشه‌ها و پیام‌ها در میزبان محلی برای یک کاربر کوچ گر که ترجیح می‌دهد یک پوشه‌ی سلسله مراتبی را بر روی سرور راه دور نگهداری نماید و از هر نوع کامپیوتری به دستیابی اطلاعات آن بپردازد، دشواری قابل توجهی را پدید می‌آورد. پروتکل POP3 توانایی ایجاد پوشه‌ها و انتقال پیام‌ها به آنها را ندارد. راهکار این چالش و نظایر آن پروتکل IMAP است که در سند RFC3501 تعریف و طراحی شده است. مشابه POP3، پروتکل IMAP به دستیابی پیام می‌پردازد. این پروتکل حامل تعداد بیشتری از ویژگی‌ها نسبت به POP3 می‌باشد، اما پیچیدگی چشمگیری را نیز شامل می‌شود. یک سرور IMAP هر پیام را وابسته به یک پوشه می‌کند. هنگامی‌که یک پیام نخست وارد سرور می‌شود به پوشه‌ی INBOX دریافت کننده وابسته می‌شود. گیرنده آنگاه پیام را به درون پوشه‌ای که به تازگی کاربر ایجاد کرده است هدایت می‌کند، پیام را می‌خواند و یا آن را حذف می‌کند. پروتکل IMAP حامل فرمان‌هایی است که به کاربران اجازه می‌دهند، پوشه تعریف کنند و یا اطلاعات را مابین پوشه‌ها جابجا کنند. علاوه بر این موارد IMAP فرمان‌های ویژه‌ی جستجوی پوشه‌های غیر محلی برای پیام‌های قابل نگاشت با ویژگی‌های مشخص را نیز فراهم می‌آورد. البته برخلاف POP3 یک سرور IMAP موقعیت اطلاعات کاربر را نیز در طول تعامل IMAP نگهداری می‌کند. به‌عنوان مثال اسامی پوشه‌ها و پیام‌های وابسته به آنها را. ویژگی مهم دیگر IMAP بهره‌گیری از فرمان‌هایی است که یک کاربر عامل (برنامه مشتری) را قادر می‌سازد عناصر و اجزای پیام‌ها را نیز بدست آورد. به‌عنوان مثال یک برنامه‌ی مشتری قادر است تنها سرپیامی از یک پیام و یا بخشی از یک **پیام چند بخشی**[1] (MIME) را بدست آورد. این ویژگی از آن جهت مفید است که در هنگام روبه‌رویی با یک ارتباط کند از لحاظ سرعت (مانند پیوندهای ارتباطی مودم‌ها) بین کاربر عامل و سرور آن، همه‌ی پیام‌ها بنا بر نیاز کاربر به درون صندوق پستی انتقال داده نمی‌شود. به‌ویژه آنکه پیام‌های طولانی که احتمالاً دارای بخش‌های صدا و تصویر می‌باشند، بنا بر اراده‌ی کاربر دریافت نمی‌شود.

پست الکترونیکی بر پایه‌ی وب

در حال حاضر تعداد بی‌شماری از کاربران ارسال و دریافت پیام‌های الکترونیکی را از طریق مرورگرهای وب انجام می‌دهند. در اواسط دهه‌ی 1990 میلادی نخستین خدمات دستیابی از سوی هات میل (hot mail) بر مبنای دستیابی از طریق وب عرضه گردید. اکنون این خدمات از سوی سامانه هایی مانند گوگل، یاهو، هر یک از دانشگاه‌های مهم و معتبر و نیز شرکتهای تجاری و تولیدی بزرگ بر پایه دستیابی از وب عرضه می‌شوند. با این خدمات کاربر عامل یک مرورگر وب معمولی است و کاربر با صندوق پستی راه دور خود از طریق پروتکل HTTP ارتباط برقرار می‌نماید. هنگامی‌که دریافت کننده‌ای مانند B می‌خواهد پیامی را از صندوق پستی خویش بازیابی نماید، پیام مورد نظر از سرور پست B به مرورگر B با استفاده از HTTP به جای پروتکل‌های POP3 و یا IMAP منتقل می‌شود. هنگامی‌که کاربری مانند A مایل به گسیل یک پیام الکترونیکی می‌باشد، پیام مورد نظر از طریق مرورگر او و به کمک HTTP به سرور پست وی به جای پروتکل SMTP منتقل می‌شود. سرور A در هر حال هنوز ارسال و دریافت پیام‌ها را با استفاده از پروتکل SMTP انجام می‌دهد.

1- Multi purpose internet mail extension

۲-۱۳ خدمات دفترچه راهنمای اینترنت (DNS)

روش‌های بی‌شماری وجود دارند که از طریق آنها می‌توان به شناسایی هویت موجودیتی به نام انسان پی برد. به‌عنوان مثال همگی ما از طریق نام درون شناسنامه، شماره‌ی کارت ملی، شماره و مشخصات گواهینامه‌ی رانندگی و غیره می‌توانیم مورد شناسایی قرار بگیریم. اگرچه از همه‌ی این شناسه‌ها می‌توان جهت تعیین هویت افراد استفاده نمود اما در یک چارچوب معین یکی از این شناسه‌ها ممکن است مناسبتر از دیگران باشد. به‌عنوان مثال در ایالات متحده کامپیوترهای سازمان امور مالیاتی آمریکا (IRS) ترجیح می‌دهند از شماره‌های تأمین اجتماعی افراد که از لحاظ تعداد ارقام ثابت می‌باشند جهت شناسایی استفاده نمایند. از طرف دیگر افراد معمولی در این کشور جهت تعامل روزانه‌ی خود ترجیح می‌دهند از نام‌های درون شناسنامه‌ی خود که یادآوری آنها ساده تر است استفاده نمایند. علت این موضوع نیز ساده می‌باشد به‌عنوان مثال صحنه‌ای را تجسّم کنید که بانویی برای معرفی خود بخواهد بگوید: نام من 9875 - 67 – 132 می‌باشد و این آقا همسر من 1146- 87 – 178 است. درست همانگونه که هویت انسان‌ها از طریق روش‌های متعددی قابل شناسایی است. یک شناسه برای یک میزبان نیز نام **میزبان**[1] می‌باشد. نام‌های میزبانی مانند: Yahoo.com، cnn.com، cis.poly.edu، hrh.ir همگی از جمله نام‌هایی هستند که یادآوری آنها برای افراد بسیار ساده و آسان است. اما نام‌های میزبانی اطلاعات اندکی را از موقعیت یک میزبان در درون اینترنت به ما نشان می‌دهند. یک نام میزبان مانند hrh.ir با پسوند ir. به ما می‌گوید این میزبان احتمالاً در ایران است اما اطلاعات بیشتری را در اختیار ما نمی‌گذارد. علاوه بر آن از آنجائیکه نام‌های میزبان می‌توانند شامل طول متغیری از لحاظ ترکیب حرف و عدد باشند، شناسایی آنها به‌وسیله‌ی مسیریاب‌ها از لحاظ پردازش بسیار دشوار خواهد شد. به این سبب است که معمولاً میزبان‌ها از طریق آدرس و با نامی شناسایی می‌شوند که اصطلاحاً آدرس IP[2] نامیده می‌شود. بررسی این آدرس‌ها مربوط به فصل ۴ می‌باشد اما سودمند خواهد بود اگر در این مقطع به اختصار به توصیف آنها بپردازیم. یک آدرس IP معمولاً شامل چهار بایت است و بین هر بایت نیز یک نقطه (.) وجود دارد که بایت‌ها را از یکدیگر جدا می‌سازد. آنچه حائز اهمیت است ساختار سلسله مراتبی این آدرس‌ها می‌باشد. یک آدرس IP می‌تواند شبیه به: 121.7.106.83 باشد. در این آدرس هر یک از بایت‌ها محدوده‌ای بین عدد صفر تا 255 را می‌توانند اختیار نمایند. یک آدرس IP را از آن جهت سلسله مراتبی می‌خوانند زیرا که آنرا می‌توان از چپ به راست پویش نمود. با این عمل اطلاعات بیشتر و مشخصی در مورد موقعیت میزبان در اینترنت و به عبارت دیگر شبکه‌ای که در آن قرار دارد و موارد دیگر بدست می‌آید. به گونه‌ی مشابه هنگامی‌که پویش یک آدرس پستی را از فرود به فراز آن انجام می‌دهیم، اطلاعات بیشتر و مشخصی را از موقعیت مکانی فردی که در آدرس از او نام برده شده است بدست می‌آوریم.

- **خدمات DNS**

ملاحظه شد که از دو طریق می‌توان به هویت یک میزبان پی برد. به‌وسیله‌ی نام میزبان و یا آدرس IP. ما به‌عنوان موجودات انسانی استفاده از نام میزبان را در تعامل خود با شبکه‌ی اینترنت ترجیح می‌دهیم در حالیکه مسیریاب‌ها آدرس‌های IP با ساختار سلسله مراتبی و طول ثابت را ترجیح می‌دهند. بنابراین جهت راضی نمودن دو طرف تعامل از یک دفترچه‌ی راهنما جهت تفسیر نام میزبان‌ها به آدرس‌ها

1- Host names
2- IP addresses

استفاده می‌شود. پیاده‌سازی این مهم در اینترنت بر عهده‌ی **سیستم قلمرو**[1] (DNS) می‌باشد. این سیستم در حقیقت (۱) یک پایگاه داده‌ی توزیع شده در سرورهای DNS به‌صورت سلسله مراتبی است و (۲) یک پروتکل لایه‌ی کاربرد است که اجازه می‌دهد میزبان‌ها به جستجوی پایگاه داده‌ی اشاره شده بپردازند. سرورهای DNS اغلب ماشین‌های از نوع UNIX می‌باشند که نرم‌افزار ویژه‌ای به نام **قلمرو نام اینترنتی برکلی**[2] (BIND) را اجرا می‌کنند. پروتکل DNS از طریق پروتکل UDP با استفاده از پورت شماره‌ی 53 تعامل خود با سایر لایه‌ها را انجام می‌دهد. در حال حاضر DNS به‌وسیله پروتکل‌های دیگری در لایه‌ی کاربرد نیز مورد استفاده قرار می‌گیرد. این پروتکل‌ها عبارتند از: HTTP، SMTP و FTP. به‌عنوان مثال هنگامی‌که یک مرورگر مانند یک مشتری HTTP در حال اجرا بر روی یک میزبان است و درخواست یک آدرس URL مانند: www.Someschool.edu/index.html را می‌نماید. برای آنکه میزبان کاربر قادر به ارسال یک پیام درخواست HTTP به یک وب سرور مانند: www.Someschool.edu باشد، میزبان کاربر نخست لازم است آدرس IP مربوط به این نام میزبان را بدست آورد. چگونگی انجام این مهم به گونه‌ی ذیل است:

۱- برنامه کاربردی مشابهی در سمت برنامه‌ی مشتری کاربرد DNS به اجرا در می‌آید.

۲- مرورگر نام میزبان www.someschool.edu را از URL استخراج و آن را به کاربرد DNS سمت مشتری گسیل می‌دارد.

۳- مشتری DNS درخواست جستجویی را که شامل نام میزبان است به سرور DNS ارسال می‌کند.

۴- مشتری DNS سرانجام پاسخی را دریافت می‌کند که شامل آدرس IP برای نام میزبان می‌باشد.

۵- هنگامی‌که مرورگر آدرس IP را از DNS دریافت می‌کند، مبادرت به ایجاد یک ارتباط TCP با پردازش سرور HTTP بر روی پورت شماره‌ی 80 در آن آدرس IP می‌کند.

بنابراین ملاحظه می‌شود که DNS سبب افزودن یک تأخیر اضافی می‌شود که بعضی مواقع چشمگیر نیز می‌باشد. این تأخیر به یک کاربرد اینترنت که از آن استفاده می‌کند، تحمیل می‌شود. البته این تأخیر چندان نگران کننده نیست زیرا همانگونه که در ادامه‌ی بحث مشخص خواهد شد، آدرس IP مورد نظر در اغلب مواقع در یک سرور DNS در نزدیکی محل درخواست نهان شده است. این خود سبب کاهش ترافیک DNS شبکه و میانگین تأخیر آن خواهد شد. علاوه بر موارد بالا DNS چندین سرویس مهم دیگر را نیز فراهم می‌آورد. از جمله عبارتند از:

- **نام‌های مستعار میزبان**[3]

یک میزبانِ دارای یک نامِ پیچیده، می‌تواند از یک یا چندین نام مستعار استفاده نماید. به‌عنوان مثال یک نام میزبان مانند flower1.red-yellow.enterprise.com می‌تواند دارای دو نام مستعار مانند enterprise.com و www.enterprise.com باشد. در این حالت گفته می‌شود flower1.red-yellow.enterprise.com یک نام میزبان متعارف به‌شمار می‌آید. حضور نام‌های میزبان مستعار از لحاظ یادآوری در ذهن ساده‌تر و آسانتر از نام‌های متعارف می‌باشند. یک برنامه‌ی کاربردی می‌تواند با احضار DNS از طریق یک نام مستعار به دستیابی نام متعارف میزبان و نیز آدرس IP آن بپردازد.

server یا service

1- Domain name systems

2- Brekeley Internet Name Domain

3- Host aliasing

- **نام‌های مستعار برنامه‌ی سرور پست**

بنا بر دلایل روشن استفاده از آدرس‌های پستی ساده و آسان جهت نگهداری در حافظه‌ی انسان لازم و ضروری می‌باشد. به‌عنوان مثال اگر شخص B دارای یک حساب کاربری با گوگل می‌باشد، آدرس پستی او می‌تواند به شکل ساده‌ی خود در قالب B@ google.com تعریف شود. اما نام میزبان سرور پستی گوگل پیچیده‌تر و در نتیجه به خاطر سپردن آن در ذهن دشوارتر است. به‌عنوان مثال نام متعارف میزبان گوگل ممکن است به‌صورت flower1.red-yellow.google.com باشد. یک برنامه‌ی کاربردی پست الکترونیکی می‌تواند با احضار DNS، نام متعارف میزبان و نیز آدرس IP آن را با استفاده از نام مستعار آن بدست آورد. در حقیقت برنامه‌های مانند نرم‌افزار ثبت MX که در ادامه توضیح داده می‌شود این امکان را ایجاد می‌کند که هر دو سرورهای پستی و وب متعلق به یک شرکت بتوانند نام‌های میزبان مستعار مشابه هم اختیار کنند. به‌عنوان مثال هر دو سرور پُستی و وب یک شرکت می‌توانند enterprise.com نامیده شوند.

توزیع بار ترافیک در DNS

از پروتکل DNS برای توزیع بار بین سرورهای تکراری مانند وب سرورها نیز استفاده می‌شود. سامانه‌های پُرکار مانند cnn.com بر روی چندین سرور تکرار شده‌اند و هر یک از آنها نیز بر روی سیستم‌های انتهایی متفاوت در حال اجرا می‌باشند. در نتیجه دارای آدرس‌های IP گوناگون نیز هستند. برای وب سرورهای تکراری مجموعه‌ای از آدرس‌های IP وابسته به یک نام متعارف و استاندارد تعریف می‌شود. هنگامی‌که مشتری به احضار DNS برای نگاشت یک نام به مجموعه‌ای از آدرس‌ها اقدام می‌کند، سرور با بازگشت همه‌ی آدرس‌های IP پاسخ مناسب را ارایه می‌دهد. از آنجائیکه یک مشتری معمولاً تقاضای HTTP خود را به آدرس IP ارسال می‌کند و آن آدرس نیز در ابتدای صورت مجموعه آدرس‌ها قرار دارد آنگاه برنامه‌ی DNS از طریق چرخش به توزیع ترافیک در بین سرورهای تکرار شده می‌پردازد. از چرخش DNS برای پُست الکترونیکی نیز استفاده می‌شود تا بتوان چندین سرور پُست که حامل نام‌های مستعار مشابه می‌باشند را دستیابی نمود. در حال حاضر شرکت‌های بزرگ اینترنتی که فعالیت آنها توزیع ظرفیت امکانات اینترنت است نیز از این پروتکل استفاده می‌کنند. در فصل ۷ فعالیت اینگونه نهادها بررسی شده و مفهوم توزیع ظرفیت تحلیل گردیده است. مشخصات DNS در مستند RFC1035 تعریف شده است. در ادامه‌ی مطالب اصول بنیادی عملکرد این سیستم پیچیده بررسی می‌شود.

چگونگی عملکرد DNS

این بخش بر روی موضوعات مربوط به عملکرد DNS، تأکید بر روی چگونگی خدمات تبدیل یک نام میزبان به آدرس IP متمرکز خواهد شد. اگر فرض شود یک برنامه‌ی کاربردی مانند مرورگر وب و یا یک سرور پُست الکترونیکی در سیستمی نیازمند تبدیل یک نام میزبان به یک آدرس IP می‌باشد. کاربرد اشاره شده به احضار برنامه‌ی مشتری سمت پروتکل DNS می‌پردازد و نام میزبان مورد نظر را که بایستی تفسیر نماید در اختیار آن می‌گذارد. در بسیاری از ماشین‌هایی که بر پایه‌ی UNIX عمل می‌کنند از تابع gethostbyname() جهت فراخوانی کاربرد مورد نظر برای انجام عملیات ترجمه استفاده می‌شود. در بخش‌های بعد نشان داده شده است که چگونه یک کاربر جاوا می‌تواند DNS را احضار نماید.

هنگامی‌که فراخوانی DNS صورت می‌گیرد این پروتکل از طریق میزبان کاربر پیام درخواست را

وارد شبکه می‌کند. همه‌ی تقاضاها و پاسخ‌های DNS از طریق داده گرام‌های UDP به پورت شماره 53 ارسال می‌شوند. پس از تأخیری که می‌تواند در محدوده‌ی چند میلی ثانیه تا چندین ثانیه به طول انجامد، پروتکل DNS ساکن در میزبان کاربر پیام پاسخی را از یک DNS که نگاشت مورد نظر نام و آدرس IP را انجام داده است، دریافت می‌کند. این نگاشت سپس به احضار کاربرد می‌پردازد. بنابراین از نقطه‌نظر فراخوانی کاربرد در میزبان کاربر، DNS همانند جعبه‌ی سیاهی عمل می‌کند که یک سرویس ترجمه‌ی ساده و شفافی را فراهم می‌آورد. اما در حقیقت جعبه‌ی سیاهی که این سرویس را پیاده‌سازی می‌کند بسیار پیچیده و شامل تعداد زیادی سرور DNS است که به همراه پروتکل لایه‌ی کاربرد که چگونگی ارتباط و تعامل بین میزبان‌های متقاضی و سرورهای DNS پاسخگو را تنظیم و کنترل می‌کنند در سراسر گیتی توزیع شده‌اند. یک طراحی ساده برای DNS می‌تواند از یک سرور DNS که شامل همه‌ی عوامل لازم برای نگاشت می‌باشد، تشکیل گردد. بدیهی است در این طراحی متمرکز برنامه‌ی مشتری می‌تواند به طرز ساده‌ای همه‌ی درخواست‌های خود را به سوی یک سرور DNS هدایت و آن سرور نیز مستقیماً به همه‌ی آنها پاسخگو می‌باشد. اگرچه سادگی این طراحی جذاب به نظر می‌رسد اما برای اینترنت امروز نامناسب است، زیرا که تعداد میزبان‌ها مرتباً در حال افزایش است. چالش‌های یک طراحی متمرکز عبارتند از:

۱- **مردودی یک نقطه**- اگر سرور DNS دچار خرابی گردد، تمامی بدنه‌ی اینترنت نیز به همراه آن متوقف می‌شود.

۲- **حجم ترافیک**- یک سرور DNS به تنهایی اگر لازم باشد همه‌ی تقاضاها را که از سوی صدها میلیون کاربر به‌صورت HTTP یا پست الکترونیکی صادر شده است ، پاسخگویی نماید آنگاه می‌توان دشواری‌های ترافیکی این موضوع را تصور نمود.

۳- **پایگاه داده‌ی متمرکز**- بدیهی است یک سرور DNS متمرکز به تنهایی نمی‌تواند از لحاظ فاصله به همه‌ی متقاضیان نزدیک باشد. اگر یک سرور DNS در شهر نیویورک مستقر باشد، آنگاه همه‌ی تقاضاهای ارسالی از یک نقطه مانند استرالیا بایستی به گوشه‌ی دیگر گیتی منتقل گردند که امکان عبور از پیوندهای متراکم از لحاظ ترافیک، و کند از نقطه‌نظر سرعت نیز دور از انتظار نمی‌باشد. این موضوع خود سبب تأخیر گسترده‌ای می‌شود.

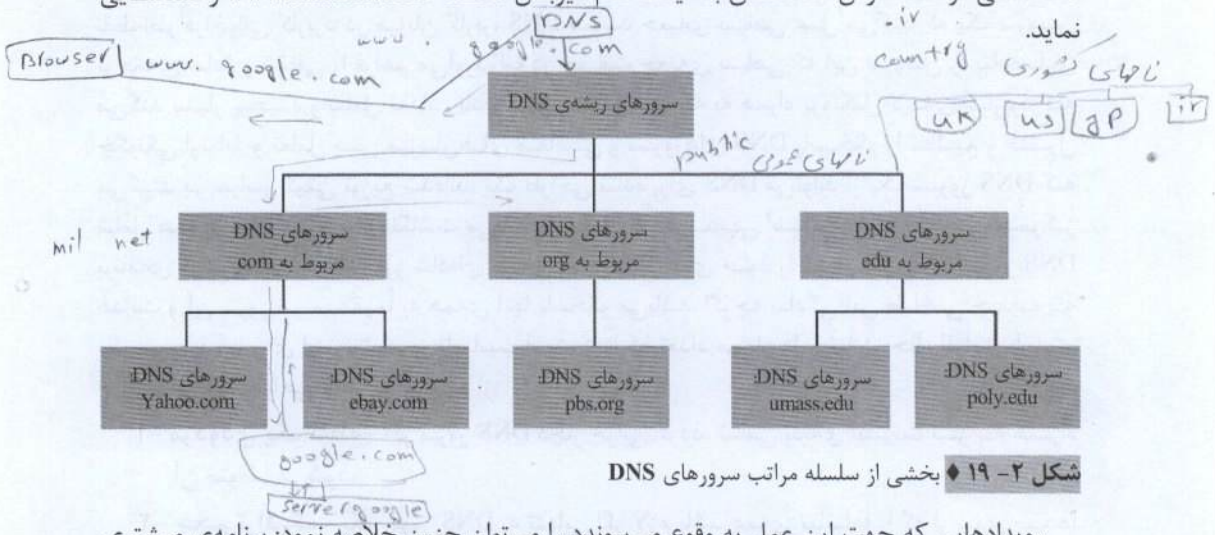
۴- **نگهداری**- یک سرور DNS تنها بایستی سوابق همه‌ی میزبان‌های اینترنت را در درون خود نگهداری نماید. این عمل نه تنها باعث ایجاد یک پایگاه داده‌ی متمرکز بسیار عظیم خواهد شد بلکه، مرتباً لازم است برای هر میزبان جدید پایگاه به‌روزرسانی شود.

به اختصار می‌توان این نتیجه‌گیری را نمود که یک پایگاه داده‌ی متمرکز فقط در یک سرور DNS تک مقیاس‌پذیر نمی‌باشد. بنابراین روش پیاده‌سازی DNS به‌صورت توزیع شده است و DNS را می‌توان نمونه‌ای جالب از چگونگی توزیع و پیاده‌سازی یک پایگاه داده در اینترنت به‌شمار آورد.

پایگاه داده‌ی سلسله مراتبی توزیع شده

مقیاس‌پذیری DNS که در بالا به آن اشاره گردید از طریق سازماندهی تعداد زیادی از سرورها با روش سلسله مراتبی و توزیع شده در سراسر گیتی امکان‌پذیر شده است. هیچ‌یک از سرورهای DNS دارای همه‌ی نگاشتها برای تمامی میزبان‌ها در اینترنت نمی‌باشند. در عمل نگاشت به گونه‌ی توزیع شده در بین همه‌ی سرورهای DNS انجام شده است. به طور کلی سه گروه برنامه‌ی سرور DNS وجود دارد.

(۱) سرورهای DNS **ریشه**[1]، (۲) سرورهای DNS محدوده‌ی **سطح بالا**[2] (TLD) و (۳) سرورهای DNS **مقتدر**[3]. سازماندهی این سه گروه سرور براساس شکل ۲- ۱۹ می‌باشد. برای آگاهی از چگونگی تعامل این سه گروه سرور با یکدیگر می‌توان به بررسی یک مثال پرداخت. فرض کنید یک مشتری DNS می‌خواهد آدرس IP متعلق به یک نام میزبان مانند: www.amazon.com را شناسایی نماید.



شکل ۲- ۱۹ ♦ بخشی از سلسله مراتب سرورهای **DNS**

رویدادهایی که جهت این عمل به وقوع می‌پیوندد را می‌توان چنین خلاصه نمود: برنامه‌ی مشتری نخست با یکی از سرورهای ریشه تماس برقرار می‌کند و آن سرور آدرس‌های IP مربوط به سرورهای TLD را برای محدوده‌ی سطح بالا (com.) در اختیار آن مشتری می‌گذارد. آنگاه او با یکی از این سرورهای com. ارتباط برقرار می‌کند و آن سرور آدرس IP متعلق به سرور مختار (مقتدر) amazon.com را بازگشت می‌دهد. در نتیجه ترجمه‌ی نام میزبان www.amazon.com به آدرس IP آن انجام می‌پذیرد. در ادامه‌ی مطلب فرایند جستجو با جزئیات بیشتری بررسی خواهد شد. اما قبل از آن بررسی بیشتری بر روی این سه گروه DNS انجام می‌گیرد.

۱- سرورهای DNS ریشه

در شبکه‌ی اینترنت تعداد 13 سرور DNS از نوع ریشه وجود دارند، آنها به گونه‌ی قراردادی از حروف لاتین A تا M نامگذاری شده‌اند. بسیاری از این سرورها در آمریکای شمالی مستقر می‌باشند. نقشه‌ی و چیدمان جغرافیایی این سرورهای ریشه در شکل ۲- ۲۰ نشان داده است. این شکل براساس آمار آخرین اطلاعات تا اکتبر 2012 میلادی می‌باشد. فهرستی از آخرین سرورها را می‌توان از طریق آدرس Root-Servers 2012 دستیابی نمود. اگرچه هر یک از این 13 سرور ریشه‌ی DNS به‌عنوان یک سرور تک به‌شمار می‌آیند اما هر یک در حقیقت خوشه‌ای از سرورهای تکراری را تشکیل داده‌اند علت آن هم به دلیل اهداف امنیتی و نیز افزایش قابلیت دسترسی است. به‌طور کلی تا پاییز 2011 تعداد سرورهای ریشه 247 فقره بوده است.

1- Root DNS
2- Top Level DNS
3- Authoriative

۲- سرورهای TLD

این سرورها جهت انجام پردازش‌های مورد نیاز در محدوده‌ی سطح بالای اینترنت یعنی com. ، org.، net.، edu. وgov. می‌باشند. علاوه بر این موارد همه‌ی محدوده‌های سطح بالای کشوری مانند UK، fr، jp و ir را نیز پوشش می‌دهند. یک شرکت تجاری با نام راهکارهای شبکه[1] (NS) عهده دار مسئولیت نگهداری و مدیریت همه‌ی سرورهای TLD مربوط به com. می‌باشد. مشابه آن شرکت Educause نیز همین مسئولیت را برای سرورهای TLD از نوع edu. انجام می‌دهد.

۳- سرورهای مقتدر (مختار)

هر نهادی که مالکیت میزبان‌های همگانی قابل دسترس را در اختیار دارد. موظف به نگهداری سوابق قابل دستیابی DNSها به گونه‌ای می‌باشد که قابلیت نگاشت اسامی این میزبان‌ها به آدرس‌های IP امکان‌پذیر باشد. سرور DNS مختار متعلق به یک نهاد همه‌ی این سوابق را در درون خود نگهداری می‌کند. یک نهاد می‌تواند به‌صورت مستقل سرور DNS مقتدر خود را گزینش و بر روی آن این سوابق را ذخیره و نگهداری نماید. البته در صورت لزوم یک نهاد می‌تواند با پرداخت هزینه‌ای این سوابق را بر روی یک سرور DNS مختار در یک ISP ذخیره و نگهداری نماید. اغلب دانشگاه‌ها و شرکت‌های بزرگ تجاری سرورهای DNS مقتدر اولیه و ثانوی (پشتیبانی) خود را پیاده‌سازی و نگهداری می‌کنند.



شکل ۲- ۲۰ ♦ مجموعه سرورهای DNS ریشه تا سال 2012 براساس نام، نهاد و موقعیت

هر سه نوع سرور DNS ریشه، TLD و مختار متعلق به سلسله مراتبی از سرورهای DNS براساس شکل ۲- ۱۹ می‌باشند. اما نوع مهم دیگری از این سرورها نیز وجود دارد که آن را سرور DNS محلی می‌نامند. یک DNS محلی به‌طور کلی وابستگی به سلسله مراتب سرورها ندارد اما در هر حال برای

1- Network Solutions

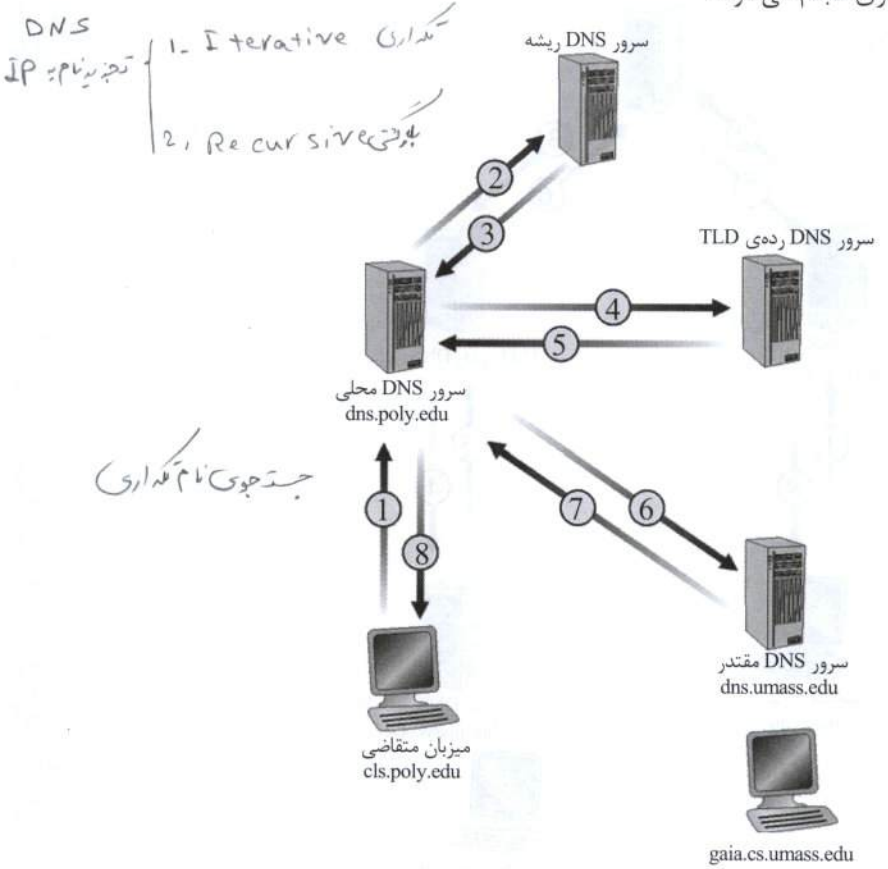
معماری DNS دارای نقش حیاتی است. هر ISP مانند یک دانشگاه، یک دانشکده، یک شرکت و یا یک ISP مسکونی دارای یک سرور DNS محلی است که آنرا **سرور نام پیش‌فرض**[1] نیز می‌خوانند. هنگامی‌که یک میزبان به یک ISP متصل می‌شود، آن ISP یک یا چند آدرس IP را از سرور محلی خود به میزبان اختصاص می‌دهد. عملیات اختصاص آدرس (ها) از طریق یک سرور به نام **پروتکل پویای پیکربندی میزبان**[2] (DHCP) که در فصل ۴ بررسی بیشتری روی آن صورت می‌گیرد، انجام می‌شود. از طریق دستیابی پنجره‌ی وضعیت شبکه در سیستم عامل ویندوز و یا UNIX به سادگی می‌توان آدرس IP سرور DNS محلی یک میزبان را بدست آورد. سرور DNS محلی یک میزبان معمولاً در نزدیکی و یا مجاورت میزبان قرار گرفته است. برای ISP یک نهاد مانند دانشگاه سرور DNS محلی ممکن است بر روی شبکه‌ی محلی LAN قرار داشته باشد که همان میزبان متصل به آن است. برای یک ISP مسکونی معمولاً سرور DNS محلی با فاصله‌ی چندین مسیریاب از یک میزبان واقع شده است. هنگامی‌که یک میزبان درخواستی را به یک DNS می‌دهد، آن تقاضا به سرور DNS محلی گسیل و این سرور به‌صورت جانشین، درخواست را به سرور DNS سلسله مراتب منتقل می‌کند. این مراحل عملیاتی در قالب یک مثال در ذیل توضیح داده شده است:

فرض کنید میزبان cls.poly.edu مایل به دانستن آدرس IP میزبان gaia.cs.umass.edu می‌باشد. همچنین اگر سرور DNS محلی این نهاد دارای نام dns.poly.edu و سرور مختار برای gaia.cs.umass.edu نیز با آدرس dns.umass.edu باشد. همانگونه که شکل ۲- ۲۱ نشان می‌دهد. میزبان cls.poly.edu ابتدا پیام درخواستی را به سرور DNS محلی dns.poly.edu ارسال می‌کند. این پیام شامل نام میزبان gaia.cs.umass.edu است که بایستی ترجمه شود. سرور محلی پیام را به DNS ریشه گسیل می‌کند. این سرور پسوند edu. را ملاحظه و صورت آدرس IP سرورهای TLD مسئول edu. را به DNS محلی بازگشت می‌دهد. سرور محلی مجدداً پیام درخواست را به یکی از این سرورهای TLD ابلاغ می‌کند و آن سرور با ملاحظه‌ی پسوند umass.edu آدرس IP سرور DNS مختار مربوط به نام میزبان dns.umass.edu را بازگشت می‌دهد. سرانجام سرور DNS محلی مجدداً درخواست خود را مستقیماً به dns.umass.edu ارسال می‌کند که این سرور پاسخ درخواست را با بازگشت آدرس IP متعلق به gaia.cs.umass .edu پاسخ می‌دهد. بدیهی است در این مثال برای نگاشت آدرس IP مربوط به میزان تعداد 8 پیام ارسال گردید. از این تعداد 4 پیام درخواست و 4 پیام پاسخ آنها بود. در ادامه ملاحظه می‌شود چگونه نهان‌سازی سبب کاهش ترافیک خواهد شد.

در مثال قبل فرض بر آن بود که سرور TLD برای نام میزبان مورد نظر سرور DNS مختار را می‌شناسد. این واقعیت در عمل همیشه وجود ندارد. آنچه سرور TLD ممکن است از آن آگاهی داشته باشد یک سرور DNS میانی یا واسط است که آن سرور ممکن است شناختی را از DNS مختار نام میزبان مورد نظر داشته باشد. به‌عنوان مثال، مجدداً فرض کنید که نهاد نام برده شده در بالا دارای یک سرور DNS برای دانشگاهی است که نام آن dns.umass.edu می‌باشد. همچنین فرض کنید هر دانشکده‌ای در دانشگاه ماساچوست سرور DNS متعلق به خود را داشته باشد. در این مقطع هنگامی‌که سرور میانی یعنی dns.umass.edu درخواستی را از یک نام میزبان که انتهای آن به‌صورت cs.umass.edu می‌باشد، دریافت نماید آن سرور آدرس IP مربوط به dns.cs.umass.edu که در حقیقت سرور مختار همه‌ی آدرس‌های میزبان‌هایی است که در انتهای خود عبارت cs.umass.edu را دارند،

1- Default name server
2- Dynamic Host Configuration Protocol

بازگشت می‌دهد. سرور DNS محلی آنگاه درخواست را به سرور مختار ارجاع می‌دهد و آن نیز نگاشت آدرس مطلوب را انجام داده و سرانجام آن آدرس را به DNS محلی بازگشت می‌دهد. در این مثال تعداد پیام‌های ارسال شده 10 می‌باشد! مثال شکل ۲-۲۱ از هر دو نوع نامبرده شده که اصطلاحاً **جستجوی بازگشتی**[1] و **جستجوی تکراری**[2] نامیده می‌شوند استفاده می‌نماید. درخواست از cls.poly.edu به dns.poly.edu ارسال می‌گردد. این یک درخواست بازگشتی است، زیرا درخواست مورد نظر از dns.poly.edu می‌خواهد که نگاشت را به وکالت از جانب آن انجام دهد. به دنبال آن سه درخواست بعدی همگی تکراری هستند زیرا همه‌ی پاسخ‌ها مستقیماً به dns.poly.edu بازگشت می‌کند. اساساً هر ارجاع به DNS می‌تواند تکراری یا بازگشتی باشد. به‌عنوان مثال شکل ۲- ۲۲ زنجیره‌ای از جستجوهای DNS را نشان می‌دهد که همگی آنها بازگشتی می‌باشند. در عمل ارجاع‌ها همگی الگوی شکل ۲- ۲۱ را دنبال می‌کنند. ارجاع از میزبان متقاضی به DNS محلی که نوع آن بازگشتی است منتقل و بقیه‌ی جستجوها به‌صورت تکراری انجام می‌شوند.

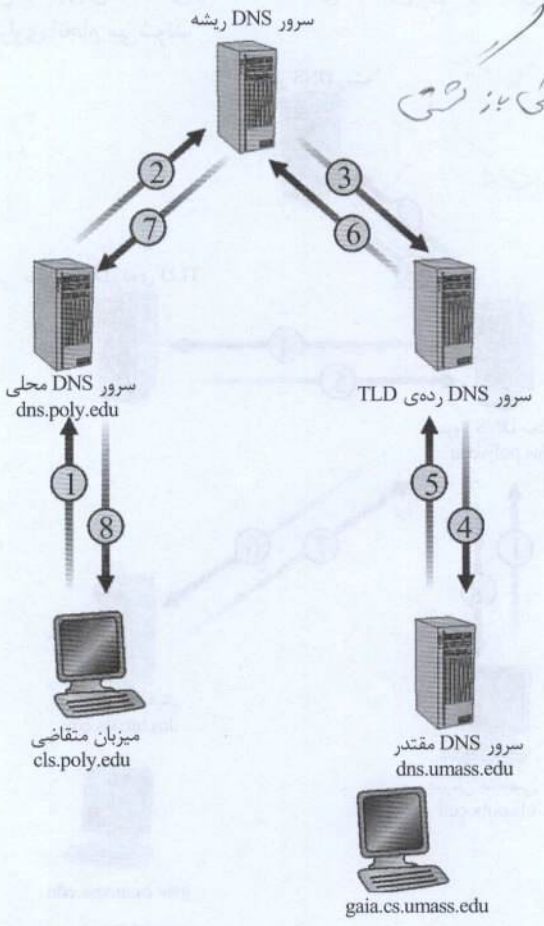


شکل ۲- ۲۱ ♦ تعامل سرورهای DNS گوناگون

1- Rrecursive queries
2- Iterative queries

• نهان سازی[1] DNS

تا به حال مواردی که بر روی آنها تحلیل انجام شد واقعیت نهان‌سازی DNS که یکی از حیاتی‌ترین ویژگی‌های سیستم DNS می‌باشد را در نظر نگرفته‌اند. آنچه در عمل اتفاق می‌افتد بهره‌گیری گسترده از نهان‌سازی DNS جهت افزایش کارایی و کاهش تعداد پیام‌های DNS است که در اطراف اینترنت گمانه زنی می‌کنند. آنچه در پس پرده‌ی نهان‌سازی DNS پنهان شده است، بسیار ساده می‌باشد. در یک زنجیره درخواست هنگامی‌که یک سرور DNS پاسخی را دریافت می‌کند که شامل نگاشتی از یک نام میزبان به یک آدرس IP است آن سرور می‌تواند این نگاشت را در حافظه‌ی محلی خود نهان کند. به‌عنوان مثال در شکل ۲- ۲۱ هر مرتبه سرور DNS محلی dns • poly • edu پاسخی را از یک DNS دریافت می‌کند، هرگونه اطلاعات درون پاسخ را می‌تواند نهان‌سازی نماید.



شکل ۲- ۲۲ ♦ ارجاع‌های بازگشتی

اگر جفت آدرس‌های میزبان و IP در یک سرور DNS نهان‌سازی شده باشند و ارجاع بعدی به سرور

1- DNS caching

DNS وارد گردد و مربوط به همان نام میزبان باشد، آنگاه سرور DNS آدرس IP مطلوب را می‌تواند تأمین کند، حتی اگر سرور مختار آن نام میزبان نباشد. این به علت آن است که میزبان‌ها و نگاشت بین نام‌های میزبان و آدرس‌های IP به هیچ عنوان دایمی نیستند. سرورهای DNS پس از گذشت زمان معینی که در اغلب موارد 2 روز می‌باشد نسبت به حذف اطلاعات نهان شده اقدام می‌کنند.

به‌عنوان یک مثال فرض کنید میزبانی مانند: apricot.poly.edu ارجاعی را به dns.poly.edu برای یک آدرس IP متعلق به نام میزبان cnn.com محول می‌کند. علاوه بر آن فرض کنید به فاصله‌ی چند ساعت بعد دانشگاه دیگری به نام: kiwi.poly.edu ارجاعی را به dns.poly.edu محول می‌کند که دارای نام میزبان مشابه قبلی است. از آنجائیکه نهان‌سازی در سرور DNS محلی انجام شده است بی‌درنگ آن سرور آدرس IP متعلق به cnn.com را به درخواست دوم بدون مراجعه به سرورهای دیگر بازگشت می‌دهد. یک سرور DNS محلی می‌تواند به نهان‌سازی آدرس‌های IP متعلق به سرورهای TLD نیز بپردازد و بنابراین این امکان را به وجود آورد که سرورهای DNS محلی از کنار سرورهای DNS ریشه در یک زنجیره‌ی جستجو عبور نمایند. این واقعیت غالباً در شبکه‌ی اینترنت به وقوع می‌پیوندد.

مشابه HTTP، FTP و SMTP پروتکل DNS یک پروتکل لایه‌ی کاربرد است زیرا: (۱) مکان اجرای آن بین سیستم‌های انتهایی در حال محاوره است که از الگوی مشتری- سرور استفاده می‌کنند و (۲) تکیه‌ی آن بر پروتکل انتقال انتها به انتها جهت مبادله‌ی پیام‌های DNS بین میزبان‌های انتهایی در حال محاوره می‌باشد. به عبارت دیگر نقش DNS کاملاً از پروتکل‌های انتقال فایل، وب و کاربرد پست الکترونیکی متفاوت می‌باشد. به عکس این کاربردها، DNS کاربردی به‌شمار نمی‌آید که یک کاربر مستقیماً با آن تعامل برقرار نماید. بلکه DNS یک عملکرد هسته‌ی اینترنت است که وظیفه‌ی آن تبدیل نام‌های میزبان به آدرس‌های IP آنها برای کاربردهای کاربر و دیگر نرم‌افزارهای اینترنت می‌باشد. قبلاً ملاحظه شد پیچیدگی معماری اینترنت در لبه‌ها یا همان نبش شبکه استقرار یافته است. بنابراین DNS که فرآیند حساس ترجمه‌ی نام به آدرس را انجام می‌دهد از برنامه‌های مشتری و سرور مستقر در لبه‌ی شبکه استفاده می‌کند که این خود فلسفه‌ی دیگری از طراحی اینترنت به‌شمار می‌آید.

۲-۱۴ پیام‌ها و سوابق DNS

سرورهای DNS در بین خود به اتفاق یک پایگاه داده‌ی توزیع شده را پیاده‌سازی می‌کنند که به ذخیره‌ی آنچه **سابقه‌ی منابع**[1] (RRS) نامیده می‌شود، می‌پردازند. این سوابق RRS نگاشت نام میزبان‌ها به آدرس‌های IP را انجام دهند. هر پیام به‌عنوان پاسخ DNS حامل یک یا چند سابقه‌ی منبع است. در این بخش خلاصه‌ای از سابقه‌ی منابع (RRS) مربوط به سرورهای DNS، پیام‌های آنها و غیره توضیح داده می‌شود. اسناد RFC به شماره‌های 1034 و 1035 جزئیات مربوط به مشخصات RRS را تعریف می‌کنند. معمولاً آنچه یک سابقه‌ی منبع خوانده می‌شود عبارت است از یک ساختار چهارتایی که شامل محدوده‌های ذیل است:

(Name , Value , Type , TTL)

(زمان خروج , نوع , ارزش , نام)

زمان خروج (TTL) از یک ثبت منبع عبارت است از زمانی که یک منبع بایستی از حافظه‌ی نهان

1- Resource records

سرور خارج و یا از آن حذف شود. در مثال‌های سابقه که ذیلاً آمده است از میدان TTL صرف نظر شده است از طرفی مفهوم هر یک از واژه‌های نام، ارزش و نوع به‌صورت زیر تعریف می‌شوند:

۱- اگر Type = A باشد آنگاه Name نام میزبان و Value آدرس IP آن میزبان می‌باشد. بنابراین یک سابقه از نوع Type A عبارت است از نگاشت استاندارد نام میزبان به آدرس IP، به‌عنوان مثال عبارت:

(relay1.bar.foo.com , 145.37.93.126, A) یک سابقه‌ی از نوع A می‌باشد.

۲- اگر Type = NS باشد، آنگاه Name یک محدوده مانند foo.com و ارزش Value نام میزبان سرور DNS مختار است که می‌داند چگونه آدرس‌های IP برای میزبان‌های آن محدوده قابل دستیابی و شناسایی می‌باشند. از این سابقه جهت مسیردهی جستجوهای DNS در زنجیره‌ی کاوش‌ها استفاده می‌شود. به‌عنوان مثال:

(foo.com , dns.foo.com , NS) از نوع سابقه‌ی Type NS است.

۳- اگر Type = CNAME باشد، آنگاه Value یک نام میزبان متعارف است برای میزبان‌های با نام مستعار Name، این سابقه برای میزبان‌های جستجوگر آدرس نام متعارف یک میزبان را فراهم می‌آورد. به‌عنوان مثال:

(foo.com , relay1.bar.foo.com , CNAME) یک سابقه‌ی CNAME است.

۴- اگر Type = MX باشد، آنگاه Value یک نام متعارف از یک سرور پست الکترونیکی است که در خود یک نام مستعار میزبان Name را دارد. به‌عنوان مثال:

(foo.com , mail.bar.foo.com , MX) یک سابقه‌ی MX است. سابقه‌های از نوع MX اجازه می‌دهند نام‌های میزبان سرور پستی از نام‌های مستعار ساده استفاده کنند. بدیهی است با استفاده از سابقه‌ی MX یک شرکت می‌تواند از همان نام مستعار مشابه برای سرور پستی خود استفاده نماید و هم برای سرورهای دیگرش مانند وب سرور و غیره. برای بدست آوردن نام متعارف سرور پستی، یک مشتری DNS ممکن است به جستجو برای یک سابقه‌ی MX، جهت بازیابی نام متعارف سرور دیگری بپردازد. در این حالت مشتری DNS به جستجو برای یافتن سابقه‌ی CNAME اقدام می‌کند.

اگر یک سرور DNS از نوع مقتدر (مختار) برای یک نام میزبان ویژه باشد، آنگاه سرور DNS شامل یک سابقه‌ی Type A برای نام میزبان است، حتی اگر سرور DNS مختار نباشد، آن سرور ممکن است شامل سابقه‌ای از نوع Type A در حافظه‌ی نهان خود باشد. اگر یک سرور از طرف دیگر حالت مختار را برای یک نام میزبان نداشته باشد آنگاه سرور شامل سابقه‌ی Type NS برای محدوده‌ای است که نام میزبان در درون آن جای گرفته است. آن سرور همچنین حامل یک سابقه‌ی Type A خواهد بود که آدرس IP متعلق به سرور DNS در میدان Value از سابقه‌ی NS را فراهم آورده است. به‌عنوان مثال اگر فرض شود یک سرور TLD که مسئولیت نام‌های edu• را برعهده دارد برای نام میزبانی مثل: gaia.cs.umass.edu سرور مختار نباشد. این سرور شامل یک سابقه برای محدوده‌ای است که میزبان: gaia.cs.umass.edu را در درون خود جای داده است. به‌عنوان مثال: (umass.edu, dns.umass.edu, NS).

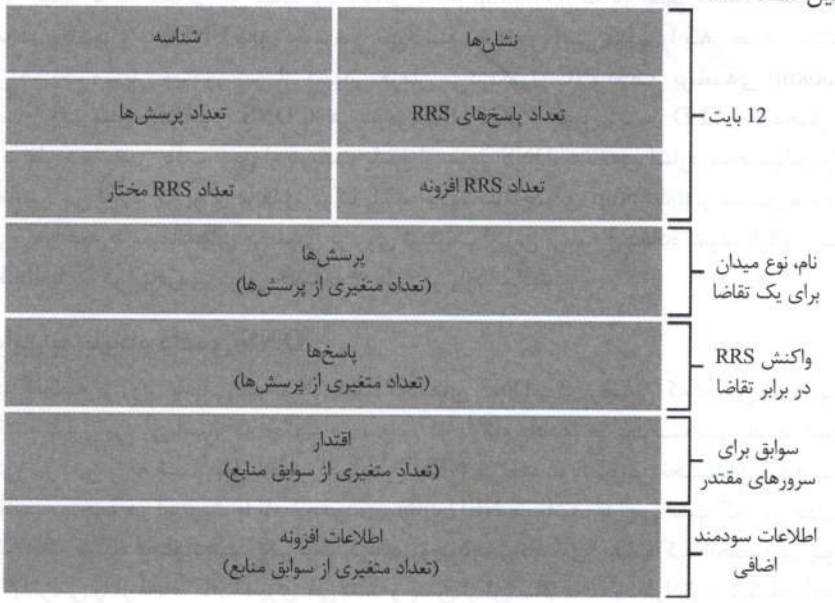
سرور TLD نام‌های edu. همچنین شامل سابقه‌ی از نوع Type A خواهد بود که به نگاشت سرور DNS دارای نام میزبان: dns.umass.edu به یک آدرس IP به‌عنوان مثال شبیه:

(dns.umass.edu , 128.119.40.111 , A) اقدام می‌کند.

• پیام‌های DNS

قبلاً در این بخش جستجو و پاسخ‌های پروتکل DNS بررسی شدند. این دو را می‌توان تنها نوع از پیام‌های DNS به‌شمار آورد. هر دو پیام از لحاظ قالب‌بندی مشابه یکدیگر می‌باشند و به‌صورت شکل ۲- ۲۳ نشان داده شده‌اند. مفاهیم مربوط به میدان‌های مختلف در یک پیام DNS به‌صورت ذیل می‌باشند:

نخستین 12 بایت عبارتست از بخش سرپیام که خود شامل چندین میدان یا محدوده است. به ترتیب این محدوده‌ها عبارتند از: یک دامنه‌ی 16 بیت که شماره‌ی شناسه‌ی تقاضا می‌باشد. این شناسه در یک پیام پاسخ مجدداً نسخه برداری می‌شود تا مشتری قادر به نگاشت پاسخ‌های دریافتی با تقاضاهای ارسالی باشد. تعدادی **نشانه**[1] نیز وجود دارند که همگی در محدوده‌ی نشانه‌ها قرار می‌گیرند. یک نشانه‌ی به طول 1 بیت با مفهوم تقاضا و پاسخ بیان کننده‌ی آن است که آیا پیام مورد نظر تقاضا (0) و یا پاسخ (1) می‌باشد. نشانه‌ی دیگر 1 بیت، نشان اختیار است، این نشانه هنگامی در پیام پاسخ (1) می‌شود که سرور DNS پاسخ دهنده برای یک نام تقاضا شده مختار باشد در غیر این صورت ارزش آن (0) می‌شود. نشانه‌ی یک بیت سوم هنگامی (1) می‌شود که یک برنامه‌ی مشتری اعم از میزبان یا سرور DNS نیازمند عملیات بازگشتی است در صورتی که سابقه‌ای را در اختیار ندارد. این نشانه را اصطلاحاً **بیت بازگشت - نیازمندی** می‌نامند. چهارمین نشانه **بیت بازگشت - در دسترس** نام دارد. هنگامی‌که این نشانه (1) شود مفهوم آن است که پیام پاسخ به‌صورت بازگشتی از طریق DNS گسیل شده است.



شکل ۲- ۲۳ ♦ قالب پیام DNS

علاوه بر موارد اشاره شده تعداد 4 میدان عددی دیگر نیز در بخش سرپیام وجود دارد. این میدان‌ها وقوع چهار نوع از بخش‌های داده را که به دنبال سرپیام می‌آیند، نشان می‌دهد. این موارد عبارتند از:

1- Flag

۱- بخش پرسش که شامل اطلاعات در مورد جستجوی در حال انجام است. این بخش شامل (۱) یک میدان نام است و نام مورد جستجو را نگهداری می‌کند. (۲) یک میدان نوع که به مفهوم گونه‌ی پرسشی است که در مورد نام مطرح شده است، به‌عنوان مثال یک آدرس میزبان وابسته به یک نام Type A یا سرور پستی برای یک نام Type MX.

۲- در یک پیام پاسخ از سرور DNS که اصطلاحاً بخش پاسخ نامیده می‌شود سابقه‌ی منابع (RRS) برای نامی که در اصل مورد جستجو قرار گرفته است، وجود دارد. یادآوری می‌شود که در هر ثبت منبع یک Type وجود دارد به‌عنوان مثال A، NS، CNAME و یا MX، یک ارزش Value و یک زمان خروج TTL. یک پیام پاسخ می‌تواند چندین RRS را در درون خود داشته باشد، زیرا یک نام میزبان می‌تواند شامل چندین آدرس IP مانند وب سرورهای تکراری باشد.

۳- بخش اعتبار یا نفوذ که سوابق سرورهای مختار دیگر را شامل می‌شود.

۴- بخش افزونه که شامل دیگر سوابق مفید است. به‌عنوان مثال بخش پاسخ در یک پیام تقاضای MX می‌تواند شامل یک سابقه‌ی منبع باشد، که نام میزبان یا متعارف را برای یک سرور پستی فراهم می‌آورد. بخش افزونه شامل یک سابقه‌ی TypeA نیز می‌شود که آدرس IP نام میزبان متعارف از یک برنامه‌ی سرور پستی را فراهم می‌آورد.

اما با وجود این. توضیحات ممکن است، این پرسش مطرح شود، چگونه می‌توان یک پیام تقاضای DNS برای جستجو را مستقیماً از میزبانی شناخته شده به دیگر سرورهای DNS گسیل نمود؟ پاسخ این پرسش آن است که به سادگی به کمک برنامه‌هایی با نام nslookup که در اغلب سیستم عامل‌های کامپیوتری مانند ویندوز و یا UNIX قابل دسترسی می‌باشند، می‌توان این مهم را به انجام رسانید. به‌عنوان مثال در یک میزبان ویندوز پس از گشودن فرمان بی‌درنگ (C:\) و احضار برنامه‌ی nslookup می‌توان با ارسال یک پیام جستجوی DNS به هر سرور DNS دیگر از قبیل ریشه، TLD و یا مختار به نتیجه‌ی مورد نظر دسترسی یافت. پس از دریافت پاسخ از سرور DNS، برنامه‌ی اشاره شده سوابق را به گونه‌ای به نمایش می‌گذارد که برای ما قابل درک باشد. علاوه بر اجرای nslookup از طریق میزبان خود می‌توان با مراجعه به سامانه‌های متعددی بر روی اینترنت از این برنامه استفاده نمود. کافی است عبارت: nslookup را در پنجره‌ی موتور جستجوی مرورگر وارد نمائید.

تزریق سوابق به پایگاه داده‌ی DNS

تمرکز مطالب گذشته بر روی بازیابی سوابق از پایگاه داده‌های DNS بود. پرسشی که ممکن است پیش روی هر خواننده قرار گیرد آن است که چگونه‌این سوابق در پایگاه داده‌ها موجودیت می‌یابند. فرض کنید به تازگی شرکتی را به ثبت رسانیده‌اید که نام آن NEU می‌باشد، بنابراین نخستین قدم ثبت شرکت تازه تأسیس خود در اینترنت با نام محدوده‌ی NEU.com می‌باشد. بدیهی است که این مهم را بایستی با یک ثبت کننده انجام دهید. یک ثبت کننده موجودیت تجاری است که انحصاری بودن محدوده‌های اینترنتی را از لحاظ نام ممیزی می‌کند و پس از آن نام محدوده را وارد پایگاه داده در سرور DNS می‌نماید. علاوه بر همه‌ی این موارد اقدام به دریافت هزینه‌ی نسبتاً ناچیزی نیز جهت ارایه‌ی این خدمات می‌کند. تا قبل از سال 1999 میلادی تنها یک ثبات به نام **راهکارهای شبکه**[1] (NS) وجود داشت که همه‌ی محدوده نام‌ها از قبیل com، net و org را در انحصار خود گرفته

1- Network Solutions

بود. اما در حال حاضر شرکتهای ثبت‌کننده‌ی متعددی در حال رقابت با یکدیگر در جلب متقاضیان می‌باشند. نهاد مهم دیگری به نام ICANN [1] نظارت و رسمیت بخشیدن به اینگونه شرکتها و سایر موارد مهم کنترل و مدیریت اینترنت را بر عهده دارد. فهرست کاملی از ثبت کننده‌های رسمی و معتبر را در آدرس URL ذیل می‌توان جستجو نمود:

http://www.internic.net

هنگامی‌که نام محدوده NEU.com از طریق ثبت‌کننده‌ای به ثبت می‌رسد، شما نیازمند اعلام نام‌ها و آدرس‌های IP متعلق به سرورهای DNS مختار ابتدایی و ثانوی خود نیز به آن شرکت ثبات می‌باشید. فرض کنید نام‌ها و آدرس‌های IP سرورها به‌ترتیب: dns1.NEU.com، dns2.NEU.com، 212.212.212.1 و 212.212.212.2 می‌باشند. برای هر یک از این دو DNS مختار، ثبت‌کننده این اطمینان را ایجاد می‌کند که به ترتیب یک سابقه‌ی منبع از نوع TypeNS و Type A وارد سرورهای TLD مسئول com. خواهد شد. به‌ویژه برای سرور مختار اولیه NEU.com، ثبات به تزریق دو سابقه‌ی منبع به درون سیستم DNS می‌پردازد:

(NEU.com, dns1.NEU.com , NS)

(dns1.NEU.com , 212.212.212.1 , A)

افزون بر همه‌ی این موارد شما بایستی اطمینان حاصل نمائید که سابقه‌ی منبع Type A برای وب سرور به صورت: www.NEU.com و سابقه‌ی منبع Type MX برای سرور پستی به گونه ی: mail.NEU.com در درون برنامه‌ی سرور DNS مختار تزریق شده است. تا این اواخر محتویات هر یک از سرورهای DNS به‌صورت ایستا (ثابت) پیکربندی می‌شد به‌عنوان مثال از یک فایل پیکربندی به‌وسیله مدیر سیستم جهت انجام این مهم استفاده می‌گردید. اخیراً به پروتکل DNS یک عامل اختیاری به نام UPDATE اضافه گردیده است که اجازه می‌دهد داده به‌صورت پویا از طریق پیام‌های DNS به پایگاه داده‌ها اضافه و یا از آنها حذف گردد. اسناد RFC شمارگان 2136 و 3007 جزئیات به روزرسان‌های پویای DNS را مشخص نموده‌اند.

هنگامی‌که همه‌ی این مراحل کامل گردد، آنگاه همه‌ی کاربران اینترنتی می‌توانند سامانه‌ی وب شما را بازدید نمایند و یا پیام الکترونیکی برای کارکنان شرکت شما ارسال نمایند. در انتهای مطالب مربوط به DNS به بیان مثال ذیل می‌پردازیم: فرض کنید A در استرالیا می‌خواهد صفحه‌ی وب دارای آدرس URL به صورت: www.NEU.com را بازدید نماید. همانگونه که قبلاً اشاره گردید، میزبان او ابتدا یک درخواست DNS را به سرور محلی ارسال می‌کند. سرور DNS محلی آنگاه با یک سرور TLD مسئول com. تماس می‌گیرد. سرور DNS محلی همچنین بایستی با سرور DNS ریشه ارتباط برقرار نماید تا بررسی نماید که آیا یک سرور TLD مسئول com. این آدرس را نهان‌سازی نموده است یا خیر؟ بدیهی است این سرور TLD سوابق منابع (RRS) از نوع Type NS و Type A از صورت بالا را در درون خود جای داده است. زیرا ثبات قبلاً این اطلاعات را به TLD تزریق نموده است. سرور TLD مسئول com. پاسخی را به سرور DNS محلی A ارسال می‌کند که در این پیام پاسخ دو RRS اشاره شده در بالا نیز موجود می‌باشد. سرور DNS محلی یک تقاضای DNS به آدرس 212.212.212.1 ارسال و از آن برای سابقه‌ی از نوع Type A در ارتباط با www.NEU.com پرسش می‌کند. این سابقه آدرس IP وب سرور مورد نظر را که می‌تواند به‌صورت 212.212.212.4 باشد، در اختیار می‌گذارد. سرور DNS محلی

1- Internet Corporation for Assigned Names and Numbers

این آدرس را به میزبان A بازگشت می‌دهد. مرورگر A اکنون می‌تواند یک ارتباط TCP را با میزبان 212.212.212.4 برقرار نماید و یک درخواست HTTP را از طریق این ارتباط گسیل دارد. ملاحظه می‌شود که چه مراحل شگفت‌انگیز و در عین حال پیچیده‌ای روی می‌دهد تا یک کاربر در گوشه‌ای از کره زمین موفق به دریافت یک صفحه‌ی وب از یک سامانه گردد.

- **آسیب‌پذیری DNS (تمرکز بر امنیت)**

ملاحظه گردید پروتکل DNS یکی از عناصر بسیار حساس و حیاتی در زیرساخت شبکه‌ی اینترنت می‌باشد. بسیاری از خدمات مهم را پشتیبانی می‌کند. که از جمله‌ی آنها وب و پست الکترونیکی است. به زبان ساده بدون DNS هیچگونه راهبردی امکان‌پذیر نیست. بنابراین طبیعی است که این پرسش را مطرح نمائیم که DNS چگونه می‌تواند مورد تهدید و تهاجم قرار گیرد؟ آیا DNS یک هدف بی دفاع است؟ که منتظر است تا از سرویس رسانی خارج گردد؟ اگر چنین شود بسیاری از خدمات اینترنتی نیز متوقف خواهند شد.

از کار انداختن سرورهای DNS نخستین نوع تهاجمی است که در ذهن متبادر می‌شود و یک تهدید از نوع DDoS می‌تواند موفق به انجام این عمل گردد. به‌عنوان مثال یک مهاجم می‌تواند تلاش کند که به سوی هر یک از سرورهای DNS ریشه بسته‌های اطلاعاتی سیلابگری گسیل نماید، بطوریکه اکثر تقاضاهای مشروع DNS هرگز پاسخگویی نگردند. چنین تهاجم گسترده‌ای به گونه‌ی DDoS بر علیه سرورهای DNS ریشه در روز 21 ماه اکتبر سال 2002 به وقوع پیوست. مهاجمین با استفاده از قدرت نفوذ یک **روبات اینترنتی**[1] (نرم‌افزار هوشمند) انبوهی از **پیام‌های پینگ**[2] ICMP را به هر یک از 13 سرور DNS ریشه گسیل نمودند. در فصل ۴ پیام‌های ICMP بررسی شده‌اند. یک بسته‌ی ICMP از داده گرام‌های ویژه‌ی IP تشکیل شده است. البته این تهاجم وسیع و گسترده خوشبختانه سبب بروز حداقل زیان گردید زیرا که تأثیر مستقیم و مخربی را بر روی کاربران اینترنت به جای نگذاشت. آنچه هدف مهاجمان بود توان مندی در هدایت بسته‌های سیلاب گر به سمت سرورهای DNS ریشه بود که بدیهی است در رسیدن به این هدف نیز موفق شدند. البته بسیاری از سرورهای DNS ریشه از طریق صافی‌های ویژه‌ی بسته، که پیکربندی آنها در جهت مسدود نمودن دایم پیام‌های پینگ ICMP، به سمت سرورهای ریشه بود محافظت شده بودند. در نتیجه سرورهای محافظت شده از این تهاجم در امان ماندند. از آنجائیکه بسیاری از سرورهای DNS محلی آدرس‌های IP متعلق به سرورهای TLD را نهان‌سازی می‌کنند، در نتیجه این امکان به وجود می‌آید که فرآیندهای جستجو غالباً از کنار سرورهای DNS ریشه عبور نمایند.

تهدید مهم و مؤثر دیگر از نوع DDoS است، که بر علیه سرورهای DNS می‌تواند صورت پذیرد، در این نوع تهاجم تمرکز بر روی ارسال پیام‌های سیلاب گر تقاضا به سمت سرورهای TLD می‌باشد. به‌عنوان مثال ارسال تقاضا به تمامی سرورهای TLD مسئول محدوده‌های com.، بدیهی است، فیلترسازی این تهاجم بسیار دشوارتر از حمله به سوی سروزهای DNS ریشه می‌باشد. زیرا عبور از کنار سرورهای TLD به سادگی عبور از سرورهای ریشه نمی‌باشد. البته شدت تأثیر این تهاجم را نیز می‌توان با نهان‌سازی آدرس‌ها در سرورهای DNS محلی تا حدود زیادی کاهش داد.

تهاجم به سرورهای DNS از روش‌های دیگر نیز امکان‌پذیر است. از جمله در یک تهاجم از طریق واسطه که در فصل یک به اختصار توضیح داده شد، برنامه‌ی مهاجم با دریافت تقاضاها از میزبان‌ها و با

1- Botnet
2- Ping

بازگشت پاسخ‌های جعلی به آنها عملاً تأثیر مخرب و مؤثری را می‌تواند از خود در اینترنت به جای گذارد. در تهدیدهایی که از طریق آنها سرورها آلوده و در نتیجه مختل شوند، مهاجم با ارسال پاسخ‌های جعلی به سرورهای DNS آنها را وادار می‌سازد که سوابق جعلی را پذیرش نموده و در حافظه‌ی خود نهان‌سازی نمایند. آنگاه از هر یک از این تهاجم‌ها می‌توان جهت انحراف مسیر یک وب سرور عاری از آلودگی به سوی سامانه‌ی وب مهاجم استفاده نمود. پیاده‌سازی این تهدیدها در عمل دشوار می‌باشد زیرا آنها نیازمند رهگیری و برخورد با بسته‌ها و یا از کار انداختن سرورها می‌باشند. تهدید مهم دیگر در قالب تهاجم مستقیم به DNS نمی‌باشد بلکه از زیرساخت سرورهای DNS برای یک حمله‌ی از نوع DDoS بر علیه یک میزبان هدف مانند سرور پست الکترونیکی یک دانشگاه استفاده می‌شود. در این نوع حمله، به سوی تعداد زیادی از سرورهای DNS مختار درخواست‌هایی از نوع پروتکل DNS از جانب مهاجم سرازیر می‌شود. هر یک از این درخواست‌ها یک آدرس منبع مجعول و بدون مفهوم از میزبان هدف را با خود حمل می‌کند. سرورهای DNS در نتیجه پاسخ‌های خود را مستقیماً به میزبان هدف گیری شده ارسال می‌کنند. اگر طراحی درخواست‌ها به گونه‌ای باشد که نتیجه‌ی آن یک پاسخ به مراتب بزرگتر از نقطه‌نظر طول بایت در مقایسه با پیام تقاضای ارسالی باشد آنگاه مهاجم می‌تواند بدون نیاز به تولید حجم انبوهی از ترافیک به نابودی هدف از لحاظ ترافیکی بپردازد. البته تا امروز چشم‌انداز چنین تهدیدهایی که DNS خود سبب نابودی خویش گردد چندان توأم با موفقیت نبوده است. به طور خلاصه DNS نشان داده است که خود به تنهایی در برابر تهاجم‌ها مقاوم بوده و هیچگونه تهدید موفقیت آمیزی تا این تاریخ نتوانسته است سبب اختلال در خدمات گردد. بدیهی است تهاجم‌های از نوع انعکاسی بسیاری به‌صورت موفقیت‌آمیزی انجام شده است اما آنها نیز عملاً در برابر پیکربندی مناسب سرورهای DNS خنثی شده‌اند.

۲-۱۵ کاربردهای نظیر به نظیر (P2P)

کاربردهایی که تاکنون بررسی شدند شامل وب، پست الکترونیکی و پروتکل DNS همگی از معماری مشتری- سرور استفاده می‌نمودند. علاوه بر آن وابستگی چشمگیری از لحاظ عملکرد خود به زیرساخت سرورهای همیشه روشن (فعال) داشتند. قبلاً در بخش ۲-۱ اشاره گردید که با یک معماری P2P تقریباً هیچگونه و یا حداقل وابستگی به اینگونه زیرساخت سرورها ضروری می‌باشد. در عوض جفت میزبان‌های تناوبی متصل به یکدیگر که نظیر خوانده می‌شوند، مستقیماً با یکدیگر ارتباط برقرار می‌کنند. از طرفی نظیرها در مالکیت ISPها نمی‌باشند بلکه کامپیوترهای شخصی و یا همراهی هستند که کاربران آنها را کنترل می‌کنند. در این بخش سه نوع کاربرد متناسب با طراحی P2P تجزیه و تحلیل می‌گردد. نخست توزیع فایل که در آن کاربرد به توزیع یک فایل از یک منبع تک به تعداد بی شماری از نظیرها می‌پردازد. توزیع فایل نقطه‌ی آغاز جالبی برای بررسی P2P می‌باشد چرا که معماری خود مقیاس پذیری P2P را از آن می‌توان دریافت. یک نمونه‌ی روشن توزیع فایل، سیستم پرطرفدار **تنداب بیت**[1] **(بیت تُرنت)** است که توصیف می‌شود. دومین کاربرد P2P بررسی توزیع پایگاه داده‌ها از طریق جمعیت انبوهی از نظیرها می‌باشد. برای این کاربرد به کاوش پیرامون اصول **جدول توزیع درهم**[2] (DHT) می‌پردازیم و سرانجام سومین کاربرد یک رویداد بسیار موفق P2P برای کاربرد تلفن اینترنتی است که از آن به‌عنوان **اسکایپ**[3] نام برده می‌شود.

1- Bit torrent
2- Distributed Hash Table
3- Skype

- **توزیع فایل P2P**

بررسی P2P را با یک کاربرد بسیار طبیعی که عبارت است از توزیع یک فایل بسیار پرحجم از یک سرور تک به تعداد قابل توجهی میزبان (نظیرها) آغاز می‌کنیم. به‌عنوان مثال فرض کنید این فایل نسخه‌ی جدیدی از سیستم عامل Linux است به عبارت دیگر یک قطعه‌ی نرم‌افزاری برای سیستم عامل فعلی یا کاربردی مانند یک فایل موسیقی MP3 و یا یک فایل ویدیویی MPEG. در توزیع فایل به‌صورت مشتری- سرور نسخه‌ای از یک فایل بایستی به هر یک از نظیرها از سوی سرور گسیل گردد که این خود بار بسیار بزرگی را بر دوش سرور می‌گذارد و سبب می‌شود مقدار وسیعی از پهنای باند سرور صرف این مبادله گردد. از طرفی در توزیع فایل P2P هر نظیر می‌تواند به توزیع مجدد هر بخشی از فایل دریافتی به هر یک از نظیرهای دیگر بپردازد و بنابراین سرور را در یک فرآیند توزیع کمک نماید. تا هنگام نگارش این اثر در پائیز 2012 میلادی پرطرفدارترین پروتکل توزیع فایل P2P پروتکل بیت ترنت (تُندآب بیت) می‌باشد. این پروتکل در آغاز به‌وسیله فردی به نام برام کوُهن طراحی و پیاده‌سازی شد اما در حال حاضر تعداد بی‌شماری برنامه‌های مشتری مستقل و متفاوت بیت ترنت در سراسر گیتی وجود دارند که همگی در سازگاری با پروتکل اصلی بیت ترنت می‌باشند. همانگونه که تعداد قابل توجهی برنامه‌های مرورگر مشتری وب وجود دارند که همگی سازگار با پروتکل HTTP می‌باشند. در این بخش نخست خود مقیاس پذیری معماری P2P در چارچوب توزیع فایل بررسی می‌گردد و سپس مهمترین ویژگی‌ها و مشخصات این پروتکل تحلیل می‌شود.

- **مقیاس پذیری معماری P2P**

برای مقایسه معماری مشتری- سرور با رقیب آن یعنی پروتکل نظیر به نظیر (P2P) و نیز مشخص نمودن خود مقیاس پذیری درونی (ذاتی) P2P، از یک الگوی کمیتی ساده برای توزیع یک فایل به یک مجموعه‌ی ثابت از نظیرها برای هر دو نوع معماری بهره می‌گیریم. همانگونه که شکل ۲- ۲۴ نشان می‌دهد سرورها و نظیرها با استفاده از پیوندهای دستیابی به اینترنت به یکدیگر متصل شده‌اند. اگر نرخ یا سرعت بارگذاری پیوند دستیابی سرور را μ_s بنامیم و نرخ بارگذاری پیوند دستیابی i اُمین نظیر را μ_i و نیز سرعت بارگیری پائین پیوند دستیابی همان نظیر را d_i و سرانجام اگر اندازه‌ی یک فایل که بایستی توزیع گردد F بیت و تعداد نظیرهایی که مایل به نسخه برداری از فایل می‌باشند N در نظر گرفته شود. زمان توزیع فایل عبارت است از مدت زمانی که نسخه‌ای از فایل به همگی N نظیر خواهد رسید. در بررسی زمان توزیع برای هر دو معماری مشتری- سرور و نظیر به نظیر از یک فرض ساده و کلی نسبتاً دقیق استفاده شده است و آن این موضوع است که هسته‌ی اینترنت را از لحاظ پهنای باند (سرعت) وافر می‌داند. به عبارت دیگر تنگراه‌ها را فقط در دستیابی شبکه می‌داند. همچنین فرض بر آن است که سرور و مشتری‌ها در هیچیک از کاربردهای شبکه شرکت ندارند، بنابراین همه‌ی سرعت عملیات بارگذاری به طرف بالا و بارگذاری‌های پائین فقط به توزیع این فایل اختصاص یافته است. ابتدا زمان توزیع فایل برای معماری مشتری- سرور که D_{cs} نامگذاری شده است، محاسبه می‌گردد. در معماری مشتری- سرور هیچیک از نظیرها در توزیع فایل کمک کار نخواهند بود البته موارد ذیل نیز در ملاحظات قرار می‌گیرند:

- سرور بایستی یک نسخه از فایل را به هر یک از تعداد N نظیر انتقال دهد. بنابراین حجم ارسالی معادل NF بیت می‌باشد. از آنجائیکه نرخ بارگذاری سرور μ_s است زمان توزیع فایل حداقل $\frac{NF}{\mu_s}$ می‌باشد.

- اگر d_{min} نرخ بارگیری پائین نظیر با کندترین سرعت ممکن باشد. به عبارت دیگر $d_{min} = \{d_1, d_p, \ldots, d_N\}$ در نتیجه نظیر با کندترین سرعت نمی‌تواند همه‌ی F بیت فایل را در زمان کمتر از $\frac{F}{d_{min}}$ دستیابی نماید. بنابراین حداقل زمان توزیع در کمترین مقدار ممکن خود برابر $\frac{F}{d_{min}}$ می‌باشد.

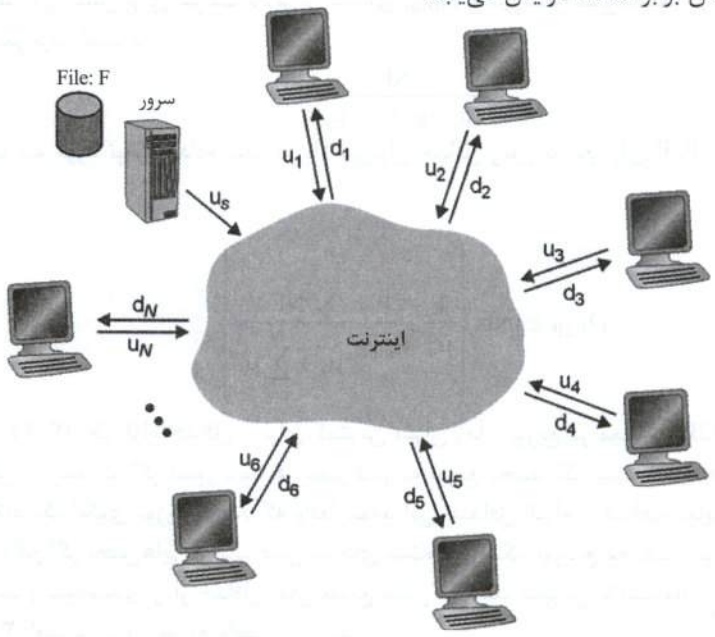
با قرار دادن این دو مورد در کنار یکدیگر رابطه‌ی زیر را می‌توان تعریف نمود:

$$D_{cs} \geq \max \left\{ \frac{NF}{\mu_s} , \frac{F}{d_{min}} \right\} \qquad (۱-۲)$$

این رابطه یک الزام حداقل را برای کمترین میزان زمان توزیع در معماری سرور- مشتری تعریف می‌کند. اگر این نیاز حداقل به‌عنوان زمان توزیع واقعی در نظر گرفته شود آنگاه:

$$D_{cs} = \max \left\{ \frac{NF}{\mu_s} , \frac{F}{d_{min}} \right\} \qquad (۲-۲)$$

از معادله‌ی (۲-۲) چنین نتیجه‌گیری می‌شود که اگر N به اندازه‌ی کافی بزرگ باشد، زمان توزیع مشتری- سرور از طریق رابطه‌ی $\frac{NF}{\mu_s}$ بدست می‌آید و بنابراین زمان توزیع به گونه‌ی خطی با تعداد نظیرهای N افزایش می‌یابد. به‌عنوان مثال اگر تعداد نظیرها از یک هفته به هفته‌ی بعد یک هزار برابر افزایش پیدا کند یعنی مثلاً از 1000 به یک میلیون برسد، آنگاه زمان لازم جهت توزیع فایل به همه‌ی نظیر با یک مقیاس برابر 1000 افزایش می‌یابد.



شکل ۲- ۲۴ ♦ نمادی از چالش توزیع فایل

اکنون تحلیل مشابهی را برای معماری P2P انجام می‌دهیم. در این حالت هر کدام از نظیرها به تنهایی می‌توانند سرور را در توزیع فایل کمک نمایند. هنگامی‌که یک نظیر مقداری از داده‌ی متعلق به

یک فایل را دریافت می‌کند، می‌تواند از ظرفیت بارگذاری خود جهت توزیع آن داده به دیگر نظیرها استفاده نماید. محاسبه‌ی زمان توزیع برای معماری P2P کمی پیچیده‌تر از معماری مشتری- سرور است زیرا زمان توزیع وابسته به رفتار هر نظیر در انتقال بخشی از فایل به نظیرهای دیگر دارد. اما در هر حال استنتاج یک عبارت ساده برای حداقل زمان توزیع امکان‌پذیر می‌باشد. برای دستیابی به چنین عبارتی ملاحظات ذیل در نظر گرفته می‌شود:

- در آغاز توزیع، فقط سرور حامل فایل می‌باشد. برای انتقال این فایل به جمعیتی از نظیرها، سرور بایستی هر یک از بیت‌های فایل را حداقل یکبار به درون پیوند دستیابی تزریق کند. بنابراین حداقل زمان توزیع برابر $\frac{F}{\mu_s}$ می‌باشد. برعکس روش مشتری- سرور با یک بار ارسال بیت از سوی سرور این احتمال قوی وجود دارد که مجدداً نیاز به ارسال آن بیت از سوی سرور نباشد زیرا نظیرها ممکن است به توزیع بیت در بین خود بپردازند.
- مشابه معماری مشتری- سرور یک نظیر با پائین‌ترین نرخ بارگیری نمی‌تواند همه‌ی F بیت یک فایل را در زمان کمتر از $\frac{F}{d_{min}}$ دستیابی نماید. بنابراین حداقل زمان توزیع برابر است با $\frac{F}{d_{min}}$.
- سرانجام مشاهده این نکته که مجموع سرعت بارگذاری در یک سیستم به طور کلی برابر است با مجموع نرخ‌های بارگذاری سرور و هر یک از نظیرها به تنهایی، به عبارت دیگر:

$$\mu_{total} = \mu_s + \mu_1 + ... + \mu_n$$

سیستم بایستی بارگذاری F بیت را به هر یک از N نظیر انجام دهد. در نتیجه تعداد NF بیت را انتقال می‌دهد. این عمل برای سرعت بالاتر از اندازه‌ی μ_{total} امکان‌پذیر نمی‌باشد. در نتیجه حداقل زمان توزیع نیز برابر است با:

$$\frac{NF}{(\mu_s + \mu_1 + ... + \mu_n)}$$

با ترکیب سه مورد توضیح داده شده در بالا می‌توان حداقل زمان توزیع برای P2P را به‌صورت ذیل تعریف نمود:

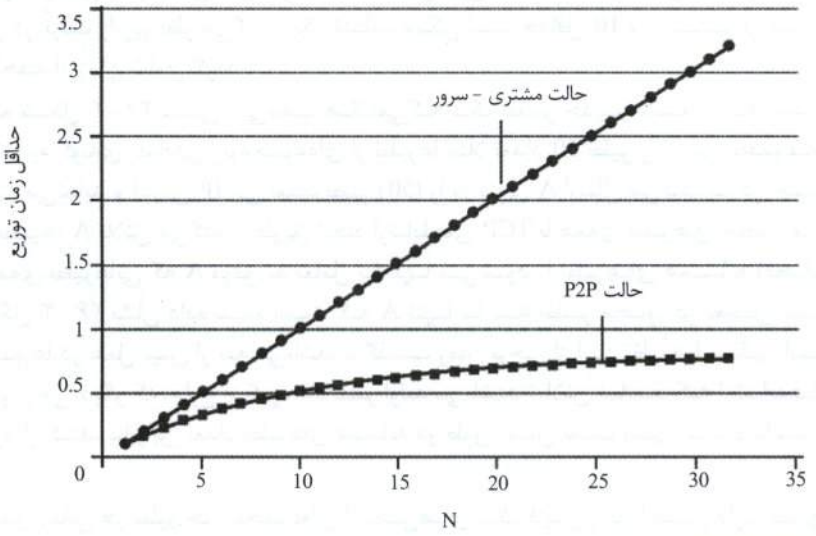
$$D_{P2P} \geq \max\left\{\frac{F}{\mu_s}, \frac{F}{d_{min}}, \frac{NF}{\mu_s + \sum_{i=1}^{N}\mu_i}\right\} \quad (۲-۳)$$

معادله‌ی (۲-۳) یک الزام حداقل را برای کمترین میزان زمان توزیع در معماری P2P نشان می‌دهد. چنین به نظر می‌رسد که اگر تصور شود هر نظیر قادر به توزیع مجدد یک بیت بلافاصله پس از دریافت آن باشد، آنگاه یک الگوی توزیع مجدد که واقعاً بتواند این حداقل الزام را فراهم نماید، امکان‌پذیر می‌باشد. در واقع اگر بخش‌هایی از یک فایل به جای بیت‌های تک توزیع مجدد شوند معادله ۲-۳ تقریب مناسب و سودمندی را از حداقل زمان توزیع نشان می‌دهد. بنابراین با استفاده از الزام حداقل معادله‌ی ۲-۳ کمترین زمان توزیع واقعی برابر است با:

$$D_{P2P} = \max\left\{\frac{F}{\mu_s}, \frac{F}{d_{min}}, \frac{NF}{\mu_s + \sum_{i=1}^{N}\mu_i}\right\} \quad (۲-۴)$$

شکل ۲- ۲۵ به مقایسه‌ی حداقل زمان توزیع برای دو معماری مشتری- سرور و نظیر به نظیر می‌پردازد. البته فرض بر آن است که همه‌ی نظیرها از سرعت بارگذاری مشابه μ برخوردار می‌باشند. در این مقدار $\frac{F}{\mu}=1$ (یک ساعت)، $\mu_s=10\mu$ و $d_{min}\geq\mu_s$ در نظر گرفته شده‌اند. بنابراین یک نظیر می‌تواند تمامی فایل را در یک ساعت انتقال دهد، نرخ انتقال سرور 10 برابر بیشتر از سرعت بارگذاری نظیر است و برای ساده شدن موضوع نرخ بارگیری پائین نظیر به اندازه‌ی کافی بزرگ تنظیم می‌شود که تأثیری را در روند انتقال سبب نشود.

از شکل ۲- ۲۵ ملاحظه می‌شود برای معماری مشتری- سرور افزایش زمان توزیع خطی و بدون الزام در ارتباط با افزایش نظیرها است، در صورتی‌که برای معماری P2P حداقل زمان توزیع نه تنها همواره کمتر از زمان توزیع معماری مشتری- سرور است بلکه، کمتر از یک ساعت برای هر تعداد نظیر N می‌باشد. نتیجه آنکه کاربرد P2P می‌تواند مقیاس پذیر باشد. این ویژگی نتیجه‌ی مستقیم این واقعیت است که نظیرها هم مصرف کننده‌ی بیت‌ها و نیز توزیع کننده‌ی مجدد آنها می‌باشند.



شکل ۲- ۲۵ ♦ زمان توزیع فایل برای معماری مشتری- سرور و نظیر به نظیر

۲- ۱۶ تُندآب بیت (بیت تُرنت)

نرم‌افزار بیت ترنت در قالب پروتکل P2P جهت توزیع فایل‌ها از محبوبیتی زیاد بین کاربران اینترنت برخوردار است. در ادبیات بیت ترنت مجموعه‌ای از همه‌ی نظیرهای شرکت کننده در توزیع یک فایل ویژه اصطلاحاً **تُرنت**[1] (تندآب) نامیده می‌شوند. نظیرها در یک تندآب به **بارگیری پائین**[2] بخش‌های مساوی از اندازه و حجم یک فایل بین خود می‌پردازند. اندازه‌ی هر قطعه از فایل معادل 256 کیلو بایت است. هنگامی‌که یک نظیر نخستین بار به یک تندآب وارد می‌شود هیچ بخشی از فایل

1- Torrent
2- Download

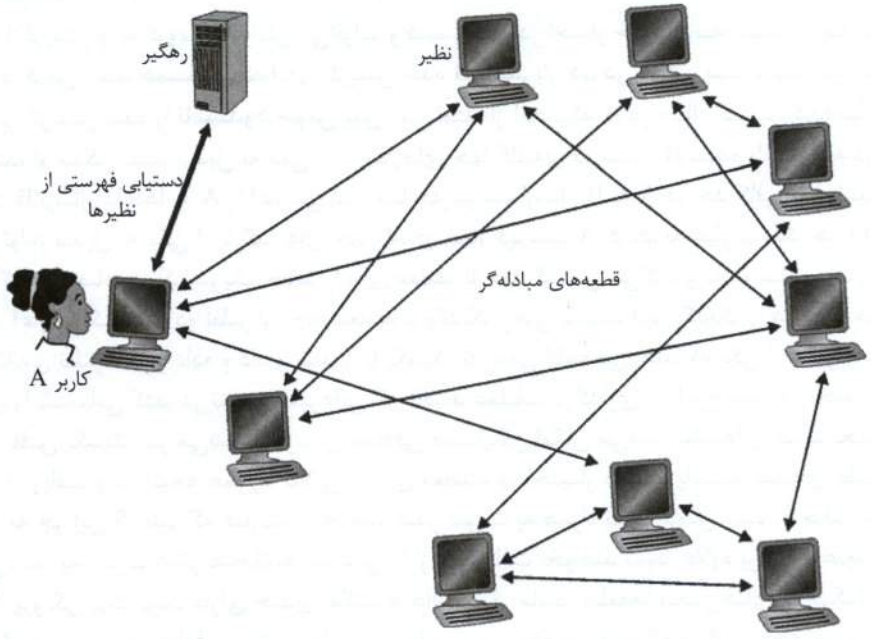
را در اختیار ندارد. با گذشت زمان آن نظیر بخش‌های بیشتر و بیشتری را گردآوری و انباشت می‌کند. هر نظیر در عین حالی که در حال بارگیری بخش‌ها می‌باشد، آنها را در نظیرهای دیگر نیز بارگذاری می‌کند. هنگامی‌که یک نظیر همه‌ی حجم فایل را بدست آورد، آن نظیر ممکن است به گونه خودخواه تندآب را ترک نماید و یا آنکه ممکن است به گونه‌ی نوعدوستی در ترنت باقی مانده و در بارگذاری بخش‌ها به دیگر نظیرها به حضور خود ادامه دهد. علاوه بر این هر نظیر ممکن است در هر لحظه فقط با دریافت بخش‌هایی از فایل، تندآب را ترک گفته و مجدداً در وقت دیگری به آن بپیوندد.

اکنون به بررسی جزئیات بیشتری از چگونگی عملکرد بیت ترنت می‌پردازیم. از آنجائیکه بیت ترنت یک سیستم و پروتکل پیچیده‌ای است، مکانیزم‌های مهم آن در این بخش از کتاب بررسی می‌شوند و برخی از جزئیات آن پنهان باقی می‌ماند. این رویکرد به ما اجازه می‌دهد جنگل را از درون درختان آن شناسایی کنیم. هر تُرنت (تُندآب) دارای یک گره‌ی زیرساخت می‌باشد که اصطلاحاً **رهگیر**[1] نامیده می‌شود. هنگامی‌که یک نظیر به تندآب پیوست، آن نظیر خود را به رهگیر معرفی و سپس ثبت می‌کند و به‌صورت متناوب رهگیر را از حضور خود در تندآب مطلع می‌سازد. با این روش رهگیر همه‌ی نظیرهای حاضر در تُرنت را زیر نظر می‌گیرد. یک تندآب ممکن است حداقل 10 و یا بیشتر از هزاران نظیر را در هر لحظه از زمان شامل گردد.

همانگونه که شکل ۲- ۲۶ نشان می‌دهد هنگامی‌که یک نظیر جدید مانند A به تندآب می‌پیوندد، رهگیر به گونه‌ی تصادفی زیرمجموعه‌ای از نظیرها مثلاً تعداد 50 نظیر را از بین مجموعه‌ی حاضرین گزینش می‌نماید و آدرس IP این تعداد نظیر (50) را به سوی A ارسال می‌کند. با در اختیار داشتن صورت نظیرها A تلاش می‌کند از طریق ایجاد ارتباط‌های TCP با همه‌ی نظیرهای حاضر تعامل برقرار نماید. همه‌ی نظیرهایی که A موفق به تعامل با آنها می‌شود را نظیرهای همسایه (مجاور) می‌نامند. در شکل ۲- ۲۶ نشان داده شده است که A تنها با سه نظیر مجاور در تعامل است. معمولاً تعداد نظیرها در عمل بیش از سه می‌باشد. با گذشت زمان برخی از این نظیرها ممکن است جریان را ترک و برخی دیگر که خارج از گروه 50 نظیر اولیه می‌باشند تلاش نمایند که ارتباط‌های TCP با A را برقرار کنند. بنابراین تعداد نظیرهای همسایه در طول زمان ثابت نمی‌باشد و دایماً در نوسان است.

در هر لحظه‌ی زمانی هر نظیر خود مجموعه‌ای از بخش‌های یک فایل را در اختیار دارد. بدیهی است نظیرهای مختلف زیر مجموعه‌های گوناگون از یک فایل را در اختیار دارند. به گونه‌ی متناوب A از طریق ارتباط‌های TCP از هر یک از نظیرهای مجاور خود تقاضای فهرست بخش‌هایی از فایل را که آنها در اختیار دارند، می‌کند. اگر A دارای تعداد L همسایه‌ی مختلف است و تعداد L فهرست از بخش‌های فایل در حال انتقال را بدست می‌آورد، با در اختیار داشتن این دانش A تقاضای خود را از طریق ارتباط TCP برای آن دسته از بخش‌هایی که در حال حاضر در اختیار ندارد، صادر می‌کند. بنابراین در هر لحظه از زمان A یک زیرمجموعه از بخش‌های فایل را در اختیار می‌گیرد و می‌داند کدامیک از این بخش‌ها مربوط به همسایگان او می‌باشند.

1- Tracker



شکل ۲- ۲۶ ♦ توزیع فایل از طریق بیت ترنت (تندآب بیت)

با این اطلاعات A بایستی دو تصمیم مهم را اتخاذ نماید. نخست کدام بخش‌ها را وی بایستی ابتدا از همسایگان خود تقاضا نماید؟ و دوم به کدامیک از مجاورانش بخش‌هایی را که تقاضا و دریافت می‌کند ارسال نماید؟ در تصمیم‌گیری مربوط به بخش‌های مورد نیاز A از یک فناوری استفاده می‌کند که اصطلاحاً **نادرترین نخست**[1] نام دارد. فلسفه‌ی این رویکرد آن است که از میان بخش‌هایی از فایل که A در اختیار ندارد آن قسمت‌هایی که در بین همسایگانش از همه نادرتر است را گزینش و تقاضا نماید. به بیانی دیگر بخش‌هایی از فایل که شامل کمترین تعداد دفعات نسخه برداری در بین مجاورانش می‌باشد را درخواست کند. نتیجه‌ی این رویکرد آن است که اینگونه بخش‌های کمیاب فایل سریعتر از بقیه‌ی قسمت‌ها توزیع مجدد می‌شوند و تقریباً تعداد نسخه‌های دریافت شده از هر بخش در تندآب یکسان می‌گردد. برای پاسخ به پرسش دوم که کدامیک از تقاضاها را در اولویت پاسخگویی قرار دهد. پروتکل بیت تُرنت از روش الگوریتم معامله‌ی هوشمند استفاده می‌کند. اندیشه‌ی اصلی آن است که A همسایگانی که در حال حاضر داده‌ها را با بیشترین سرعت در اختیار او قرار می‌دهند، در صدر اولویت خود منظور نماید. بنابراین A هر یک از مجاورانش را مرتباً از نقطه‌نظر پهنای باند ارسالی آنها اندازه گیری می‌کند و از میان گروه آنها تعداد چهار نظیر را که دارای بیشترین سرعت می‌باشند شناسایی و سپس عملکرد آنها را با ارسال بخش‌های فایل از جانب خود به سمت آن چهار نظیر جبران می‌کند. معمولاً در فاصله‌ی زمانی هر 10 ثانیه مجدداً سرعت‌ها را محاسبه و ممکن است در فهرست نظیرهای چهارگانه تغییراتی را به وجود آورد. در ادبیات بیت ترنت این نظیرهای چهارگانه را اصطلاحاً **نامسدود**[2] می‌نامند. حائز اهمیت است که هر 30 ثانیه یک بار A به‌صورت تصادفی یک همسایه‌ی

1- Rarerest first
2- Unchocked

دیگر را گزینش و به گروه مجاورانش می‌افزاید و قسمت‌های در اختیار خود را به سمت آنها گسیل می‌دارد. فرض کنید همسایه‌ی تصادفی گزینش شده B باشد. باز هم در واژگان بیت ترنت این مجاور تصادفی گزینش شده را **نامسدود خوش بین**[1] می‌نامند. از آنجائیکه A در حال گسیل داده‌ها به B می‌باشد، او ممکن است تبدیل به یکی از بارگذارهای چهارگانه‌ی در صدر B شده باشد که در این صورت B ارسال داده‌ها به A را آغاز می‌کند. چنانچه سرعت ارسال B به A در حد کافی بالا باشد آنگاه B می‌تواند تبدیل به یکی از بارگذارهای چهارگانه‌ی صدر فهرست A گردد. به عبارت دیگر هر 30 ثانیه A به گونه‌ی تصادفی یک شریک جدید را برای معامله با خود گزینش می‌کند و داد و ستد خود را با آن همکار آغاز می‌کند. اگر دو نظیر از انجام معامله با یکدیگر راضی باشند، آنها یکدیگر را در صدر صورت چهارگانه‌ی نظیرها قرار داده و داد و ستد را با یکدیگر تا زمانی ادامه می‌دهند که یکی از نظیرها شریک بهتری را شناسایی کند. در نتیجه نظیرهایی که قادرند عملیات بارگذاری را با نرخ سازگار انجام دهند، سبب یافتن یکدیگر نیز می‌شوند. گزینش تصادفی همسایگان امکان می‌دهد نظیرهای جدید بخش‌های فایل را دریافت و در نتیجه همواره کالایی را برای معامله در اختیار داشته باشند. بقیه‌ی نظیرهای مجاور به جز این 5 نظیر که عبارتند از 4 نظیر صدر صورت به همراه نظیر کاوش شده به حالت مسدود در می‌آیند. به عبارت دیگر هیچگونه اطلاعاتی را از A دریافت نخواهند نمود. علاوه بر موارد عنوان شده در بالا پروتکل بیت ترنت دارای چندین مکانیزم جالب دیگر مانند: قطعه‌ها (بخش‌های کوچک)، خط لوله، گزینش نخست تصادفی، روش انجام بازی نهایی و ضد دفع می‌باشد که در این مقطع مورد بررسی قرار نمی‌گیرند.

روش معامله‌ی تشویقی که در بالا توضیح داده شد، غالباً روش این به آن در خوانده می‌شود. البته در مواردی نشان داده شده است که این روش می‌تواند حالت گیرانداز نیز داشته باشد. اما در هر حال روش بسیار موفق و دارای طرفداران بی شماری است. همواره میلیون‌ها نظیر همزمان در قالب صدها هزار تندآب فایل‌ها را به اشتراک می‌گذارند. اگر بیت ترنت براساس این به آن در طراحی نشده بود شاید امروز وجود خارجی نداشت و اکثریت کاربران از حالت آزاد برای همه استفاده می‌نمودند. پروتکل بیت ترنت در قالب کاربردهای دیگری نظیر کاربردهای زنده‌ی P2P مانند PPLive و PPStream نیز طراحی و به‌کار گرفته شده است.

۲-۱۷ جداول درهم توزیع شده

یکی از عناصر حیاتی در بسیاری از کاربردهای P2P و توزیع شده عبارت است از یک شاخص عامل کاوش و به بیانی دیگر یک پایگاه داده‌ی ساده که قادر به پشتیبانی عملیات جستجو و به روز رسانی باشد. هنگامی‌که چنین پایگاه داده یی توزیع شده باشد، نظیرها ممکن است عملیات نهان‌سازی اطلاعات و مسیریابی‌های پیشرفته برای درخواست‌ها را در بین خود انجام دهند. از آنجائیکه شاخص گذاری و جستجوی اطلاعات یکی از عناصر حیاتی در چنین سیستم‌هایی به‌شمار می‌رود، بنابراین بایستی به دنبال روش‌های مؤثری جهت تحقق این مهم بود. یکی از معروفترین روش‌های شاخص گذاری و جستجوی **جداول درهم توزیع شده**[2] (DHTs) می‌باشد.

فرض کنید قصد دارید یک پایگاه داده‌ی توزیع شده برای جمعیت انبوهی مانند میلیون‌ها نظیر

1- Optimistically Unchocked
2- Distributed Hash Tables

خلق کنید که قادر به پشتیبانی شاخص گذاری ساده و جستجو باشد. اطلاعات ذخیره شده در این پایگاه داده یا بانک اطلاعاتی شامل جفت: کلید و ارزش می‌باشد. به‌عنوان مثال کلیدها می‌توانند شماره‌ی ملی و ارزش‌ها، نام و نام خانوادگی وابسته به افراد باشد. در این حالت یک مثال از جفت کلید و ارزش می‌تواند به صورت:

(JohnnyWu , 128-554-4821) باشد، یا کلیدها می‌توانند اسامی محتوایی مثل عناوین فیلم‌های سینمایی، آلبوم یا مجموعه‌های موسیقی و یا نرم‌افزارها باشد و ارزش‌ها در این حالت آدرس‌های IP که این مجموعه‌ها در آن ذخیره شده باشند. در این حالت مثالی از جفت کلید و ارزش می‌تواند در قالب:

(Led ZeppeLinIV , 230 • 17 • 123 • 38) تعریف گردد. نظیرها با تأمین کلید به جستجوی بانک اطلاعاتی می‌پردازند. اگر جفت کلید و ارزشی در پایگاه موجود باشد که با کلید نگاشت شود، آنگاه پایگاه، جفت نگاشت شده را به نظیر کاوشگر بازگشت می‌دهد. به‌عنوان مثال اگر بانک اطلاعاتی شماره‌های ملی را به همراه نام و نام خانوادگی ذخیره نماید، یک نظیر می‌تواند به جستجوی یک شماره‌ی ملی مشخص پرداخته و بانک اطلاعات شناسه‌ی وی را بازگشت دهد. نظیرها همچنین قادرند جفت کلید و ارزش را به پایگاه داده اضافه نمایند.

طراحی و ساخت چنین بانک اطلاعاتی به کمک یک معماری مشتری- سرور که در آن همگی جفت کلید و ارزشها بر روی یک سرور متمرکز ذخیره شده باشند، بسیار ساده و امکان‌پذیر می‌باشد. این رویکرد متمرکز در سیستم‌های نخستین P2P مانند **نَپستر**[1] پیاده‌سازی شده بودند. اما در سیستم‌های توزیع شده که شامل میلیونها نظیر متصل به یکدیگر و بدون یک هماهنگ کننده‌ی مرکزی می‌باشند این رویکرد بسیار دشوار و چالش‌های فراوانی را به همراه دارد. در یک سیستم P2P معمولاً لازم است جفت‌های کلید و ارزش در بین همه‌ی نظرها توزیع گردد. زیرا هر نظیر بایستی یک زیرمجموعه‌ی کوچکی از کل جمعیت جفت‌های کلید و ارزش را اختیار داشته و آنها را نگهداری نماید. یک رویکرد خام برای ساختن یک بانک اطلاعات P2P می‌تواند با رعایت موارد ذیل محقق شود:

۱- توزیع تصادفی جفت‌های کلید و ارزش در بین نظیرها

۲- اجبار هر نظیر به نگهداری فهرستی از آدرس‌های IP متعلق به همه‌ی نظیرهای حاضر. در این رویکرد، نظیر کاوشگر می‌تواند جستجوی خود را به همه‌ی نظیرهای دیگر ارسال و نظیرهای حاملِ جفت‌های کلید و ارزش که قابل نگاشت با کلید می‌باشند، پاسخ خود را از طریق بازگشت جفت نگاشت شده‌ی خود اعلام می‌کنند. بدیهی است چنین رویکردی حالت مقیاس پذیر اختیار نمی‌کند. زیرا، هر نظیر بایستی قادر به رهگیری همه‌ی نظیرهای دیگر باشد، که ممکن است تعداد آنها به میلیون‌ها نظیر برسد. دشوارتر آنکه هر نظیر ممکن است مجبور به گسیل هر جستجویی به نظیرهای دیگر باشد.

اکنون به معرفی رویکرد ظریفی جهت طراحی یک بانک P2P می‌پردازیم. برای انجام این مهم نخست یک شناسه به هر نظیر اختصاص می‌یابد. هر شناسه یک عدد کامل در محدوده‌ی $[0 , 2^n-1]$ برای یک ارزش ثابت n می‌باشد. بدیهی است هر یک از ارزش‌های n در قالب یک نهاد n بیت تعریف می‌شوند. افزون بر مورد قبلی فرض کنید هر کلید نیز یک عدد در همان محدوده‌ی تعریف شده باشد. یک پژوهشگر زیرک ممکن است دریافته باشد که کلیدهای توصیف شده در مثال‌های گذشته یعنی شماره‌های ملی و نام محتوا هیچکدام ارزش‌های عددی کامل نبودند. برای تعریف اعداد

1- Napster

کامل از میان این کلید بایستی از یک تابع درهم‌ریزی استفاده گردد که هر کلید را به عدد کامل در محدوده‌ی [0 , 2^n-1] تبدیل نماید. کلیدی مثل شماره‌ی ملی. یک تابع درهم‌ریزی عبارت است از یک تابع که در آن دو ورودی متفاوت می‌تواند خروجی مشابه یا به عبارت دیگر عدد مشابه را اختیار کند. بدیهی است احتمال اختیار خروجی مشابه بسیار نادر است. در فصل ۷ توابع درهم‌ریزی بررسی و تحلیل شده‌اند. فرض بر آن است که تابع درهم‌ریزی به گونه‌ای همگانی در یک سیستم در اختیار همه‌ی نظیرها می‌باشد. بنابراین هنگامی‌که به واژه‌ی **کلید** اشاره می‌شود منظور ارجاع به آمیزش با کلید اصلی است.

به‌عنوان مثال اگر کلید اصلی به صورت: LedZepplinIV باشد، کلید مورد نظر عددی است که معادل آمیزه‌ی LedZepplinIV است. علاوه بر این موضوع چون از آمیزه‌های کلید به جای خود کلیدها استفاده می‌شود، این بانک اطلاعاتی توزیع شده را اصطلاحاً **جدول توزیع درهم‌ریزی** (DHT) می‌نامند.

در این مقطع به بررسی دشواری‌های ناشی از ذخیره‌ی جفت‌های کلید و ارزش در DHT می‌پردازیم. موضوع اصلی در این نقطه از بحث تعریف قاعده‌ای است که کلیدها را به نظیرها اختصاص دهد. با توجه به اینکه هر نظیر دارای یک شناسه‌ی عددی است و نیز هر کلید عددی است در همان محدوده، رویکرد طبیعی آن است که هر جفت کلید و ارزش به نظیری اختصاص یابد که شناسه‌ی آن به کلید نزدیکتر است. برای پیاده‌سازی چنین روشی نیازمند تعریف مفهوم **نزدیکتر می‌**باشیم زیرا تعبیرهای مختلف می‌توان برای آن عنوان نمود. جهت ساده‌سازی مطلب فرض کنید **نظیر نزدیکتر** را اینگونه تعریف کنیم: **جانشین بی‌درنگی از کلید**. برای درک بهتر این تعریف به این مثال دقت نمائید: فرض کنید n = 4 است بنابراین همه‌ی نظیرها و شناسه‌های کلید در محدوده‌ی [0,15] می‌باشند. علاوه بر آن فرض کنید تعداد نظیرها در سیستم برابر 8 است و شناسه‌های آنها نیز عبارتند از: 1، 3، 4، 5، 8، 10، 12 و 15 و سرانجام فرض کنید می‌خواهیم جفت (11, Johnny Wu) در یکی از 8 نظیر ذخیره کنیم. اما در کدام نظیر؟ با استفاده از مفهوم نظیر نزدیکتر از آنجائیکه نظیر 12 جانشین بی‌درنگ کلید 11 می‌باشد در نتیجه جفت کلید و ارزش نامبرده در نظیر 12 ذخیره می‌شوند. برای تکمیل تعریفی که از واژه‌ی نزدیکتر نمودیم به بیان این مطلب می‌پردازیم که اگر کلید عیناً برابر است با یکی از شناسه‌های نظیرها، ذخیره‌ی جفت کلید و ارزش در آن نظیر انجام می‌شود و اگر کلید بزرگتر از همه‌ی شناسه‌ها می‌باشد از قرارداد **پیمانه‌ی - 2^n** [1] جهت ذخیره‌سازی جفت کلید و ارزش در نظیری که دارای کوچکترین شناسه است، استفاده می‌شود.

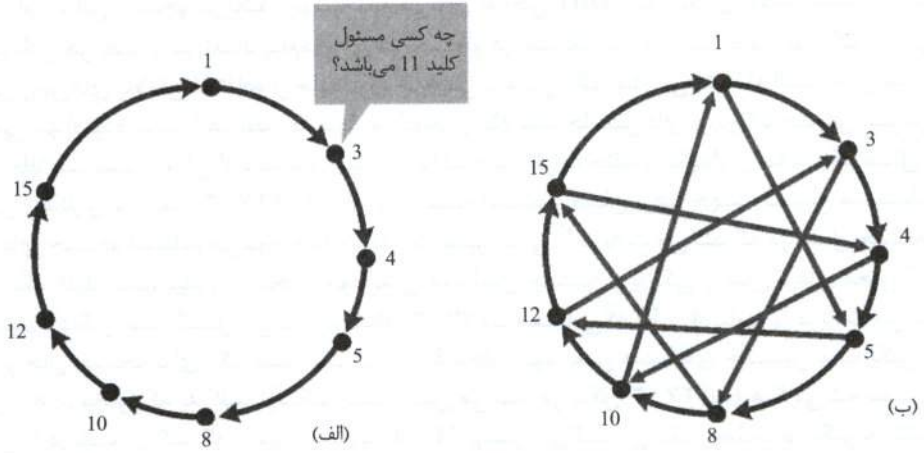
اکنون فرض کنید یک نظیر A مایل به اضافه نمودن یک جفت کلید و ارزش در DHT می‌باشد. از لحاظ مفهوم بسیار روشن است: A نخست نظیری را شناسایی می‌کند که شناسه‌ی آن با کلید نزدیکتر است. او سپس پیامی را به آن نظیر ارسال می‌کُند و از او می‌خواهد که جفت کلید و ارزش را ذخیره نماید. اما A چگونه متقاعد شده است که آن نظیر به کلید نزدیکتر است؟ اگر A مجبور به رهگیری همه‌ی نظیرها در سیستم باشد، به گونه‌ی محلی می‌تواند نزدیکترین نظیر را شناسایی کند اما چنین رویکردی نیازمند آن است که هر نظیر اطلاعات رهگیری همه‌ی نظیرهای دیگر را در DHT قرار دهد. با اندکی تأمل ملاحظه خواهد شد که چنین رویکردی در یک سیستم پایگاه داده‌ی بسیار بزرگ که ممکن است شامل میلیونها نظیر باشد در عمل غیر ممکن است.

1- Modulo-2^n

۲- ۱۸ جدول توزیع آمیزش چرخشی[1]

چالش مقیاس پذیری که در مطالب گذشته به آن اشاره شد از طریق مدورسازی نظیرها برطرف می‌گردد. اگر نظیرها در قالب مدور با یکدیگر تعامل نمایند آنگاه هر نظیر تنها به رهگیری جانشین بی‌درنگ خود که همان پیمانه‌ی 2^n- است، می‌پردازد. مثالی از این سازماندهی مدور در شکل ۲- ۲۷ (الف) نشان داده شده است. در این مثال n برابر 4 فرض شده است و تعداد نظیرها هم 8 نظیر قبلی است که قبلاً به آنها اشاره گردید. هر نظیر تنها از همانند جانشین بی‌درنگ خود آگاهی دارد. به‌عنوان مثال نظیر 5 آگاهی از آدرس IP و شناسه‌ی همانند 8 دارد، اما لزوماً هیچگونه آگاهی در مورد بقیه‌ی نظیرهای احتمالاً حاضر در جدول DHT ندارد.

این چیدمان مدور نظیرها یک روش ویژه‌ای از یک نوع **شبکه‌ی جایگذاری**[2] است. در یک شبکه‌ی جایگذار، نظیر تشکیل یک شبکه‌ی منطقی مجرد را می‌دهد که بر فراز یک شبکه‌ی کامپیوتری لایه‌ی زیرین شامل پیوندهای فیزیکی، مسیریاب‌ها و میزبان قرار می‌گیرد. پیوندهای شبکه‌ی جایگذار فیزیکی نیستند اما به نوعی رابطهای ساده‌ی منطقی بین جفت‌های نظیر می‌باشند. در شبکه‌ی جایگذار شکل ۲- ۲۷ (الف) تعداد 8 نظیر با 8 پیوند جایگذار دیده می‌شود. در شکل ۲- ۲۷ (ب) شبکه‌ی جایگذار دارای تعداد 8 همانند و 16 پیوند جایگذار است. هر پیوند جایگذار می‌تواند از تعداد زیادی پیوندها و مسیریاب‌های فیزیکی حاضر در یک شبکه‌ی لایه زیرین بهره مند باشد. با استفاده از شبکه‌ی جایگذاری چرخشی شکل ۲- ۲۷ (الف) اگر فرض گردد نظیر 3 مایل است بداند کدامیک از همانندها در DHT مسئول کلید 11 می‌باشند؟ این مهم را می‌تواند از طریق تزریق و یا جستجوی یک جفت کلید و ارزش انجام دهد. با استفاده از جایگذاری چرخشی نظیر 3 (مبداء) مبادرت به خلق یک پیام می‌کند که مضمون آن عبارت: "چه کسی مسئول کلید 11 است؟" می‌باشد و سپس آن را به همانند جانشین خود یعنی 4 گسیل می‌دارد.



شکل ۲- ۲۷ ♦ (الف) یک **DHT** چرخشی. نظیر 3 مایل به آگاهی از نظیر با کلید 11 است. (ب) یک **DHT** چرخشی با میان بُر

1- Circular DHT
2- Overlay network

هنگامی‌که نظیری چنین پیغامی را دریافت نماید، چون شناسه‌ی جانشین خود را می‌شناسد، می‌تواند تعیین کند که آیا آن نظیر مسئول کلید مورد نظر است یا خیر؟ اگر یک نظیر مسئول کلید نمی‌باشد، پیام را به جانشین خود ارجاع می‌دهد. به‌عنوان مثال هنگامی‌که نظیر 4 پیام را دریافت می‌کند که از وی در مورد کلید 11 پرسش کرده است، او مشخص می‌کند که مسئول کلید نیست زیرا جانشین 4 یعنی 5 به کلید نزدیکتر است. بنابراین پیام را به نظیر جانشین خود 5 انتقال می‌دهد. این فرآیند تا آنجا ادامه می‌یابد که پیام سرانجام به نظیر 12 می‌رسد و او در نهایت مشخص می‌کند که وی نزدیکترین نظیر به کلید 11 می‌باشد. در این مقطع نظیر 12 می‌تواند پیامی را به همانند 3 بازگشت دهد و در آن پیام خود را مسئول کلید 11 اعلام کند.

رویکرد DHT مُدور راهکار هوشمندانه‌ای را فراهم می‌آورد که نتیجه آن کاهش میزان اطلاعات جایگذاری شده‌ای است که هر یک از نظیرها بایستی مدیریت نمایند. به‌ویژه آنکه هر نظیر تنها آگاه از وضعیت دو همانند دیگر است. نظیرهایی که بی‌درنگ جانشین و یا سلف او می‌باشند. بدیهی است در عمل همواره یک نظیر آگاه از وضعیت سلف خود می‌باشد زیرا این نظیر پیشین است که پیام را برای او ارسال می‌کند. اما حتی این راهکار اخیر نیز سبب تولید چالش جدیدی می‌شود. اگرچه هر نظیر تنها آگاه از دو همانند مجاورش می‌باشد که از طریق آنها می‌تواند گره‌ی مسئول یک کلید را بیابد. اما همگی تعداد N گره در DHT بایستی پیام را در پیرامون یک دایره به حرکت درآورند و مفهوم این موضوع آن است که به گونه‌ای میانگین تعداد پیام‌های ارسالی $\frac{N}{2}$ است. بنابراین در طراحی DHT یک توافق دو طرفه بین تعداد همسایگانی که هر نظیر بایستی رهگیری کند و تعداد پیام‌هایی که DHT نیازمند به ارسال آنها جهت یافتن پاسخ برای یک جستجو می‌باشد، همواره وجود دارد. از یک طرف اگر هر نظیر همه همانندهای دیگر را به‌صورت **جایگذاری تنیده**[1] رهگیری نماید. آنگاه فقط یک پیام برای جستجو ارسال شده است، در صورتی‌که هر نظیر مجبور به نگهداری اطلاعات N همانند است. از طرف دیگر در روش DHT مُدور هر نظیر تنها آگاه از وضعیت دو همانند است، که نتیجه‌ی آن تعداد $\frac{N}{2}$ پیام ارسالی جستجو می‌باشد. خوشبختانه می‌توان طراحی DHT را به گونه‌ای پالایش نمود که تنها همسایگان هر نظیر و نیز تعداد پیام‌ها برای هر جستجو در حد یک اندازه‌ی قابل قبول قرار گیرد. یکی از این روش‌های پالایش استفاده از جایگذاری چرخشی به‌عنوان یک اصل می‌باشد که البته به آن میان بُرهایی نیز افزوده شده تا هر نظیر نه تنها به نگهداری اطلاعات جانشین‌های بی‌درنگ خویش بپردازد بلکه اطلاعات نسبتاً اندکی از نظیرهای میان بُر پراکنده در اطراف حلقه را نگهداری نماید. یک مثال از چنین راهکاری در شکل ۲- ۲۷ (ب) نشان داده شده است. از میان بُرها جهت تسریع مسیریابی پیام‌های جستجو استفاده می‌شود. هنگامی‌که یک نظیر پیامی را دریافت می‌کند که در حال جستجو برای یک کلید است، پیام را به مجاور خود یعنی همسایه‌ی جانشین و یا یکی از میان بُرهای مجاور که به کلید نزدیکتر است گسیل می‌کند. در شکل ۲- ۲۷ (ب) هنگامی‌که نظیر 4 پیامی را دریافت می‌کند که در حال جستجو برای یک کلید است، پیام را به مجاور خود یعنی همسایه‌ی جانشین و یا یکی از میان برهای مجاور که به کلید نزدیکتر است گسیل می‌کند. در شکل ۲- ۲۷ (ب) هنگامی‌که نظیر 4 پیامی را دریافت می‌کند که از وی در مورد کلید 11 پرسش می‌کند، آن نظیر همانند نزدیکتر به کلید 11.را از میان همسایگان خود شناسایی می‌کند. در این مثال نظیر 10 که یک مجاور میان بُر می‌باشد گزینش شده و پیام مستقیماً به سوی آن گسیل می‌شود. روشن است که استفاده از میان بُرها سبب کاهش چشمگیر تعداد پیام‌های استفاده شده جهت پردازش یک جستجو می‌باشد.

1- Mesh overlay

دومین پرسش تعیین تعداد مجاوران میان بُری است که هر نظیر می‌تواند اختیار نماید، و نیز کدامیک از نظیرها بایستی همسایگان میان بُر آن باشند؟ در بین پژوهشگران اینترنت یافتن پاسخ برای این پرسش‌ها علاقه مندان بسیاری را به خود مشغول داشته است. از همه مهمتر آنکه نشان داده شده است طراحی DHT را می‌توان به گونه‌ای انجام داد که تعداد مجاوران در هر نظیر و تعداد پیام‌ها در هر جستجو برابر O(logN) باشد. در این رابطه N تعداد نظیرها است. گزینش چنین رویکردی سبب ایجاد یک توازن مطلوب بین دو راهکار دشوار و تقریباً غیر ممکن جایگذاری‌های تنیده و دوار می‌شود.

۲- ۱۹ تحرکات نظیر[1]

در سیستم‌های P2P یک قرین می‌تواند بدون اعلام قبلی وارد و یا خارج شود. بنابراین هنگام طراحی یک DHT، طراح بایستی نگران جایگذاری DHT با آگاهی از چگونگی تحرکات نظیر باشد. برای آنکه تصویر بزرگتری از این موضوع بدست آید. بار دیگر DHT چرخشی در شکل ۲- ۲۷ (الف) بررسی می‌شود. برای کنترل تحرکات قرین‌ها از هر یک از آنها خواسته می‌شود که به رهگیری اطلاعات نخستین و دومین جانشین خود پردازند. به بیان دیگری از آدرس‌های IP این دو جانشین آگاهی پیدا کنند. در مثال شکل ۲- ۲۷ (الف)، نظیر 4 به رهگیری قرین‌های 5 و 8 می‌پردازد. افزون بر این مورد هر نظیر مجبور باشد به گونه‌ی متناوب ارزیابی از حضور و یا زنده بودن قرین‌های جانشین خود را نیز انجام دهد. به‌عنوان مثال متناوباً پیام‌های پینگ به آنها گسیل و پاسخ‌های آنها را طلب نماید. اکنون وضعیت نگهداری DHT را به هنگام ترک ناگهانی یک نظیر از سیستم P2P بررسی می‌کنیم. به‌عنوان مثال اگر قرین 5 در شکل ۲- ۲۷ (الف) به طور ناگهانی حلقه را ترک نماید، در این حالت دو نظیر قبل آن یعنی 4 و 3 آگاه می‌شوند که ۵ حلقه را ترک گفته است. زیرا او به پیام‌های پینگ ارسالی از سوی آنها پاسخ نمی‌دهد. در نتیجه همتاهای 4 و 3 نیازمند به روزرسانی وضعیت اطلاعات جانشین‌های خود می‌شوند. پرسش مطرح آن است که چگونه نظیر 4 وضعیت خود را به روز رسانی می‌کند؟ پاسخ:

۱- نظیر 4 جانشین نخست خویش یعنی همتای 5 را با دومین جانشین خود یعنی قرین 8 جایگزین می‌کند.

۲- قرین 4 آنگاه از جانشین نخست جدید خود یعنی نظیر 8 می‌خواهد که او شناسه و آدرس IP جانشین بی‌درنگ خود را که همتای 10 می‌باشد در اختیار نظیر 4 بگذارد. همتای 4 آنگاه قرین 10 را به‌عنوان جانشین دوم خود گزینش می‌کند.

در این توصیف خلاصه مشخص گردید، هنگامی‌که یک نظیر سیستم P2P را که از روش DHT استفاده می‌کند، ترک نماید، راهکار مناسب چیست. اکنون حالتی را در نظر بگیرید که یک نظیر می‌خواهد به DHT اضافه شود. فرض کنید نظیری با شناسه‌ی 13 مایل است به DHT ملحق شود و در زمان پیوستن تنها در مورد حضور و وجود قرین 1 در DHT آگاهی دارد. نظیر 13 ابتدا پیامی را برای قرین 1 می‌فرستد و در آن پیام پرسش می‌کند: "جانشین و سلف همتای 13 کیست؟" این پیام از درون DHT تا دریافت آن به‌وسیله‌ی نظیر 12 به جلو رانده می‌شود. همتای 12 می‌داند او سَلف 13 است و جانشین جاری آن یعنی نظیر 15 به‌عنوان جانشین همتای 13 گزینش خواهد شد. در این مرحله قرین 12 از طریق یک پیام اطلاعات سلف و جانشین خود را به همتای 13 گسیل می‌کند. نظیر 13 اکنون با آرامش خیال به DHT می‌پیوندد و قرین 15 را به‌عنوان جانشین خود گزینش می‌کند و در عین حال با

1- Peer churn

ارسال پیامی به همتای ۱۲ از او می‌خواهد که با انجام تغییر، نظیر 13 را به‌عنوان جانشین بی‌درنگ خود منظور نماید.

در حال حاضر استفاده از کاربرد DHT در شبکه‌ی اینترنت بسیار گسترده می‌باشد. به‌عنوان مثال نرم‌افزار بیت تُرنت از DHT طراحی شده‌ی **کادملیا**[1] برای ایجاد یک رهگیر توزیع شده استفاده می‌کند. در بیت ترنت کلید شناسه‌ی تُرنت (تندآب) و ارزش آدرس‌های IP عبارت است از همه‌ی نظیرهای جاری در تندآب، به این ترتیب از طریق شناسه‌ی تُرنت می‌توان از DHT جهت جستجوی یک همتای جدید وارد شده به بیت تُرنت، بهره جست و سپس، از آن خواستار شناسایی نظیری شد که مسئول شناسه‌ی رهگیر همتاها در تُرنت می‌باشد. پس از یافتن آن قرین، آنگاه نظیر وارد شده می‌تواند به جستجوی DHT برای صورتی از همتاها بپردازد که در ترنت حضور دارند. جدول‌های DHT به گونه‌ای گسترده در سیستم‌های اشتراک فایل eMule جهت مکان یابی محتوای نظیرها استفاده می‌شود.

۲- ۲۰ موضوع مطالعاتی: تلفن اینترنتی P2P و Skype

در حال حاضر یکی دیگر از کاربردهای محبوب P2P در سراسر گیتی Skype می‌باشد. در اغلب مواقع بین 7 تا 8 میلیون کاربر همزمان به آن متصل می‌شوند. علاوه بر آنکه خدمات تلفن اینترنتی PC به PC را فراهم می‌آورد، اسکایپ امکاناتی چون PC تلفنی به گوشی تلفن، گوشی تلفن به PC تلفنی و کنفرانسهای تصویری PC به PC را نیز پشتیبانی می‌کند. اگرچه بنیانگذاران آن همان افرادی بودند که کاربردهای کازا (Kazaa) و پیگردی سریع (Fast Track) را طراحی نمودند، اما اسکایپ در حال حاضر و در حقیقت از سال 2005 میلادی با هزینه‌ی 2.6 میلیارد دلار به مالکیت شرکت ebay درآمد. در ماه مارس 2011 شرکت مایکروسافت آن را به مبلغ 8.5 میلیارد دلار خریداری نمود.

نرم‌افزار کاربردی اسکایپ از پروتکل P2P به گونه‌ی خلاق که نشان می‌دهد چگونه P2P می‌تواند در کاربردهایی که فراتر از توزیع محتوا و یا اشتراک فایل قرار دارند به‌صورت گسترش یافته‌ای مورد استفاده قرار گرفته است. کاربردهایی چون پیام رسانی بی‌درنگ، تلفن اینترنتی PC به PC مانند انتقال پیام بی‌درنگ از ویژگی‌های ذاتی P2P می‌باشند زیرا در هسته‌ی کاربرد خود، شامل جفت کاربرانی (نظیرهایی) می‌باشند که با یکدیگر به شکل زمان واقعی ارتباط برقرار می‌کنند. اما علت بکارگیری اسکایپ از روش‌های P2P به علت دو عملکرد مهم دیگر نیز می‌باشد. این دو عبارتند از: (۱) مکان یابی کاربر و (۲) پیمایش ترجمه‌ی آدرس شبکه و به بیان دیگری پیمایش NAT.

نه تنها پروتکل‌های اسکایپ خصوصی هستند بلکه تمامی بسته‌های در حال انتقال اسکایپ شامل تصاویر و بسته‌های کنترل نیز رمزگذاری می‌شوند. در هر حال با توجه به اطلاعات موجود در سامانه‌ی اینترنت Skype و برخی اندازه گیری‌های مطالعاتی، پژوهشگران در یک قالب کلی دریافته‌اند که چگونه نرم‌افزار اسکایپ عمل می‌کند. مشابه Fast Track گره‌ها در اسکایپ به گونه‌ی شبکه‌ی جایگذار سلسله مراتبی سازماندهی می‌شوند. هر نظیر در دو قالب اَبر همتا و قرین معمولی نامگذاری می‌شود. نرم‌افزار اسکایپ همچنین از یک شاخص که نام کاربران خود را به آدرس‌های IP جاری و یا به شماره پورت‌ها نگاشت می‌کند، بهره می‌گیرد. این شاخص بر روی اَبر نظیرها توزیع می‌گردد. هنگامی‌که A مایل به انجام مکالمه با B می‌باشد، برنامه مشتری اسکایپ متعلق به آن به جستجوی شاخص توزیع شده برای یافتن آدرس IP جاری متعلق به B می‌پردازد. از آنجائیکه پروتکل اسکایپ خصوصی و انحصاری است و

1- Kademlia DHT

نه همگانی، در حال حاضر روشن نیست، چگونه نگاشت شاخص در اَبَر نظیرها انجام می‌شود، اگرچه استفاده از نوعی سازماندهی DHT دور از انتظار نیست. از روش‌های P2P همچنین در تکرارکننده‌های اسکایپ نیز استفاده می‌شود که می‌تواند برای ایجاد ارتباط بین میزبان‌ها در شبکه‌های خانگی سودمند باشد. بسیاری از این شبکه‌ها از پیکربندی دستیابی به اینترنت از طریق یک مسیریاب و به‌ویژه مسیریاب‌های بدون سیم استفاده می‌کنند. این مسیرها در واقع از لحاظ امکانات فراتر از روترهای معمولی شبکه می‌باشند. زیرا از پروتکل تبدیل آدرس شبکه (NAT) نیز بهره می‌گیرند. موضوع NAT در فصل ۴ بررسی شده است. آنچه در این مقطع در مورد NAT لازم است، بیان شود آن است که یک NAT از ارتباط یک میزبان خارج از شبکه‌ی خانگی به میزبانی در درون آن شبکه جلوگیری می‌کند. اگر هر دو کاربر اسکایپ از NAT بهره گیرند، آنگاه هیچیک از آنها نمی‌توانند ارتباط تقاضا شده با دیگری را پذیرش نمایند. در نتیجه ارتباط غیر ممکن به نظر می‌رسد. استفاده‌ی زیرکانه از اَبَر نظیرها و نیز تکرار کننده‌ها راهکار این مشکل می‌باشد. فرض کنید هنگامی‌که A وارد فرآیند ارتباط می‌شود یک اَبَرنظیر غیر NAT را به خود اختصاص دهد. بنابراین A ارتباطی را با اَبَرهمتای خود برقرار می‌کند، زیرا NAT متعلق به او اجازه‌ی ارتباط خارج از شبکه‌ی خانگی را به او نمی‌دهد. در حالیکه به نظر می‌رسد چیدمان داخلی می‌باشد. در هر حال A اجازه می‌یابد به همراه اَبَرنظیر خود به مبادله‌ی پیام‌های کنترل از طریق این تعامل بپردازد. هنگامی‌که B نیز وارد فرآیند شود مشابه این عمل برای وی به وقوع می‌پیوندد. بنابراین در حال حاضر که A مایل به گفتگوی با B می‌باشد، A ابتدا اَبَرقرین خود را آگاه می‌کند و آن نظیر نیز اَبَر قرین B را و سپس آن همتا B را از ورود ارتباط A آگاه می‌سازد. اگر B ارتباط را قبول نماید، آنگاه دو اَبَرنظیر به گزینش یک اَبَرقرین غیر NAT سوم می‌پردازند. این همتای سوم اصطلاحاً گره‌ی تکرار کننده نام دارد که وظیفه‌ی آن انتقال و یا به عبارت دیگر تکرار داده‌ها بین A و B می‌باشد. اَبَرنظیرهای A و B آنگاه به ترتیب A و B را مجبور به آغاز یک تعامل با تکرار کننده می‌کنند. سپس A بسته‌های صوتی را از طریق ارتباط A و تکرار کننده‌ای که A آغازگر آن بوده است به تکرار کننده گسیل می‌کند و آن ارتباط این بسته‌ها را به ارتباط تکرار کننده و نیز B که آغازگر آن بوده است ارسال می‌کند. بسته‌ها از B به A نیز در جهت عکس از طریق همان پیوندهای ارتباطی گسیل می‌شوند. سرانجام A و B دارای یک ارتباط انتها به انتها براساس تقاضای بی‌درنگ می‌شوند. اگرچه هیچیک از آنها نمی‌تواند ارتباطی را که سرچشمه‌ی آن از شبکه‌ی LAN خارج از آنها باشد پذیرش نماید. استفاده از تکرار کننده‌ها نشان پیچیدگی و ظرافت‌های طراحی در سیستم‌های P2P می‌باشد، که نظیرها از طریق آنها به انجام عملیات خدمات هسته‌ی سیستم برای دیگران می‌پردازند. خدمات شاخص گذاری یا جستجو و تکرار از جمله‌ی این موارد به‌شمار می‌آیند. این در حالی است که آنها خود از خدمات انتهایی مثل بارگیری پائین فایل و یا تلفن IP که سیستم P2P فراهم می‌آورد نیز بهره می‌جویند.

پروتکل اسکایپ کاربردی سراسر موفقیت آمیز را در اینترنت داشته است. ده‌ها میلیون کاربر روزانه از خدمات آن بهره می‌گیرند و هنگام اوج ترافیک این تعداد به 23 میلیون کاربر همزمان می‌رسد. گزینش گسترده و نفس گیر کاربردهایی مانند اسکایپ، اشتراک فایل P2P، وب و انتقال پیام بی‌درنگ همگی مؤید این خرد جمعی است که ساختار کلی و معماری طراحی اینترنت یک فرآیند غیر قابل پیش‌بینی را از لحاظ مجموعه‌ای از کاربردهای اینترنتی غنی و قابل گسترش در پیش رو می‌گذارد که در طی 30 سال آینده امکان طراحی، توسعه و بهره وری از آنها در افق روشنی خودنمایی می‌کند. خدمات ارائه شده به کاربردهای اینترنت مانند انتقال غیر اتصال گرای داده گرام‌ها (UDP)، انتقال

اتصال‌گرا و مطمئن داده گرام‌ها (TCP)، واسط‌های سوکت، آدرس دهی، خدمات DNS و بسیاری دیگر گواهی بر این واقعیت می‌باشند که هزاران کاربرد از طریق آنها طراحی و گسترش یافته است. از آنجائیکه لایه‌بندی همگی این کاربردها بر فراز چهار لایه‌ی زیرین موجود در مجموعه‌ی پشته‌ی پروتکلی اینترنت بنا شده است. این رویکرد باعث خلق نرم‌افزارهای جدید نظیر به نظیر و مشتری-سرور برای کاربرد در سیستم‌های انتهایی شده است و به دنبال خود بکارگیری سریع و گزینش گسترده‌ی اینگونه کاربردها را به همراه داشته است.

۲-۲۱ برنامه‌نویسی سُوکت با TCP

اکنون که با بررسی تعدادی از کاربردهای مهم شبکه به اهمیت لایه‌ی کاربرد پی بردیم، در ادامه قصد بر آن است که چگونگی طراحی و نگارش واقعی برنامه‌های کاربردی شبکه را کاوش و پی گیری نمائیم. در این بخش از مطالب فصل تلاش می‌کنیم برنامه‌های کاربردی را به تحریر درآوریم که پروتکل TCP از آنها استفاده می‌کند. در بخش بعدی همین مطلب برنامه‌های کاربردی که از UDP استفاده می‌کنند، معرفی خواهند شد. قبلاً در بخش ۲- ۱ اشاره گردید، بسیاری از کاربردهای شبکه شامل یک جفت برنامه است. یک برنامه‌ی مشتری و دیگری سرور، هر دو برنامه در دو سیستم انتهایی متفاوت ساکن می‌باشند. هنگامی‌که این دو برنامه اجرا می‌شوند، یک فرآیند مشتری و سرور را خلق می‌کنند. این فرآیندها از طریق خواندن و یا نوشتن در سوکت‌ها با یکدیگر تعامل برقرار می‌کنند. هنگامی‌که یک کاربرد شبکه در حال تعریف است، طراح وظیفه‌ی اصلی خود را نگارش کُدها برای برنامه‌های سرور و مشتری می‌داند. اساساً دو نوع کاربرد شبکه وجود دارد. یک گروه پیاده‌سازی یک پروتکل استاندارد تعریف شده در یک سند مانند یک RFC می‌باشد که برای چنین پیاده‌سازی برنامه‌های سرور و مشتری بایستی سازگار و قابل قبول برای قواعدی باشند که به‌وسیله‌ی RFC رعایت آنها الزامی می‌باشد. به‌عنوان مثال برنامه‌ی مشتری می‌تواند یک پیاده‌سازی از سمت مشتری پروتکل FTP باشد که در RFC959 به گونه‌ی ویژه و انحصاری تعریف شده است. یا به طور مشابه برنامه‌ی سرور می‌تواند یک پیاده‌سازی از پروتکل سرور FTP باشد که آن هم به گونه‌ی ویژه و انحصاری در همان RFC تعریف شده است. اگر یک طراح کُدهایی را برای برنامه‌ی مشتری و یک طراح مستقل دیگر کُدهایی را برای برنامه سرور بنویسند و هر دو قواعد تعریف شده در RFC را با دقت دنبال نمایند آنگاه این دو برنامه قادرند با یکدیگر ارتباط برقرار نمایند. یقیناً امروز بسیاری از کاربردهای شبکه نیازمند تعامل بین دو برنامه‌ی مشتری و سرور می‌باشند که در اغلب موارد به‌وسیله‌ی برنامه نویسان مستقل طراحی شده‌اند. به‌عنوان مثال یک مرورگر **فایر فاکس**[1] با یک وب سرور آپاچی ارتباط برقرار می‌کند و یا یک مشتری FTP در حال اجرا روی یک PC بارگذاری یک فایل را به یک سرور FTP سیستم عامل لینوکس انجام می‌دهد. هنگامی‌که یک برنامه‌ی مشتری یا سرور پروتکلی را که از طریق یک RFC تعریف شده است پیاده‌سازی می‌کند، آن برنامه بایستی از شماره پورت وابسته به آن پروتکل استفاده نماید. قبلاً به طور خلاصه در بخش ۲-۱ شماره پورت‌ها بررسی و تحلیل گردیدند.

گروه دوم از کاربردهای شبکه برنامه‌های خصوصی و انحصاری می‌باشند. در این حالت پروتکل لایه‌ی کاربرد که دو برنامه‌ی سرور و مشتری از آن استفاده می‌کنند لزوماً نیازی به سازگاری با هیچیک از RFCهای موجود ندارد. یک برنامه نویس تنها و یا یک گروه از برنامه نویسان هر دو برنامه‌ی مشتری

1- Firefox

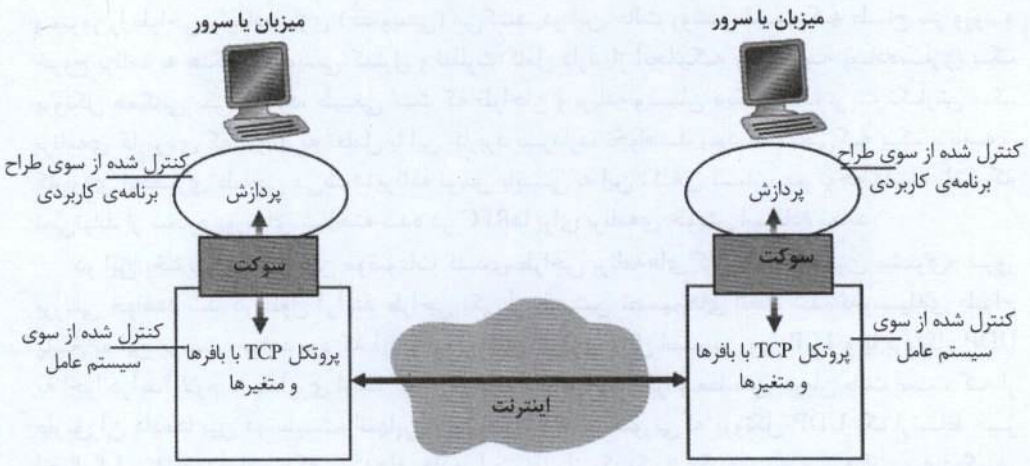
و سرور را طراحی و پیاده‌سازی (کُدنویسی) می‌کنند. در این حالت روشن است که طراح بر ورود و خروج برنامه به هنگام کُدنویسی کنترل و نظارت کامل دارد. از آنجائیکه برنامه به پیاده‌سازی یک پروتکل همگانی نمی پردازد، طبیعی است که طراحان و برنامه‌نویسان دیگری قادر به نگارش یک برنامه‌ی کاربردی که بتواند به تعامل با این کاربرد بپردازد، نخواهند بود. هنگامی‌که یک برنامه‌ی کاربردی انحصاری طراحی می‌شود، برنامه نویس بایستی به این نکته‌ی بسیار مهم توجه داشته باشد که نمی‌تواند ار شماره پورتهای شناخته شده در RFCها برای برنامه‌ی خویش استفاده نماید.

در این بخش و به دنبال آن موضوعات کلیدی طراحی برنامه‌های کاربردی انحصاری مشتری- سرور بررسی خواهند شد. در طول فرآیند طراحی یکی از نخستین تصمیم‌های اتخاذ شده به‌وسیله‌ی طراح پاسخ به این پرسش خواهد بود که آیا برنامه‌ی کاربردی وی مایل است بر روی TCP و یا پروتکل UDP به اجرا درآید؟ لازم به یادآوری است که TCP یک ارتباط اتصال‌گرا و مطمئن جریان بایت است که از طریق آن داده‌ها بین دو سیستم انتهایی جاری می‌شوند. در صورتی‌که پروتکل UDP یک ارتباط غیر اتصال‌گرا و نامطمئن است که بسته‌های داده را مستقل از یکدیگر از یک سیستم انتهایی به دیگری ارسال می‌کند. در این بخش برنامه‌های کاربردی ساده‌ی مشتری- سرور که بر روی هر دو پروتکل TCP و UDP قابلیت اجرا دارند، طراحی و کُدنویسی می‌شوند. این برنامه‌ها به کمک زبان برنامه‌نویسی جاوا به نگارش در می‌آیند اگرچه آنها را می‌توان با استفاده از زبان‌های برنامه‌نویسی دیگر مانند C، C^{++} و غیره نیز به تحریر درآورد. علت گزینش جاوا به‌عنوان زبان برنامه‌نویسی کاربردها نظم در ترتیب کُدنویسی و کاهش تعداد سطر کُدها می‌باشد. از همه مهمتر توصیف آسان و ساده‌ی خطوط برنامه‌های جاوا برای آن دسته از افرادی است که از لحاظ برنامه‌نویسی در شرایط بسیار ابتدایی می‌باشند. در هر حال اگر طراحی به یک زبان برنامه‌نویسی آگاهی نسبی داشته باشد، دشواری چندانی را از لحاظ دنبال نمودن مفاهیم جاوا احساس نخواهد نمود.

با یادآوری این نکته از بخش ۲- ۱ که فرآیندهای در حال اجرا بر روی سیستم عامل‌های مختلف ارتباط با یکدیگر را از طریق ارسال پیام به درون سوکت‌ها انجام می‌دهند. گفته شد هر پردازش مشابه یک خانه و هر سوکت مربوط به آن پردازش مانند یک در ورود و خروج می‌باشد. همانگونه که از شکل ۲- ۲۸ می‌توان ملاحظه نمود سوکت همانند درِ ورود و خروج بین کاربرد و TCP می‌باشد. طراح یک کاربرد کنترل هر موضوعی را در سمت لایه‌ی کاربرد سوکت در اختیار دارد اگرچه او دارای کنترل به مراتب کمتری در سمت لایه‌ی انتقال است. بدیهی است حداکثر کنترل او بر روی این لایه می‌تواند اصلاح و اعمال تغییر چند عامل TCP مانند حداکثر اندازه‌ی حجم بافر و یا قطعه باشد.

اکنون نگاه نزدیکتری به تعامل بین برنامه‌های مشتری و سرور می‌اندازیم. وظیفه‌ی مشتری پیش قدم شدن جهت ارتباط با سرور است. از طرفی برای آنکه سرور قادر به پاسخ مناسب به مشتری باشد، نیازمند ابزار لازم جهت آماده‌سازی خود می‌باشد. این به مفهوم دو نکته است. قبل از آنکه مشتری در ارتباط پیش قدم شود، بایستی در حال اجرا باشد، نکته‌ی دوم آنکه برنامه‌ی سرور بایستی دارای نوعی دریچه‌ی ورود باشد. به عبارت دیگر یک سوکت که نسبت به پذیرش تقاضای ارتباط از سوی پردازش در حال اجرای مشتری بر روی یک میزبان دلخواه اقدام نماید. با استفاده از مثال، خانه و در برای یک پردازش و سوکت، پیش قدمی مشتری برای تماس با سرور اصطلاحاً کوبیدن بر **در خوش آمدگو**[1] نامیده می‌شود.

1- Knocking on the welcoming door



شکل ۲- ۲۸ پردازش‌های ارتباطی درون سوکت‌های **TCP**

هنگامی‌که پردازش سرور در حال اجرا می‌باشد، برنامه‌ی مشتری می‌تواند جهت ایجاد و ارتباط TCP با آن پیش قدم شود. این مهم در برنامه‌ی مشتری از طریق ایجاد یک سوکت امکان‌پذیر می‌شود. هنگامی‌که مشتری به خلق سُوکت می‌پردازد، آدرس پردازش سرور را نیز مشخص می‌کند. به عبارت دیگر آدرس IP سرور میزبان و شماره‌ی پورت پردازش سرور. بلافاصله پس از ایجاد سوکت در برنامه‌ی مشتری، TCP در مشتری اقدام به دست‌دهی سه جانبه می‌کند و ارتباط TCP با سرور ایجاد می‌شود. دست‌دهی سه جانبه در لایه‌ی انتقال انجام می‌شود و از برنامه‌های مشتری و سرور دیگر کاملاً متمایز می‌باشد.

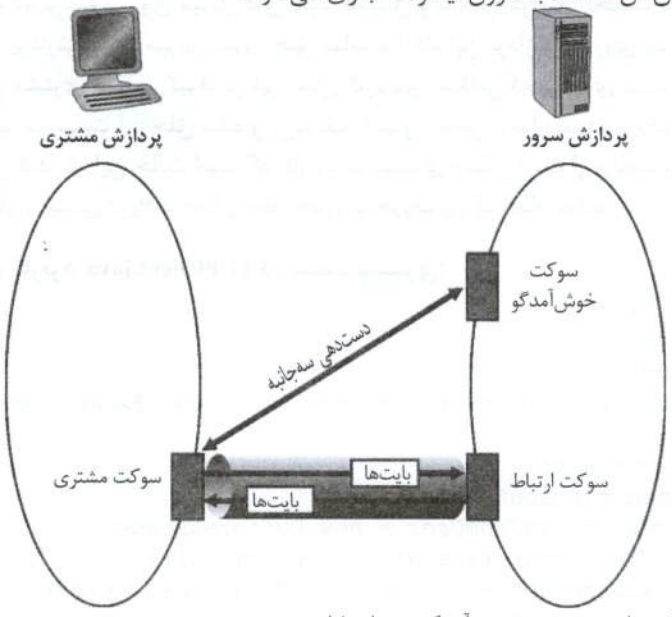
در طول انجام دست‌دهی سه جانبه، پردازش مشتری بر روی در خوش آمدگو که همان پردازش سرور است، می‌کوبد. هنگامی‌که سرور صدای کوبیدن را شنید، یک در تعریف می‌کند. به عبارت دیگر به تعریف یک سوکت جدید می‌پردازد که به این مشتری ویژه اختصاص داده می‌شود. در نمونه مثال ذیل در خوش آمدگو عنصری از Server socket است که آن را welcome socket می‌نامیم. هنگامی‌که یک مشتری بر روی در می‌کوبد برنامه به فراخوانی تابع: () welcome socket's accept می‌پردازد، که یک سوکت جدید (درِ جدید) را برای مشتری خلق می‌کند. در انتهای فرآیند دست‌دهی، یک ارتباط TCP بین سوکت و مشتری و سوکت جدید، سرور به وجود می‌آید. این سوکت جدید و اختصاصی **سوکت ارتباط**[1] سرور نامیده شده است.

از نقطه‌نظر کاربردی ارتباط TCP یک تونل مجازی مستقیم بین سوکت‌های مشتری و سرور است. برنامه‌ی مشتری می‌تواند بایت‌های دلخواه را به درون سوکت گسیل نماید و TCP نیز این ضمانت را فراهم آورد که برنامه‌ی سرور آنها را از طریق سوکت ارتباطی خود دریافت کند. چگونگی دریافت نیز براساس ترتیب ارسال بایت‌ها می‌باشد. بنابراین TCP یک سرویس مطمئن جریان بایت را بین برنامه‌های سرور- مشتری فراهم می‌آورد. علاوه بر این موضوع همانگونه که افراد می‌توانند از یک در مشابه وارد و یا خارج شوند، برنامه‌ی مشتری نه تنها بایت‌ها را به درون سوکت گسیل می‌دارد بلکه آنها را از طریق سوکت نیز دریافت می‌کند. به شکل مشابه برنامه‌ی سرور نه فقط بایت‌ها را از طریق سوکت

1- Connection Socket

ارتباطی خود دریافت می‌کند بلکه بایت‌های ارسالی را به درون آن نیز هدایت می‌کند. فرآیند عملکرد سوکت‌ها در شکل ۲- ۲۹ نشان داده شده است. از آنجائیکه سوکت‌ها نقش مهمی را در کاربردهای سرور- مشتری ایفا می‌کنند، بنابراین طراحی و تکمیل کاربردهای مشتری و یا سرور را برنامه‌نویسی سوکت نامیده‌اند.

قبل از بیان مثالی از کاربرد سرور- مشتری، سودمند خواهد بود که واژه‌ی جریان که در متن از آن بارها صحبت به میان آمده بیشتر توضیح داده شود. یک **جریان**[1] عبارت است از رشته‌ای از کاراکترها که به درون یک پردازش وارد و یا از آن خارج می‌شوند. هر جریان معمولاً به دو گونه‌ی **جریان ورودی**[2] به برنامه و **جریان خروجی**[3] از آن می‌باشد. اگر جریان ورودی است آنگاه آن جریان به منبع ورودی برنامه پیوسته است مانند یک ورودی استاندارد از طریق صفحه کلید و یا یک سوکت که جریان داده از اینترنت به درون آن جاری می‌شود. اگر جریان، یک خروجی می‌باشد آنگاه به برخی منابع خروجی برنامه پیوسته است مانند یک خروجی استاندارد از طریق صفحه‌ی نمایشگر و یا یک سوکت خروجی که از طریق آن داده‌ها به درون اینترنت جاری می‌شوند.



شکل ۲- ۲۹ ♦ سوکت‌های مشتری، خوش آمدگویی و ارتباطی

۲- ۲۲ نمونه‌ای از کاربرد مشتری- سرور در جاوا

در ذیل از یک کاربرد ساده‌ی مشتری- سرور جهت نشان دادن برنامه‌نویسی سوکت برای هر دو پروتکل TCP و UDP استفاده شده است:

۱- یک برنامه‌ی مشتری سطری را از ورودی استاندارد خود مانند صفحه کلید می‌خواند و خط خوانده شده را از طریق سوکت خود به سوی سرور خارج می‌کند.

1- Stream
2- Input Stream
3- Output Stream

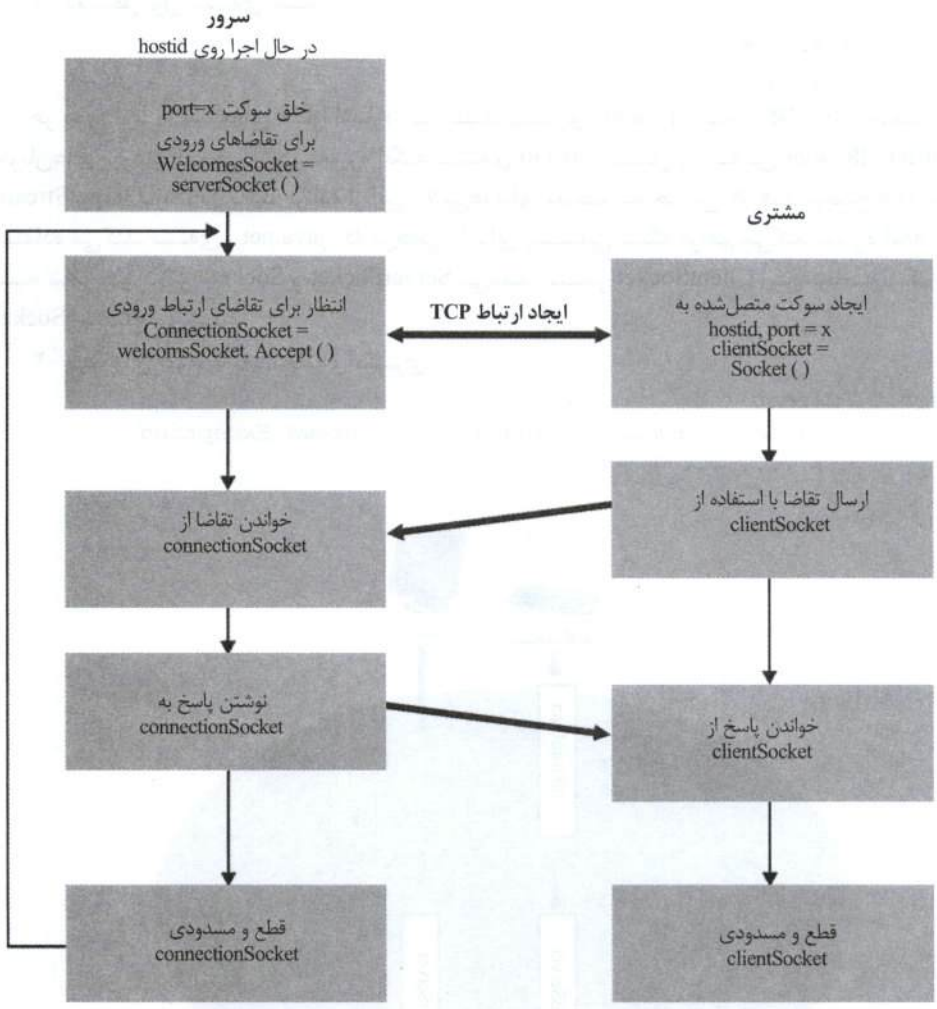
۲- سرور یک خط را از طریق سوکت ارتباطی خود می‌خواند.
۳- سرور آن سطر را تبدیل به حروف بزرگ می‌کند.
۴- برنامه‌ی سرور خط تغییر یافته را از طریق سوکت ارتباطی به سمت مشتری خارج می‌کند.
۵- برنامه‌ی مشتری سطر تغییر یافته را از طریق سوکت خویش می‌خواند و آن خط را بر روی خروجی استاندارد خود مانند یک صفحه‌ی نمایشگر ظاهر می‌کند.

شکل ۲- ۳۰ فعالیت‌های اصلی سوکت مشتری- سرور را نشان می‌دهد. در ادامه‌ی مطلب جفت برنامه‌ی سرور- مشتری برای پیاده‌سازی یک برنامه‌ی کاربردی در TCP آورده شده است. تحلیل هر سطر از برنامه در پی هر یک آورده شده است. برنامه‌ی مشتری TCPClient.Java نامیده شده و برنامه‌ی سرور TCPServer.Java. از آنجائیکه هدف از این مثال‌ها تأکید بر روی موضوعات کلیدی است به همین منظور از کُدهای برنامه‌نویسی جاوا در ساده‌ترین و کوتاهترین شکل ممکن استفاده گردیده است. بدیهی است یک برنامه‌ی مطلوب دارای جزئیات به مراتب بیشتری خواهد بود.

هنگامی‌که دو برنامه بر روی میزبان‌های خود کامپایل و به اجرا درآیند، نخست برنامه‌ی سرور اجرا می‌شود تا یک پردازش را در میزبان سرور خلق نماید و آنگاه این پردازش بر روی سرور در حالت انتظار تماس پردازش مشتری قرار می‌گیرد. در این مثال کاربردی هنگامی‌که برنامه‌ی مشتری اجرا می‌شود یک پردازش نیز در سمت آن خلق شده و بی‌درنگ با سرور تماس برقرار نموده و یک ارتباط TCP را با سرور برقرار می‌کند. در این حالت است که کاربرد در مشتری ممکن است از برنامه‌ی کاربردی جهت ارسال یک سطر و سپس دریافت همان خط به‌صورت حروف بزرگ اقدام نماید.

- **برنامه‌ی کاربرد TCPClient.java در سمت مشتری:**

```
import java.io.*;
import java.net.*;
class TCPClient {
  public static void main(String argv[]) throws Exception
  {
     String sentence;
     String midigiedsentences;
     BufferedReader inFromUser = new BufferedReader (
             New InputstreamReader(System.in));
     Socket cliensocket = new Socket ("hostname", 6789);
     DataOutputStream outToServer = new DataOutputStream(
             clientSocket.getOutputStream());
     BufferedReader inFromServer =
        New BufferedReader (new InputstreamReader (
             clientSocket.getInputStream()));
     sentence = inFromUser.readLine();
     outToServer.writeBytes(sentence + '\n');
     modifiedSentence = in FromServer.readLine();
     System.out.println("FROM SERVER: " +
                        modifiedSentence);
     cliensocket.close();
```



شکل ۲- ۳۰ ♦ کاربرد مشتری- سرور، استفاده از خدمات اتصال‌گرا

نتیجه‌ی اجرای برنامه‌ی بالا خلق سه جریان و یک سوکت می‌باشد، براساس آنچه در شکل ۲- ۳۱ نشان داده شده است، نام سوکت ایجاد شده: ClientSocket می‌باشد و جریان ورودی به برنامه: inFromUser نام دارد که به یک ورودی استاندارد مانند صفحه کلید وابسته شده است. هنگامی‌که کاربر به وارد نمودن کاراکترها از طریق آن اقدام می‌کند، کاراکترها در قالب یک جریان وارد inFromUser می‌شوند. جریان ورود دیگر: inFromServer است که وابسته به سوکت می‌باشد. کاراکترها از شبکه وارد جریان inFromServer می‌شوند. سرانجام جریان outToServer می‌باشد که خروجی برنامه است، این جریان نیز دارای پیوندی به سوکت می‌باشد. در نتیجه کاراکترهایی که مشتری ارسال می‌کند پس از ورود به شبکه در: outToServer جاری می‌شوند.

در ادامه برای سطرهای مختلف برنامه‌ی بالا توضیحات مختصری ارائه می‌گردد:

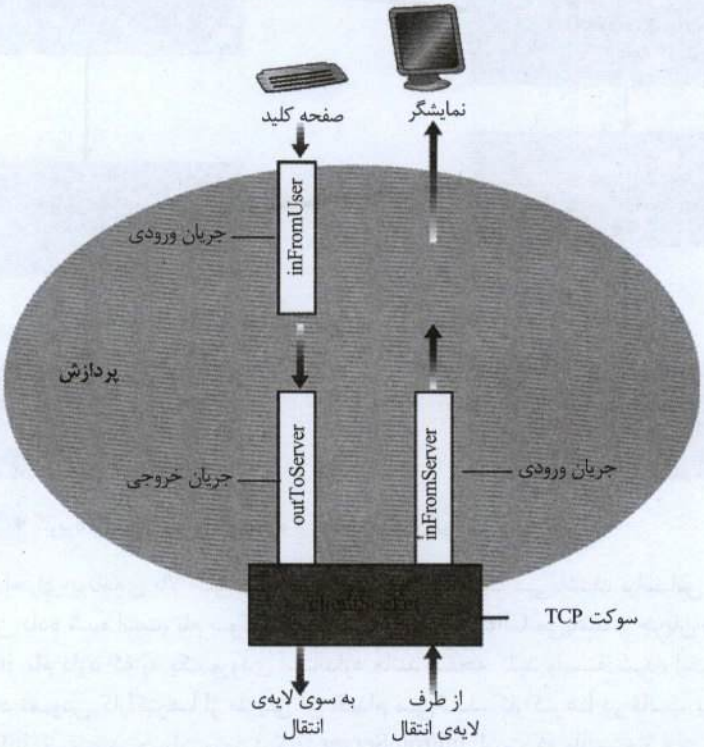
۱- دو سطر اول برنامه‌ی مشتری:

```
import java.io.* ;
import java.net.* ;
```

هر دوی آنها به بسته‌های جاوا اشاره می‌کنند. بسته‌ی java.io شامل کلاسهای متعلق به جریان‌های ورود و خروجی است. به‌ویژه آنکه بسته‌ی java.io شامل دو کلاس BufferedReader و DataOutputStream می‌باشد. برنامه از این کلاس‌ها برای تعریف سه جریانی که قبلاً توضیح داده شد، استفاده می‌کند. بسته‌ی java.net کلاس‌هایی را برای پشتیبانی شبکه فراهم می‌کند به‌ویژه آنکه این بسته شامل دو کلاس: Socket و ServerSocket می‌باشد. عنصر ClientSocket این برنامه از کلاس Socket استنتاج می‌شود.

۲- سطرهای سوم به بعد برنامه‌ی مشتری:

```
Class TCPClient {
  public static void main(string argv[]) throws Exception
          {..... }
}
```



شکل ۲- ۳۱ ♦ عبور اطلاعات از سه جریان **TCPClient**

آنچه تاکنون ملاحظه شد موارد استاندارد شروع اغلب کُدهای جاوا می‌باشد. از خط سوم به بعد بدنه‌ی کلاسها تعریف شده است. واژه‌ی کلیدی class آغازگر تعریف یک کلاس مانند TCPClient است. آنچه یک کلاس حامل آن است عبارتند از متغیرها و روش‌ها، بدیهی است آغاز و پایان یک

کلاس مانند زبان برنامه‌نویسی C با نماد { } تعریف شده است. کلاس TCPClient دارای متغیر نیست اما فقط یک متُد یا روش بنام ()main دارد. روش‌ها مشابه توابع در زبان برنامه‌نویسی C می‌باشند، روش ()main در زبان جاوا همانند تابع ()main در C و یا C++ است. هنگامی‌که کامپایلر جاوا به اجرای یک کاربرد می‌پردازد و یا به عبارت دیگر بر مبنای کنترل کلاس به‌وسیله‌ی کاربر احضار می‌شود آغاز به فراخوانی کلاس در روش ()main می‌کند. متُد ()main آنگاه همه‌ی روش‌های لازم برای اجرای کاربرد را فراخوانی می‌کند. در این معرفی کوتاه و اختصاری برنامه‌نویسی سوکت در جاوا از واژه‌های کلیدی مانند: public، static، viod، main و Throws Exceptions می‌توان صرف نظر نمود اگرچه آنها را بایستی در برنامه منظور نمود.

دو سطر ذیل به بیان و معرفی اشیاء از نوع رشته (string) می‌پردازند:

```
String Sentence ;
String modified Sentence ;
```

عبارت عنصر Sentence رشته‌ای می‌باشد که در حقیقت کاربر آنرا وارد و به سرور ارسال می‌کند. عنصر یا شیئ modified Sentence رشته‌ای است که از سرور حاصل می‌شود و به خروجی استاندارد کاربر گسیل می‌گردد. جهت خلق جریان inFromUser که از نوع Buffered Reader می‌باشد از خط ذیل استفاده می‌شود:

```
BufferedReader inFromUser = new BufferedReader(new
InputStreamReader (System.in));
```

جریان ورودی با عبارت System.in به ورودی استاندارد می‌پیوندد و سپس مقداردهی اولیه می‌شود. این فرمان به برنامه‌ی مشتری اجازه می‌دهد که از طریق صفحه کلید خود متن را بخواند. عبارتی که در ذیل آمده است عنصر ClientSocket را از نوع Socket ایجاد می‌کند. علاوه بر آن فرمان ارتباط TCP بین سرور و مشتری را نیز صادر می‌کند. رشته‌ی "hostname" بایستی با نام میزبان سرور جایگزین گردد. به‌عنوان مثال: "apple.poly.edu". بدیهی است قبل از آنکه ارتباط TCP برقرار شود، برنامه‌ی مشتری یک جستجوی DNS انجام می‌دهد تا آدرس IP میزبان را بدست آورد:

```
Socketclient Socket = new Socket ("hostname", 6789);
```

ارزش 6789 همان شماره پورت است. از شماره پورت‌های گوناگون می‌توان استفاه نمود اما این نکته نبایستی فراموش گردد که از شماره‌های شناخته شده در RFCها نمی‌توان استفاده نمود و نیز از شماره‌ی پورت مشابه نیز بایستی در سمت سرور کاربر استفاده شود. قبلاً اشاره گردید آدرس IP میزبان به همراه شماره‌ی پورت به شناسایی پردازش سرور می‌پردازد. عبارت‌های:

```
DataOutputStream out To Server =
   new DataOutputStream(clientSocket.getOutputStream());
BufferedReader inFromServer =
   new BufferedReader(new inputStreamReader(
                  clientSocket.getInputStream());
```

به ایجاد جریان اشیاء وابسته به سوکت می‌پردازد. جریان: outToServer خروجی پردازش به سوکت را فراهم می‌آورد. عبارت inFromServer ورودی پردازش را از سوکت نتیجه می‌دهد. این موارد از شکل ۲- ۳۱ قابل مشاهده می‌باشد.

```
Sentence = inFromUser.readLine();
```

این سطر به جایگذاری کاراکترهای وارد شده از طریق کاربر به درون رشته ی: sentence می‌پردازد. رشته تا هنگامی که کاربر کلید بازگشت را فشار نداده است، کاراکترها را دریافت می‌کند. خط وارد شده

از ورودی استاندارد مانند صفحه کلید با عبور از جریان inFromUser وارد رشته‌ی sentence می‌شود. عبارت:

```
outToServer.writeBytes (sentence + '\n');
```

آرگومان رشته‌ی sentence را به همراه مفهوم کلید بازگشت (لـ) به درون جریان outToServer گسیل می‌دارد. آرگمان اشاره شده از درون سوکت مشتری عبور و وارد تونل TCP می‌شود. برنامه‌ی مشتری آنگاه به انتظار دریافت کاراکترها از سرور می‌نشیند. از طرف دیگر عبارت ذیل پاسخ برنامه‌ی سرور است:

```
modifiedSentence = inFromserver.readLine();
```

هنگامی‌که کاراکترها از سمت سرور وارد می‌شوند، آنها از درون جریان inFromServer عبور کرده و در رشته‌ی modifiedSentence جایگذاری می‌شوند. کاراکترها مرتباً در این رشته انباشت شده تا زمانیکه رشته با کلید بازگشت پایان پذیرد. جمله‌ی بعدی که در ذیل آمده است رشته‌ی modifiedSentence را از طریق سرور بر روی صفحه نمایشگر، آشکار می‌کند:

```
System.out.println ("FromServer" + modifiedSentence);
```

سرانجام آخرین عبارت برنامه‌یعنی:

```
clientSocket.close();
```

سبب می‌شود سوکت بسته شود و نیز ارتباط TCP بین مشتری و سرور هم قطع گردد. این عملیات از طریق ارسال یک پیام TCP از سوی TCP در مشتری به همتای آن در سرور صورت می‌پذیرد.

- **برنامه‌ی سرور TCPServer.java در سمت سرور**

برنامه‌ی ذیل برنامه‌ی سرور در سمت میزبان سرور می‌باشد:

```
import java.io.*;
import java.net.*;
class TCPServer {
   public static void main(String argv[]) throws Exception
   {
      String clienSentence;
      String capitalizedSentence;
      ServerSocket welcomeSocket = new ServerSocket
         (6789);
      while (true)  {
         Socket connectionSocket = welcomeSocket.
           accept();
         BufferedReader inFromClient =
           new BufferedReader (new InputStreamReader(
              connectionSocket.getInputStream()));
         DataoutputStream outToClient =
           new DataoutputStream(
             connectionSocket.getoutputStream());
         clientnentence = inFrimClient.readLine();
         capitalizedsentence =
              clientnentence.toUpperCase() + '\n';
         outToClient.writeBytes (capitalizedsentence);
      }
   }
}
```

برنامه‌ی TCPServer شباهت‌های بسیار نزدیکی به برنامه‌ی TCPclient دارد. اکنون به بررسی جزئیات سطرهای برنامه‌ی TCPServer.java می‌پردازیم. در مورد سطرهایی که مشابه فرمان‌های TCPClient.java می‌باشند توضیحاتی ارائه نشده است، زیرا قبلاً بررسی شده‌اند.

۱- نخستین سطر برنامه‌ی TCPServer.java تفاوت عمده‌ای را از آنچه در TCPClient مشاهده گردید، شامل می‌شود:

```
ServerSocket welcomeSocket = new ServerSocket(6789);
```

این خط عنصر welcomeSocket را که از نوع ServerSocket است، تعریف می‌کند. این سوکت همانند یک دریچه‌ای است که منتظر است تا از جانب برنامه‌ی مشتری کوبیده شود. سوکت welcoeSocket از طریق شماره پورت 6789 به کوبیدن در از جانب مشتری گوش می‌کند.

سطر بعدی یعنی:

```
Socket connectionSocket = welcomeSocket.accept();
```

یک سوکت جدید که ConnectionSocket نام دارد را تعریف می‌کند. این سوکت جدید هنگامی پدیدار می‌شود که برنامه‌های مشتری عنصر welcomeSocket را صدا می‌کنند. این سوکت نیز دارای شماره پورت 6789 می‌باشد.

علت آنکه چرا سوکت‌ها دارای شماره پورت مشابه می‌شوند؟ در فصل ۳ توضیح داده شده است. پس از انجام فرآیندهای بالا، TCP یک تونل مستقیم مجازی را بین clientSocket در سمت برنامه‌ی مشتری و ConnectionSocket در سمت برنامه‌ی سرور ایجاد می‌کند. پس از آن برنامه‌های سرور و مشتری می‌توانند به مبادله‌ی بایت‌ها بین یکدیگر از طریق تونل بپردازند. همه‌ی بایت‌های ارسالی با همان ترتیب نیز در سمت مقابل دریافت می‌شوند. هنگامی‌که سوکت ارتباط ایجاد گردید آنگاه، سرور می‌تواند به تقاضاهای یک برنامه‌ی مشتری که از جانب برنامه‌ی کاربردی مرتباً در حال ارسال است با استفاده از ConnectionSocket گوش فرا دهد. بدیهی است این نسخه از برنامه که در این مثال آمده است در واقع به تقاضاهای دیگر غیر از آنچه برایش تعریف شده است، گوش نمی‌کند، اما به کمک رشته‌ها می‌توان در آن تغییراتی را اعمال نمود تا این مهم را نیز رعایت نماید. پس از انجام همه‌ی این مراحل آنگاه برنامه به تعریف چندین عنصر (اشیاء) اقدام می‌کند که مشابه هستند با آنچه ClientSocket ایجاد می‌کند. اکنون سطر ذیل بررسی می‌شود:

```
capitalizedSentence = ClientSentence.toUpperCase() + '\n');
```

این فرمان هسته‌ی کاربرد را تشکیل می‌دهد. با استفاده از خط ارسال شده بوسیله‌ی برنامه‌ی مشتری این فرمان، رشته را به حروف بزرگ تبدیل و به آن کلید بازگشت (↵) را اضافه و سپس مجدداً به سمت مشتری گسیل می‌کند. از تابع ()toUpperCase برای انجام این منظور استفاده می‌کند. فرمان‌های باقیمانده‌ی دیگر در برنامه حالت واسط را داشته و از آنها برای ارتباط با برنامه‌ی مشتری استفاده می‌شود. برای آنکه جفت برنامه‌ی مشتری و سرور آزمایش شوند برنامه‌ی TCPClient.java بر روی یک میزبان نصب و کامپایل می‌شود و برنامه‌ی TCPServer.java بر روی میزبان دیگری. پس از آن برنامه‌های TCPServer.class و نیز برنامه‌ی سرور کامپایل شده و هر دو در میزبان سرور اجرا می‌شوند. این موضوع سبب می‌شود پردازشی که در سرور تا آن زمان بیکار بوده است، تعریف شده تا آنکه با برنامه‌ی مشتری در اثر تماس با وی ارتباط برقرار نماید. به دنبال این رویداد برنامه‌ی TCPClient.class و برنامه‌ی مشتری در میزبان مشتری به اجرا در می‌آید. اجرای این عملیات باعث می‌شود یک پردازش در مشتری تعریف و در نتیجه یک ارتباط TCP بین پردازش‌های سرور و مشتری

حاصل گردد. سرانجام برای استفاده‌ی برنامه‌ی کاربرد یک جمله در قالب رشته به همراه کلید بازگشت از سوی کاربر وارد می‌شود.

برای طراحی یک برنامه‌ی کاربردِ مشتری- سرور می‌توان با ایجاد تغییراتی اندک در برنامه به‌عنوان مثال به جای تبدیل کاراکترها به حروف بزرگ، برنامه‌ی سرور تعداد مرتبه‌ای که حرفی مانند s تکرار شده است را شمارش و ارزش عددی آن را بازگشت دهد.

۲- ۲۳ برنامه‌نویسی سوکت با UDP

در بخش قبل آموختیم، هنگامی‌که دو پردازش از طریق TCP تعامل برقرار می‌کنند، مانند تونلی در بین دو پردازش می‌باشند. این تونل یا خط لوله تا هنگامی که دو پردازش متوقف نشده‌اند به حیات خود ادامه می‌دهد. هنگامی‌که یکی از پردازش‌ها مایل به ارسال تعدادی بایت به پردازش دیگر است، آن پردازش به آسانی بایت‌ها را وارد تونل می‌کند. فرآیند ارسال نیازی به پیوستن یک آدرس مقصد به بایت‌ها ندارد، زیرا لوله به‌صورت منطقی به مقصد متصل شده است. علاوه بر آن، تونل یک جریان بایت مطمئن را تعریف می‌کند که ترتیب دریافت بایت‌ها به همان شکل و یا قالب ارسال آنها می‌باشد. پروتکل UDP نیز این اجازه را می‌دهد که دو و یا چندین پردازش در حال اجرا بر روی میزبان‌های مختلف با یکدیگر تعامل برقرار نمایند. امّا UDP در بسیاری از موارد اصولی با TCP تفاوت دارد. اختلاف نخست آنکه UDP یک سرویس غیر اتصال‌گرا می‌باشد به عبارت دیگر در UDP مرحله‌ی دست‌دهی مقدماتی بین دو فرآیند در طول مدت تعریف خط لوله وجود ندارد. زیرا UDP به تعریف تونل نمی پردازد. هنگامی‌که یک پردازش مایل به ارسال یک گروه بایت به فرآیند دیگری است، پردازش گُسیل کننده بایستی آدرس پردازش مقصد را نیز به آنها پیوست نماید. پیوستِ آدرس مقصد بایستی برای هر یک از گروه بایت‌ها صورت گیرد. برای درک بهتر موضوع به مثالی که در ادامه‌ی بحث آمده است، دقت کنید: فرض کنید یک گروه 20 نفری از تعداد 5 تاکسی جهت رسیدن به یک مقصد مشترک استفاده می‌کنند. هنگامی‌که این افراد در حال سوار شدن به تاکسی می‌باشند هر یک از رانندگان تاکسی به گونه‌ی مجزا و مستقل بایستی از آدرس مقصد آگاه شوند. بنابراین UDP را می‌توان مشابه هر یک از رانندگان تاکسی به‌شمار آورد. آدرس مقصد چندگانه است و شامل آدرس‌های IP از میزبان‌های مقصد و نیز شماره پورت پردازش‌های مقصد می‌باشد.

گروه بایت اطلاعات را به همراه آدرس IP آنها و نیز شماره‌ی پورتها از این به بعد بسته می‌نامیم. پروتکل UDP یک الگوی ارتباطی غیر مطمئن پیام را تعریف می‌کند به این مفهوم که حداکثر تلاش خود را جهت تحویل بسته‌ها به مقصد به‌کار می‌گیرد. این پروتکل اصطلاحاً **پیام گرا**[1] می‌باشد زیرا که بسته‌ها، بایت هایی هستند که به طور یک جا و در یک عملیات ارسال از سوی مبداء گُسیل و در قالب دسته‌ها به گیرنده تحویل می‌شوند. این روش کارکرد در تقابل مستقیم با روش TCP است که مفهوم جریان بایت را دنبال می‌کند. از طرفی UDP یک سرویس بر مبنای حداکثر تلاش در تحویل است، بدون آنکه تحویل قطعی دسته‌های بایت به مقصد را ضمانت نماید. در صورتی‌که TCP براساس جریان بایت مطمئن این ضمانت را به وجود می‌آورد که تحویل قطعاً صورت می‌پذیرد.

پس از آماده‌سازی یک بسته، پردازش ارسال کننده آن را از طریق یک سوکت وارد شبکه می‌نماید. اگر بحث مربوط به مثال تاکسی را ادامه دهیم در سمت دیگر سوکت ارسال کننده‌یک تاکسی وجود دارد که انتظار یک بسته را می‌کشد. تاکسی آنگاه حرکت کرده و بسته را به سوی آدرس مقصد هدایت

1- Message oriented

می‌کند و این در حالی است که تاکسی ضمانت تحویل بسته‌ها را به مقصد نمی‌کند. زیرا در بین راه ممکن است دچار عیب شود و یا آنکه حوادث غیر قابل پیش‌بینی دیگری روی دهد. به عبارت دیگر پروتکل UDP انتقال نامطمئن را برای پردازش‌های ارتباطی خود فراهم می‌آورد. در این بخش با استفاده از همان کاربردی که قبلاً اشاره شد. برنامه‌نویسی سوکت را برای UDP انجام می‌دهیم. ملاحظه خواهد شد که برنامه‌ی UDP کاملاً متفاوت از TCP می‌باشد. به‌ویژه این تفاوت در موارد ذیل خود را بیشتر جلوه می‌دهد: (۱) مرحله‌ی دست‌دهی اولیه بین دو پردازش وجود ندارد، بنابراین نیاز به تعریف سوکت خوش‌آمدگویی نمی‌باشد. (۲) جریانی به سوکت‌ها پیوسته نیست. (۳) میزبان ارسال کننده بسته‌ها را با پیوستن آدرس IP مقصد و نیز شماره پورت هر یک از گروه بایت‌ها تعریف و گسیل می کند و (۴) پردازش دریافت کننده بایستی بسته‌ی دریافت شده را از هم باز کند تا به بایت‌های اطلاعات بسته دسترسی پیدا کند. با یادآوری مجدد برنامه‌ی کاربرد ساده‌ای که قبلاً در مورد TCP تشریح شد:

۱- یک برنامه‌ی مشتری سطری را از یک ورودی استاندارد مانند صفحه کلید خوانده و خط را از طریق سوکت خود خارج و به سرور گسیل می‌دارد.

۲- آنگاه برنامه‌ی سرور سطر را از طریق سوکت خود می‌خواند.

۳- سرور کاراکترهای خط را به حروف بزرگ تبدیل می‌نماید.

۴- برنامه‌ی سرور سطر تغییر یافته را از طریق سوکت خود خارج و به سوی برنامه‌ی مشتری گسیل می‌کند.

۵- برنامه مشتری سطر تغییر یافته را از طریق سوکت خود خوانده و آن را بر روی خروجی استاندارد خود مثل یک صفحه‌ی نمایشگر ظاهر می‌کند.

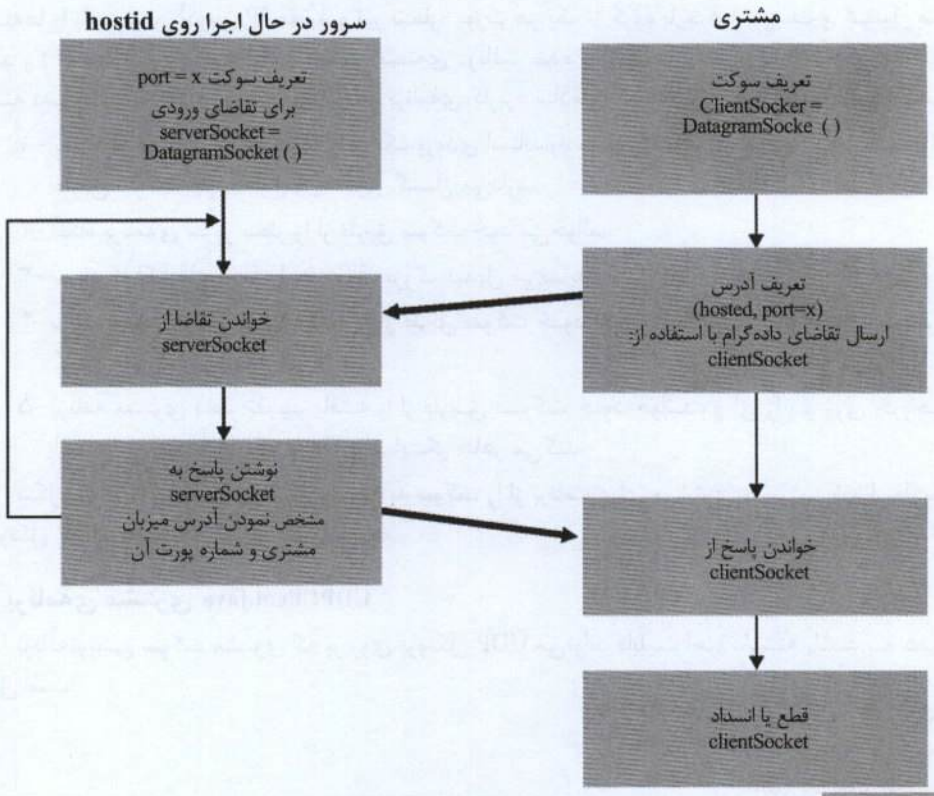
شکل ۲- ۳۲ فعالیت‌های اصلی وابسته به سوکت را از برنامه‌های مشتری و سرور که از طریق پروتکل UDP تعامل می‌کنند، نشان می‌دهد.

- **برنامه‌ی مشتری UDPClient.java**

برنامه‌نویسی سوکت مشتری که بر روی پروتکل UDP می‌تواند قابلیت اجرا داشته باشد به شرح ذیل است:

```
import java.io.*;
import java.net.*;
class UDPClient {
   prublic static void main(String args[]) throws Exception
   {
      BufferedReader inFromUser =
        new BufferedReader(new InputStreameReader
                (System.in);
      DatagramSocket ClientStocket = new DatagramSocket();
      InetAddress IPAddress =
                    InetAddress.getByName("hostname");
      byte[] sendData = new byte[1024];
      byte[] receiveData = new byte [1024]
      String sentence = inFromUsereadLine ();
      sendData = sentence.getBytes();
      DatagramPacket sendPacket =
        new DatagramPacket (sendData, sendData.lenght,
```

```
                    IPAddress, 9876);
        cliensToket.send (sendPacket);
        DatagramPacket receivePacket =
            new String(receivePacket.getData());
        System.out.println('FROM SERVER:" +
                    modifiedSentence);
        clientSocket.close();
    }
}
```



شکل ۲- ۳۲ ♦ کاربرد مشتری- سرور با استفاده از خدمات غیر اتصال‌گرا

برنامه‌ی UDPClient.java یک جریان و یک سوکت را می‌سازد. شکل ۲- ۳۳ این دو مورد را از لحاظ عملکرد نشان داده است. سوکت ساخته شده: Client Socket نام دارد و از نوع: Datagram Socket می‌باشد. لازم به توضیح است که نوع سوکت استفاده شده به‌وسیله UDP متفاوت از TCP در برنامه‌ی مشتری می‌باشد. به‌ویژه در UDP برنامه‌ی مشتری از یک سوکت داده گرام استفاده می‌کند در حالیکه TCP از یک Socket به جای Datagram Socket در برنامه‌ی مشتری استفاده می‌کند. جریان ورودی برنامه inFromUser است که وابسته به یک ورودی استاندارد مانند صفحه کلید است. معادل همین جریان نیز در TCP وجود دارد هنگامی‌که کاربر کاراکترها را وارد می‌کند، آنها در جریان inFromUser جاری می‌شوند اما در مقایسه با TCP جریان‌های ورودی و یا خروجی وجود ندارند که به سوکت وابسته باشند. به جای آن بایت‌ها هستند که جریان وابسته‌ی عنصر Socket را تغذیه می‌کنند، UDP بسته‌ها را به گونه‌ی منفرد به درون Datagram Socket تغذیه می‌کند.

اکنون سطرهای برنامه‌ی مشتری UDP که بسیار متفاوت در مقایسه با برنامه‌ی مشتری TCP است، تحلیل می‌شود:

```
DatagramSocket ClientSocket = new DatagramSocket();
```

این سطر یک عنصر ClientSocket از نوع DatagramSocket را تعریف می‌کند. در مقایسه با برنامه‌ی مشتری TCP که قبلاً بررسی شد این خط پیشقدم در ایجاد یک ارتباط TCP نمی‌شود. به‌ویژه آنکه میزبان مشتری به هنگام اجرای این سطر با میزبان سرور تماسی را برقرار نمی‌کند. به همین دلیل، سازنده‌ی ()DatagramSocket نام میزبان سرور و یا شماره پورت را به‌عنوان آرگمان اختیار نمی‌کند. با استفاده از مثال دریچه و لوله که در برنامه‌ی قبلی از آن استفاده شد، اجرای سطر بالا به تعریف یک دَر برای فرآیند مشتری می‌پردازد که در آن تونل یا لوله‌ای بین دو پردازش مشتری و سرور ایجاد نمی‌شود. جهت ارسال بایت‌ها به یک پردازش مقصد، به آدرس پردازش نیاز می‌باشد که بخشی از آن آدرس IP میزبان مقصد است. سطر ذیل:

```
InetAddress IPAddress=InetAddress.getByName("host name");
```

یک جستجوی DNS را فعال می‌کند که نام میزبان (hostname) را که طراح تعریف کرده است، به یک آدرس IP ترجمه می‌کند. پروتکل TCP نیز در برنامه‌ی مشتری خود به فراخوانی DNS اقدام می‌کند اگرچه این فراخوانی در آنجا به‌صورت تلویحی انجام می‌گرفت. تابع ()getByName در برنامه‌ی سرور آرگمان hostname را اختیار و آدرس IP همان سرور را بازگشت می‌دهد. این آدرس در عنصر IPAddress که از نوع InetAddress است جایگذاری می‌شود. در دو سطر ذیل از دو آرایه‌ی بایت استفاده شده است:

```
byte[ ] sendData = new byte [1024];
byte[ ] receiveData = new byte [1024];
```

دو آرایه‌ی sendData و receiveData به ترتیب به ذخیره‌ی داده‌هایی می‌پردازد، که برنامه مشتری آنها ارسال و یا دریافت می‌کند. تبدیل نوع که عبارت است از دریافت رشته‌ی sentence و تغییر نام آن به‌صورت sendData که در حقیقت یک آرایه‌ی بایت است از طریق کد زیر انجام می‌شود:

```
sendData = sentence.getBytes();
```

ساختن بسته‌ی sendPacket که برنامه‌ی مشتری آنرا با عبور از سوکت خود به درون شبکه تزریق می‌کند، به‌وسیله‌ی کُد ذیل انجام می‌شود:

```
DatagramPacket sendPacket =      new DatagramPacket(
   sendData, sendData.length, IPAddress, 9876);
```

این بسته شامل داده‌ای متعلق به بسته‌ی sendData، طول و یا اندازه داده، آدرس IP سرور و شماره پورت کاربرد که با ارزش 9876 تنظیم شده است، می‌گردد. بدیهی است بسته‌ی sendPacket از نوع DatagramPacket است. تابع () send از عنصر ClientSocket، بسته‌ی جدید ساخته شده را تحویل و از طریق عبور از clientSocket به شبکه تزریق می‌کند. توجه به این نکته ضروری است ارسال سطر کاراکترها از طریق UDP کاملاً متفاوت با TCP می‌باشد:

```
ClientSocket.send(sendPacket);
```

پروتکل TCP به سادگی رشته‌ای از کاراکترها را وارد جریانی می‌نمود که یک ارتباط مستقیم منطقی را با برنامه‌ی سرور داشت. در حالیکه UDP بسته‌ای را تعریف می‌کند که در برگیرنده‌ی آدرس سرور است. پس از ارسال بسته برنامه‌ی مشتری منتظر می‌ماند تا بسته‌ای را از سرور دریافت نماید.

هنگامی‌که در حال انتظار است، برنامه‌ی مشتری به تعریف یک ذخیره کننده‌ی بسته به نام receivePacket می‌پردازد که شیئی از نوع DatagramPacket می‌باشد. این عمل با استفاده از کُد زیر انجام می‌شود:

```
DatagramPacket receivePacket =
   new DatagramPacket (receiveData, receiveData.length);
```

در سطر ذیل برنامه‌ی مشتری تا هنگام دریافت یک بسته ساکن می‌ماند و در لحظه‌ی دریافت، آن بسته را به درون receivePacket تزریق می‌نماید. عملیات جداسازی داده از receivePacket و تبدیل نوع از طریق کُد ذیل انجام می‌شود:

```
String modifiedSentence =
   new String (receivePacket.getData());
```

سطر ذیل که مشابه آن در برنامه‌ی TCPClient • java نیز به‌کار گرفته شده، رشته حاصل از اجرای سطر بالا بر روی خروجی استاندارد برنامه‌ی مشتری را به نمایش در می‌آورد:

```
System.out.println("From Server:" + midifiedSentence);
```

و سرانجام سطر آخر که سوکت را به حالت مسدود تبدیل می‌کند. از آنجائیکه UDP غیر اتصال‌گرا می‌باشد. این کُد سبب ارسال یک پیام لایه‌ی انتقال به سرور در مقایسه با TCP از سوی برنامه‌ی مشتری نمی‌شود.

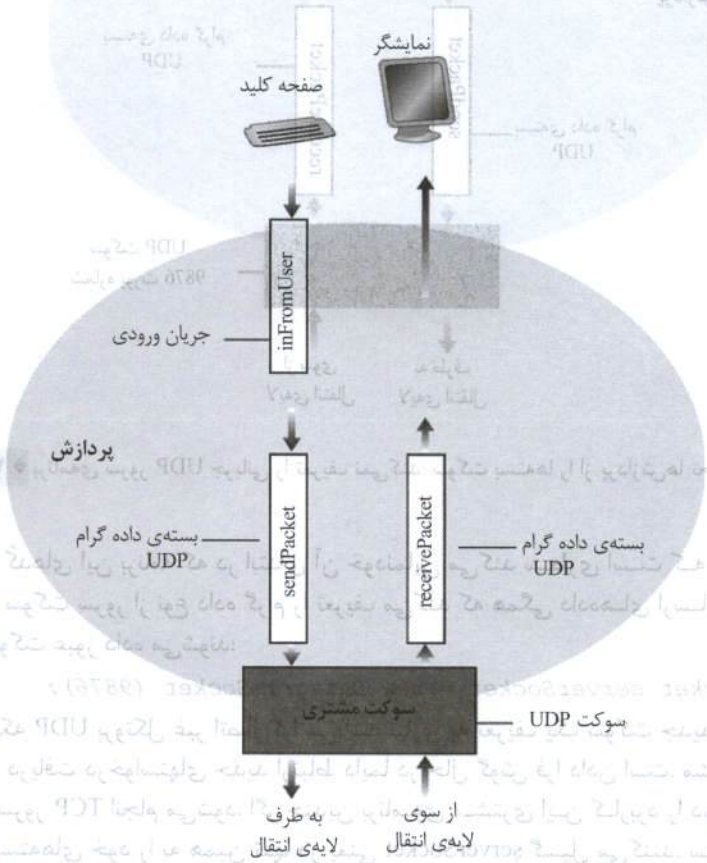
```
ClientSocket.close ();
```

- **برنامه‌ی سرور**

در ذیل برنامه‌ی کاربرد سمت سرور UDP آمده است:

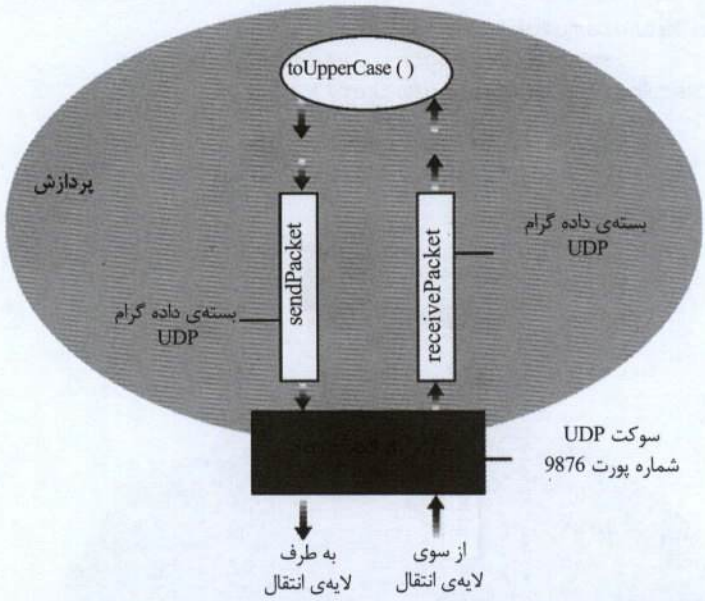
```
import java.io.*;
import java,net.*;
class UDPServer {
  public static void main(string args[]) throws Exception
     {
     DatagramSocket serverSocket = new
                        DatagramSocket (9876)
     byte [] recieveData = new byte[1024];
     byte [] sendData = new byte [1024];
     while(true)
        {
        DatagramPacket recievepacket =
           new DatagramPacket(receiveData,
                        receiveData.length);
        serverSocket.recieve (recievePacket);
        String sentence = new String(
                        recievePacket.getData());
        InetAddress IPAddress =
                        recievePacket.getAddress());
        int prot = recievePacket.getPort();
        string captslizedSentence =
                        sentence.toUpperCase();
        sendData = capitalizedSentence.getBytes();
        Datagrampacket sendPacket =
```

```
            New Datagrampacket (sendData,
                    sendData.length, IPAddress, port);
            serverSocket.send(sendPacket);
        }
    }
}
```



شکل ۲- ۳۳ ♦ برنامه‌ی مشتری UDP یک جریان تعریف می‌کند. سوکت بسته‌ها را از پردازش‌ها دریافت و آنها را تحویل می‌دهد.

برنامه‌ی UDPServer.java براساس شکل ۲- ۳۴ به تعریف یک سوکت می‌پردازد. این سوکت serverSocket نام دارد و یک عنصر از نوع DatagramSocket است. مشابه آنچه در کاربرد سمت مشتری تعریف شد. بار دیگر در این برنامه هیچگونه جریان وابسته به سوکت وجود ندارد. در ادامه کُدهای برنامه به همراه جزئیات آنها بررسی می‌شوند. این برنامه کاملاً متفاوت از همتای خود TCPServer.java می‌باشد.



شکل ۲- ۳۴ برنامه‌ی سرور UDP جریانی را تعریف نمی‌کند. سوکت بسته‌ها را از پردازش‌ها تحویل و یا به آنها گسیل می‌دارد.

از جمله کُدهای این برنامه که در ابتدای آن خودنمایی می‌کند سطری است که بر روی شماره پورت 9876 سوکت سرور از نوع داده گرام را تعریف می‌کند که همگی داده‌های ارسالی و دریافتی از درون این سوکت عبور داده می‌شوند:

```
DatagramSocket serverSocket = new DatagramSocket (9876);
```

از آنجائیکه UDP پروتکل غیر اتصال‌گرا می‌باشد نیازی به تعریف یک سوکت جدید نمی‌باشد و این سوکت برای دریافت درخواستهای جدید ارتباط دایماً در حال گوش فرا دادن است. مشابه همین عمل در برنامه‌ی سرور TCP انجام می‌شود. اگر چندین برنامه‌ی مشتری این کاربرد را دستیابی نمایند، همگی آنها بسته‌های خود را به همین تنها در یعنی serverSocket گسیل می‌کنند. سه سطر دیگر در ذیل، بسته‌های دریافتی از مشتری را از یکدیگر جدا می‌کنند. نخستین سطر داده‌ها را از بسته جدا کرده و آنها را در رشته‌ی String sentence قرار می‌دهد:

```
String sentence = new  Stzring(receivePacket.getData());
InetAddress IPAddress = receivePacket.getAddress();
int port = receivePacket.getport();
```

مشابه سطر نخست در سمت مشتری UDP نیز وجود دارد. سطر دوم آدرس IP را جداسازی می‌نماید و خط سوم شماره‌ی پورت مشتری را که برنامه‌ی مشتری گزینش نموده و با شماره پورت سرور یعنی 9876 تفاوت دارد، جداسازی می‌کند. در فصل ۳ شماره پورت‌های برنامه‌های مشتری با جزئیات بیشتری بررسی خواهند شد.

بدیهی است به دلیل ضرورت، سرور بایستی آدرس IP و شماره پورت مشتری را دستیابی نماید تا قادر به ارسال جمله‌ی تبدیل شده به حروف بزرگ (به‌عنوان مثال) برای برنامه‌ی مشتری باشد. در اینجا تحلیل جفت برنامه‌های مشتری و سرور UDP به انتها می‌رسد. بدیهی است برای آزمایش کارکرد

این دو برنامه لازم است برنامه‌ی UDPClient.java بر روی یک میزبان و برنامه‌ی java • UDPServer بر روی میزبان دیگری نصب، کامپایل و اجرا شوند. رعایت این نکته ضروری است که نام میزبان مناسب سرور بایستی در برنامه مشتری UDP قرار گیرد. به بیان دیگر جایگزین hostname در برنامه UDPClient.java بایستی نام کامل و درست سرور مورد نظر باشد. برعکس TCP می‌توان نخست برنامه‌ی مشتری را اجرا نمود و سپس برنامه سمت سرور را، علت این موضوع آن است که پردازش مشتری تلاش نمی‌کند که ارتباطی با سرور به هنگام اجرای برنامه‌ی مشتری برقرار نماید. هنگامی‌که دو برنامه‌ی مشتری و سرور به حالت اجرا درآیند، آنگاه می‌توان از کاربرد جهت وارد نمودن پیامی به برنامه‌ی مشتری استفاده نمود.

۲-۲۴ برنامه‌نویسی سوکت با UDP در محیط زبان برنامه‌نویسی پیتون (Python)

از جمله تغییرات در عرصه‌ی برنامه‌نویسی سوکت بهره‌گیری از محیط زبان برنامه‌نویسی (Python) است که در ویراست ششم جایگزین زبان برنامه‌نویسی جاوا شده است. در ادامه‌ی مطالب مربوط به طراحی برنامه‌های کاربردی در این فصل برنامه‌های مشتری و سرور برای هر دو مورد TCP و UDP به زبان پیتون درج می‌شود. ابتدا برنامه‌ی کاربرد مشتری با نام UDPClient.py و به دنبال آن برنامه‌ی سرور وابسته به آن با نام UDPServer.py نگارش می‌شود. در مثال‌های برنامه‌ای عنوان شده سعی شده است بر روی موارد کلیدی تأکید بیشتری اعمال شده و از درج نکات ساده پرهیز گردد. برنامه‌ی کاربردی طراحی شده در ذیل از شماره پورت سرور فرضی 12000 بهره گرفته است:

- **برنامه‌ی UDPClient.py**

دستورها مربوط به برنامه‌ی مشتری UDP به شرح ذیل است:

```
from socket import*
serverName = 'hostname'
serverPort = 12000
clientSocket = socket (socket.AF_INET, socket.SOCK_DGRAM)
message = raw_input ('Input lowercase sentence:')
clientSocket.sendto (message, (serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom (2048)
print modifiedMessage
clientSocket.close ( )
```

اکنون به تحلیل سطرهای قطعه برنامه‌ی بالا می‌پردازیم:

- عبارت: *from socket import اساس همه‌ی ارتباط‌های شبکه در پیتون را تشکیل می‌دهد. با درج این جمله امکان خلق یک سوکت در درون برنامه میسر می‌شود.
- عبارت‌های: serverName = 'hostname' و serverPort=12000 به‌ترتیب رشته‌ی serverName را به نام میزبان (hostname) ارتباط می‌دهد. در این مرحله رشته‌ای تعریف می‌شود که شامل آدرس IP سرور مانند: 128.138.32.126 یا نام سرور مانند: cis.poly.edu می‌باشد. اگر از نام میزبان بهره‌گیری شود آنگاه یک جست‌وجوی DNS به‌صورت خودکار آدرس IP را استخراج می‌کند. عبارت دوم یک متغیر با ارزش کامل (integer) را به پورت سرور اختصاص می‌دهد که در این مثال این ارزش برابر 12000 است.
- سطر چهارم برنامه که با عبارت: clientSocket آغاز می‌شود یک سوکت مشتری را ایجاد

می‌کند. نخستین پارامتر خانواده‌ی آدرس AF_INET را تعریف می‌کند که مفهوم آن استفاده از پروتکل IPV4 در شبکه می‌باشد و پارامتر دوم نوع سوکت را تعریف می‌کند که در این مثال SOCK-DGRAM است و مفهوم آن یک سوکت از نوع UDP است. بدیهی است نیازی به تعریف شماره پورت سوکت مشتری نمی‌باشد زیرا سیستم عامل شبکه این مهم را برای ما انجام می‌دهد. اکنون دریچه‌ی ورودی پردازش مشتری ایجاد شده و نیاز به خلق یک پیام برای ارسال از طریق این دریچه است.

- سطر پنجم قطعه برنامه‌ی اشاره شده این مهم را انجام می‌دهد. در زبان برنامه‌نویسی پیتون تابع درون‌سازی شده‌ی: ( ) raw_input هنگام فراخوان پیام: Input data را به سمت کاربر ارسال و درنتیجه کاربر از طریق صفحه کلید، یک خط عبارت را وارد می‌نماید و عبارت درج شده وارد متغیر message می‌شود. اکنون یک سوکت و یک پیام به وجود آمده است و مایل به ارسال پیام از طریق سوکت به میزبان مقصد هستیم.
- سطر ششم از طریق تابع ( ) sendto آدرس مقصد یعنی (serverName, serverPort) را به پیام پیوست نموده و بسته‌ی نتیجه شده را به پردازش سوکت یعنی clinetSocket گسیل می‌کند. قبلاً اشاره گردید که آدرس مبدأ نیز به‌صورت خودکار به بسته اضافه می‌شود. ارسال یک پیام از مشتری به سرور از طریق سوکت UDP به همین سادگی انجام‌پذیر است. پس از ارسال بسته برنامه‌ی مشتری منتظر دریافت داده از سوی برنامه‌ی سرور می‌شود.
- در سطر هفتم برنامه هنگامی که یک بسته از طریق اینترنت به سوکت مشتری وارد می‌شود محموله‌ی داده در بسته وارد متغیر: modifiedMessage می‌شود و آدرس مبدأ بسته نیز به متغیر: serverAddress اختصاص می‌یابد. متغیر serverAddress شامل آدرس IP سرور و شماره پورت سرور است. برنامه‌ی UDPClient نیازی به اطلاعات آدرس سرور ندارد زیرا در آغاز چنین تعامل‌هایی از وجود این آدرس آگاه می‌شود. این سطر پیتون در هر صورت این آدرس را تأمین می‌کند. فرآیند recvfrom نیز اندازه‌ی بافر یعنی 2048 را به‌عنوان ورودی قلمداد می‌کند. این اندازه‌ی بافر برای بسیاری از کاربردها مناسب است.
- در سطر هشتم نتیجه‌ی modifiedMessage روی صفحه ی نمایشگر کاربر ظاهر می‌شود و بایستی همان سطر اصلی باشد که کاربر آن را از طریق صفحه کلید وارد نموده است با این تفاوت که پیام اشاره شده به‌صورت حروف بزرگ ظاهر می‌شود.
- سرانجام سطر نهم برنامه‌ی UDPClient.py سوکت خلق شده را پایان می‌دهد و درنتیجه فرآیند خاتمه می‌یابد.

- **برنامه‌ی UDPServer.py**

اکنون سمت برنامه سرور یک کاربرد تحلیل می‌شود:

```
from socket import*
serverPort = 12000
ServerSocket = socket (AF_INET, SOCK_DGRAM)
ServerSocket.bind (('', serverPort))
print "The server is ready to receive"
While 1:
    message, clientAddress = ServerSocket.recvfrom (2048)
    modifiedMessage = message.upper ( )
    serverSocket.sendto (modifiedMessage, clientAddress )
```

توجه به این نکته ضروری است که سه سطر نخست در آغاز برنامه‌ی UDPServer.py همانند UDPClient.py است و مشابه همان عملیات در برنامه‌ی مشتری می‌باشد. پس از آن از سطر چهارم به بعد مراحل قطعه‌ی بالا را می‌توان چنین تعریف کرد:

- سطر چهارم برنامه شماره پورت 12000 را به سوکت سرور اختصاص می‌دهد. بنابراین در سرور UDP دستور نگارش شده از سوی طراح کاربرد یک شماره انحصاری را به این سوکت اختصاص می‌دهد. با این رویکرد ارسال یک بسته از سوی هر کاربری می‌تواند به پورت 12000 در آدرس IP سرور و سرانجام به این سوکت هدایت شود. پس از آن سرور UDPServer وارد یک حلقه‌ی while می‌شود و این حلقه اجازه‌ی دریافت و پردازش نامحدود بسته‌ها را از سوی برنامه‌ی مشتری امکان‌پذیر می‌سازد. در حلقه‌ی while آنگاه UDPServer در انتظار ورود یک بسته قرار می‌گیرد.
- در سطر هفتم برنامه هنگامی که یک بسته وارد سوکت سرور می‌شود، محموله‌ی داده‌ی بسته به متغیر message انتقال می‌یابد و آدرس مبدأ بسته نیز وارد متغیر ClientAddress می‌شود. متغیر ClientAddress شامل آدرس IP مشتری و شماره پورت مشتری می‌باشد. در این مرحله UDPServer از اطلاعات این آدرس استفاده می‌کند تا یک آدرس بازگشت مشابه آدرس برگشتی یک نامه‌ی معمولی در تعامل‌های پستی را تعریف نماید. با بهره‌گیری از اطلاعات آدرس مبدأ، سرور از مکان بازگشت پاسخ پیام آگاه می‌شود.
- سطر هشتم که مهم‌ترین بخش این کاربرد ساده است سطر ارسال شده از سوی برنامه‌ی مشتری را از طریق بهره‌گیری از روش ( ) upper به حروف بزرگ تبدیل می‌کند.
- سرانجام سطر نهم برنامه UDPServer.py آدرس مشتری یعنی آدرس IP و شماره پورت برنامه‌ی مشتری را به پیام تبدیل شده به حروف بزرگ پیوست می‌نماید و بسته‌ی نتیجه شده را وارد سوکت سرور می‌کند. قبلاً اشاره شد آدرس سرور نیز به‌صورت خودکار به بسته پیوست می‌شود. پس از آن اینترنت بسته را به برنامه‌ی مشتری تحویل می‌دهد. هنگامی که سرور در حال ارسال بسته است از فرآیند حلقه‌ی while خارج نمی‌شود زیرا منتظر دریافت بسته‌ی UDP دیگری از هر برنامه‌ی مشتری و روی هر میزبان دیگری می‌باشد.

۲-۲۵ برنامه‌نویسی سوکت با TCP در محیط زبان برنامه‌نویسی پیتون

برعکس UDP پروتکل TCP اتصال‌گرا می‌باشد و بنابراین فرآیندهای نشان داده شده در شکل ۲-۳۰ به قوت خود باقی است تنها تفاوت در سمت برنامه‌ی سرور و در بالای آن است که عبارت‌ها تغییر عنوان پیدا کرده‌اند. در این شکل برنامه‌ی مشتری یک سطر داده را به سرور گسیل می‌کند و سرور سطر اطلاعات را به حروف بزرگ تبدیل و آن را به مشتری بازگشت می‌دهد.

- **برنامه‌ی TCPClient.py**

دستورهای مربوط به سمت برنامه‌ی کاربرد مشتری TCP به‌صورت ذیل خلاصه می‌شوند:

```
from socket import*
serverName = 'servername'
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket.connect ((serverName, serverPort))
```

```
Sentence = raw_input ('Input lowercase sentence:')
clientSocket.send (sentence)
modifiedSentence = clientSocket. recv (1024)
print 'From Server: ', modifiedSentence
clientSocket.close ( )
```

اکنون به بررسی سطرهای مختلف این قطعه برنامه و تفاوت‌های مهم آن نسبت به یک برنامه‌ی UDP می‌پردازیم: سه سطر نخست برنامه مشابه UDPClient.py با مفهوم یکسان است.

- در سطر چهارم سوکت برنامه‌ی مشتری تعریف می‌شود که پارامتر نخست آن مجدداً به تعریف شبکه با استفاده از IPV4 می‌پردازد. آرگومان دوم سوکت را از نوع SOCK-STREAM تعریف می‌کند که مفهوم آن سوکت TCP است. بدیهی است شماره پورت متعلق به سوکت مشتری هنگام خلق آن تعریف نمی‌شود بلکه اجازه داده می‌شود تا سیستم عامل آن را تعریف نماید. سطر بعدی (سطر پنجم) نقش کاملاً متفاوتی با آنچه همین خط در UDPClient انجام می‌داد را ایفا می‌کند.
- در سطر پنجم لازم به یادآوری است که قبل از آنکه مشتری قادر به ارسال داده به سرور یا برعکس آن از طریق یک سوکت TCP باشد ابتدا یک ارتباط TCP بین دو برنامه‌ی مشتری و سرور بایستی ایجاد شود. آرگومان تابع: ( ) connect آدرس سمت سرور ارتباطی است. پس از اجرای این سطر دستور عملیات دست‌دهی سه جانبه که قبلاً به آن اشاره رفته است انجام می‌شود و یک ارتباط TCP بین دو برنامه‌ی سرور و مشتری به‌وجود می‌آید.
- در سطر ششم همانند برنامه‌ی UDPClient این سطر جمله‌ای را از کاربر دریافت می‌کند. رشته‌ی sentence تا زمانی به جمع‌آوری کاراکترها اقدام می‌کند که کاربر از طریق صفحه کلید کاراکتر بازگشت نورد یا (↵) را وارد نماید.
- سطر هفتم برنامه رشته‌ی sentence را از طریق سوکت مشتری گسیل و آن را وارد ارتباط TCP می‌کند. بدیهی است برنامه انحصاراً یک بسته را به همراه آدرس مقصد آن همانند UDPClient ایجاد نمی‌کند بلکه برنامه‌ی مشتری بایت‌های درون رشته sentence را به درون ارتباط TCP منتقل می‌کند. پس از آن برنامه‌ی مشتری منتظر دریافت بایت‌ها از برنامه‌ی سرور می‌شود.
- سطر هشتم هنگامی‌که کاراکترها از سرور وارد می‌شوند آنها را درون رشته‌ی modifiedSentence جایگذاری می‌کند. کاراکترها در این رشته تا دریافت کاراکتر (↵) بازگشت نورد انباشت می‌شوند. پس از چاپ جمله‌ی تبدیل شده به حروف بزرگ در سطر نهم از طریق سطر دهم سوکت برنامه‌ی مشتری مسدود می‌شود. سطر آخر سبب می‌شود ارتباط TCP بین برنامه‌ی مشتری و سرور بسته شود. این سطر همچنین باعث می‌شود تا TCP در برنامه‌ی مشتری یک پیام TCP را به TCP واقع در برنامه‌ی سرور ارسال نماید.

- **برنامه‌ی TCPServer.py**

اکنون برنامه‌ی TCPServer.py که در ذیل درج شده است تحلیل می‌شود:

```
from socket import*
serverPort = 12000
ServerSocket = socket (AF_INET, SOCK_STREAM)
ServerSocket.bind (('' ,ServerPort))
```

```
ServerSocket.listen(1)
print 'The server is ready to receive'
while 1:
   connectionSocket, addr = serverSocket.accept ( )
   Sentence = connectionSocket.recv (1024)
   capitalizedSentence = sentence.upper ( )
   connectionSocket.send (capitalizedSentence)
   connectionSocket.close( )
```

سطرهای این برنامه نیز از نقطه‌نظر مفاهیم تفاوت چشمگیری با UDPServer.py دارند. دو سطر نخست برنامه مفهوم قبلی خود را دارند. در سطر سوم برنامه‌ی سرور یک سوکت TCP مانند برنامه مشتری TCP خلق می‌کند. مانند UDPServer شماره پورت سرور یعنی ServerPort وابسته به این سوکت بر مبنای سطر چهارم می‌شود. در TCP سوکت سرور یعنی ServerSocket در حقیقت سوکت خوش‌آمدگو به‌شمار می‌آید و پس از خلق این درِ خوش‌آمدگو، سرور در انتظار فراخوان یک برنامه‌ی مشتری روی آن در قرار می‌گیرد. در سطر پنجم برنامه‌ی سرور منتظر تقاضاهای ارتباط TCP از سوی مشتری می‌ماند. آرگومان درون تابع حداکثر تعداد ارتباط‌های در صف را مشخص می‌کند که در این مثال حداقل یک ارتباط است. سطر هشتم در درون حلقه‌ی while هنگامی که یک برنامه‌ی مشتری از طریق دریچه‌ی خوش‌آمدگو فراخوانی را انجام می‌دهد تابع () accept را برای serverSocket تحریک نموده و در نتیجه یک سوکت جدید روی سرور به نام: connectionSocket ایجاد می‌شود و به مشتری انحصاراً اختصاص می‌یابد. پس از آن سرور و مشتری عملیات دست‌دهی سه جانبه را تکمیل و یک ارتباط TCP بین clientSocket مشتری و connectionSocket سرور ایجاد می‌کنند. هنگامی که این ارتباط TCP برقرار شود هر دو برنامه‌ی مشتری و سرور از طریق این ارتباط بایت‌های اطلاعات را بین یکدیگر مبادله می‌کنند. با پروتکل TCP نه‌تنها تمامی بایت‌های ارسال شده از یک طرف ضمانت ورود کامل به طرف مقابل را دارند بلکه نظم ورود آنها نیز ضمانت می‌شود و سرانجام در سطر آخر پس از ارسال جمله‌ی تغییر یافته به مشتری سوکت ارتباط مسدود می‌شود اما از آنجا که ServerSocket باز باقی می‌ماند. برنامه‌های مشتری دیگر می‌توانند به همین طریق به سرور عبارت‌هایی را برای تغییر ارسال نمایند.



تمرین‌های فصل دوم

۱- چه تفاوتی بین معماری‌های شبکه و کاربردها وجود دارد؟

۲- چه اطلاعاتی را یک پردازش در حال اجرا بر روی یک میزبان جهت شناسایی یک فرآیند در حال اجرا بر روی یک میزبان دیگر نیاز دارد؟

۳- برای انجام یک تعامل بین جفت پردازش‌ها کدامیک پردازش مشتری و کدامیک سرور است؟

۴- در مطالب فصل گفته شد پروتکل TCP را می‌توان با استفاده از SSL گسترش داد. این موضوع می‌تواند در جهت امنیت بیشتر به‌کار گرفته شود. توضیح دهید حوزه‌ی عملکرد SSL در کدامیک از لایه‌های انتقال و یا کاربرد قرار می‌گیرد؟ اگر طراح کاربردی بخواهد از طریق SSL عملکرد TCP را گسترش دهد چه تدابیری را بایستی اتخاذ نماید؟

۵- با مراجعه به شکل ۲- ۴ که صورتی از کاربردهای بدون اتلاف زمان و داده را نشان می‌دهد آیا می‌توانید کاربردی را اضافه نمایید که سبب اتلاف داده نشده اما نسبت به زمان بسیار حساس باشد؟

۶- مفهوم پروتکل دست‌دهی چیست؟

۷- یک سامانه‌ی تجارت الکترونیکی را در نظر بگیرید که مایل به نگهداری سوابق خرید هر یک از خریداران خود می‌باشد. توضیح دهید چگونه‌این مهم از طریق استفاده از کوکی‌ها امکان‌پذیر می‌شود؟

۸- علت آنکه پروتکل‌های HTTP، FTP، SMTP و POP3 همگی بر روی لایه‌ی TCP در عوض UDP اجرا می‌شوند، چیست؟

۹- فرض کنید A با استفاده از یک حساب کاربری پست الکترونیکی بر پایه‌ی وب مانند gmail پیامی را به B ارسال می‌دارد. اگر B پیام‌های خود را از سرور پُست با استفاده از POP3 دریافت نماید، توضیح دهید چگونه پیام‌ها از A به میزبان B گسیل می‌شوند. صورتی از زنجیره‌ی پروتکل‌های لایه‌ی کاربرد که برای این جابه جایی استفاده می‌شود را نام ببرید.

۱۰- از نقطه‌نظر کاربر چه تفاوتی بین بارگیری پایین و حذف و نوع بارگیری پایین و نگهداری در پروتکل POP3 وجود دارد؟

۱۱- با چه روشی ارسال پیام رسانی لحظه‌ای (messaging) به‌صورت جستجوی متمرکز بر روی ترکیبی از معماری سرور- مشتری و P2P انجام می‌گیرد؟

۱۲- پروتکل Skype از روش P2P برای دو کاربرد مهم بهره می‌گیرد آن دو کاربرد چیست؟ توضیح دهید.

۱۳- حداقل تعداد چهار کاربرد مختلف که مناسب برای معماری P2P می‌باشند را نام ببرید؟ (به‌عنوان مثال توزیع فایل و پیام رسانی لحظه‌ای دو نمونه از این کاربردها هستند).

۱۴- در بخش مربوط به برنامه‌نویسی سوکت توضیح داده شد که سرور UDP فقط نیازمند یک سوکت است، در حالیکه سرور TCP به دو سوکت احتیاج دارد. توضیح دهید چرا؟ اگر لازم باشد سرور TCP تعداد n ارتباط همزمان را پشتیانی کند و هر کدام از میزبان‌های مشتری نیز مختلف باشند به چه تعداد سوکت سرور از نوع TCP نیازمند است؟

۱۵- یک برنامه‌ی مشتری HTTP را تصور نمایید که مایل به بازیابی یک صفحه‌ی وب در یک URL است. آدرس IP سرور HTTP در آغاز شناخته شده نمی‌باشد. علاوه بر پروتکل HTTP چه پروتکل‌های لایه‌ی کاربرد و انتقال دیگر لازم است؟

۱۶- با مطالعه‌ی RFC959 برای پروتکل FTP صورتی از فرمان‌های مشتری که از طریق این RFC پشتیبانی می‌شود را ارائه نمایید.

۱۷- با توجه به شکل ۲- ۱۲، فرض کنید میانگین اندازه‌ی یک عنصر 900000 بیت و میانگین تقاضاها از مرورگر به سرور اصلی 15 تقاضا بر ثانیه باشد. همچنین فرض کنید زمان لازم برای آنکه یک مسیریاب یک تقاضای HTTP را از طریق اینترنت گسیل و پاسخ رسید آن را دریافت نماید 2 ثانیه باشد. اگر میانگین کل زمان لازم مجموع زمان‌های تأخیر در دستیابی و تأخیر اینترنت باشد و اگر برای میانگین تأخیر دستیابی از رابطه‌ی:

$\frac{\Delta}{(1-\Delta\beta)}$ استفاده شود که Δ میانگین زمان ارسال یک عنصر از طریق پیوند دستیابی و β سرعت ورود عناصر به پیوند دستیابی باشد، آنگاه:

الف) کل میانگین زمان دریافت تأیید که همان زمان لازم برای پاسخگویی است چقدر است؟

ب) اگر فرض شده باشد از نهان‌سازی در شبکه‌ی LAN با نرخ برخورد 0.4 استفاده گردد، آنگاه مجموع زمان پاسخگویی چقدر است؟

۱۸- یک پیوند کوتاه 10 متری را در نظر بگیرید که سرعت ارسال آن 150 bps در دو طرف می‌باشد. فرض کنید بسته‌های حامل داده دارای طول 100000 بیت می‌باشند و طول اطلاعات کنترل شامل دست‌دهی و Ack نیز 200 بیت باشد. اگر تعداد ارتباط‌های موازی N و هر ارتباط معادل $\frac{1}{N}$ از سرعت پیوند را در اختیار گیرد. با فرض آنکه از پروتکل HTTP استفاده گردد و بارگیری پایین هر عنصر نیز طولی معادل 100 کیلوبیت را اختیار کند و بارگیری آغازین شامل تعداد 10 عنصر ارجاعی از همان ارسال کننده باشد. آیا راهکار بارگیری موازی از طریق لحظه‌های موازی متناوب HTTP امکان‌پذیر است؟ اگر HTTP به‌صورت پی در پی به جای متناوب استفاده نماید آیا تغییر چشمگیری حاصل می‌شود؟ توضیح دهید.

۱۹- فرض کنید پیکربندی برنامه مشتری POP را به گونه‌ای انجام داده اید که با روش بارگیری پایین و حذف عمل می‌کند. تعامل انجام شده در ذیل را کامل کنید:

```
S: ..... blah
S:    .
?
?
```

۲۰- اگر توزیع یک فایل به حجم F = 10 Gb به تعداد N نظیر منظور باشد و سرور دارای سرعت بارگذاری μ_s = 20 Mbps باشد و هر نظیر دارای نرخ‌های بارگیری پایین d_i = 1 Mbps و بارگذاری u باشد. برای ارزش‌های N به ترتیب 10، 100 و 1000 و برای uهای به ترتیب 200، 600 و 1000 کیلوبیت بر ثانیه. جدولی را تنظیم نمایید که در آن حداقل زمان توزیع هر ترکیب N و u برای هر دو حالت توزیع مشتری- سرور و P2P نشان داده شده است؟

۲۱- با توجه به معماری P2P توزیع فایل به حجم F و تعداد نظیر N را در نظر بگیرید. برای ساده شدن موضوع فرض کنید، d_{min} بسیار بزرگ است بطوریکه سرعت بارگیری نظیرها هرگز سبب تنگراه نخواهد شد:

الف) فرض کنید $u_s \le \frac{(u_s+u_1+\ldots+u_N)}{N}$ می‌باشد. مطلوبست یک روش توزیع که زمان آن برابر $\frac{NF}{(u_s+u_1+\ldots+u_N)}$ باشد.

ب) فرض کنید $u_s \ge \frac{(u_s+u_1+\ldots+u_N)}{N}$ می‌باشد. مطلوبست یک روش توزیع که زمان آن برابر $\frac{F}{u_s}$ باشد.

ج) با توجه به موارد بالا آیا می‌توانید این نتیجه‌گیری را به عمل آورید که حداقل زمان توزیع بطورکلی از طریق رابطه‌ی $\max\left\{\frac{F}{u_s}, \frac{NF}{(u_s+u_1+\ldots+u_N)}\right\}$ بدست می‌آید.

۲۲- در این پرسش طراحی یک جایگذاری سلسله مراتبی که از سه نوع نظیر: معمولی، اَبر و اَبرمجازی استفاده می‌کند، بررسی می‌شود:

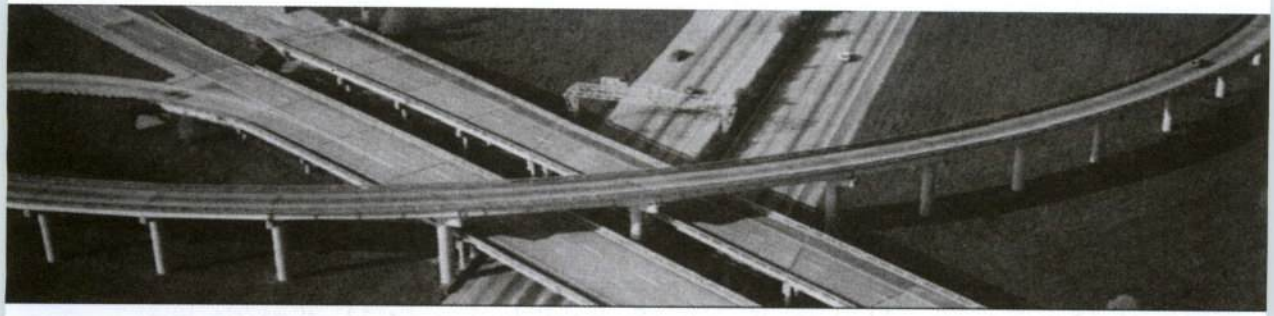
الف) اگر فرض شود هر نظیر اَبرمجازی تقریباً مسئولیت 200 نظیر اَبر و هر یک از آنها نیز شامل 200 نظیر معمولی باشند. چه تعداد نظیر از نوع اَبرمجازی برای یک شبکه شامل 4 میلیون همتا لازم است.

ب) چه اطلاعاتی را هر یک از نظیرهای: اَبر و اَبرمجازی در درون خود ذخیره می‌کنند و جستجوی انجام شده در یک طراحی سه لایه‌ای چگونه بایستی انجام گردد؟

۲۳- یک شبکه‌ی جایگذاری را در نظر بگیرید با تعداد N نظیر فعال که هر یک جفت نظیر دارای یک ارتباط TCP فعال می‌باشند. اگر فرض شود ارتباط TCP از درون تعداد N مسیریاب عبور نماید به چه تعداد گره و لبه در این شبکه‌ی جایگذار نیاز می‌باشد؟

۲۴- اکثر سامانه‌های تجارت الکترونیکی و دیگر سامانه‌های وب از یک بانک اطلاعات پشتیبان استفاده می‌کنند. توضیح دهید چگونه یک سرور HTTP با این پایگاه‌های داده‌ی پشتیبان ارتباط برقرار می‌کند؟

۲۵- آیا می‌توانید مرورگر خود را به گونه‌ای پیکربندی نمایید که چندین ارتباط همزمان به یک سامانه‌ی وب را بازگشایی نماید؟ نقاط قوت و ضعف تعداد زیادی ارتباط TCP همزمان چیست؟ توضیح دهید.



فصل ۳

لایه‌ی انتقال

با استقرار در بین لایه‌های کاربرد و شبکه، لایه‌ی انتقال یک هسته‌ی مرکزی برای معماری لایه‌بندی شبکه به‌شمار می‌آید. نقش حیاتی این لایه ایجاد تسهیلات برای ارتباط مستقیم پردازش‌های کاربردی است که بر روی میزبان‌های مختلف در حال اجرا می‌باشند. رویکرد آموزشی این فصل بر مبنای تناوب بحث پیرامون اصول لایه‌ی انتقال و چگونگی پیاده‌سازی پروتکل‌های موجود آن می‌باشد. تأکید ویژه بر روی پروتکل‌های اینترنت لایه‌ی انتقال و به‌ویژه دو پروتکل TCP و UDP خواهد بود. در آغاز سخن، بحث پیرامون ارتباط بین لایه‌های انتقال و شبکه می‌باشد. این رویکرد زمینه لازم برای پژوهش بر روی نخستین عملکرد حیاتی لایه‌ی انتقال که عبارت است از: گسترش خدمات تحویل در لایه‌ی شبکه بین دو سیستم انتهایی، و پیاده‌سازی همان خدمات بین دو پردازش لایه‌ی کاربرد که در حال اجرا بر روی سیستم‌های انتهایی می‌باشند. این عملکرد در بخش مربوط به پروتکل انتقال غیر اتصال‌گرای اینترنت UDP پوشش ویژه‌ای را به خود اختصاص داده است.

در پی این تحلیل، بازگشت به اصول و بررسی یکی دیگر از دشواری‌های بنیادی شبکه‌های کامپیوتری انجام می‌گیرد. این چالش‌ها عبارتند از: ایجاد ارتباط مطمئن برای دو موجودیت از طریق یک رسانه در شبکه که ممکن است داده‌ها را فاسد و یا تلف نماید. با کمک راهکارهای مختلف و عمل‌گرا سعی می‌شود مجموعه‌ای از روش‌ها که قابلیت کاربرد در پروتکل انتقال را داشته باشند شناسایی نموده و از این طریق اقدام به یافتن راه حلی برای آن‌ها نمود. به دنبال آن نشان داده خواهد شد چگونه این اصول در TCP رخنه داده می‌شوند. لازم به یادآوری است که TCP یک پروتکل انتقال اتصال‌گرای اینترنت است.

به دنبال این مباحث، گردش خود را متمرکز بر روی دومین چالش مهم دیگر شبکه یعنی کنترل سرعت انتقال موجودیت‌های لایه‌ی انتقال خواهیم نمود. علت بررسی این چالش جهت جلوگیری و یا بهینه‌سازی تراکم در درون شبکه می‌باشد. علل و تأثیرهای ناشی از تراکم و نیز روش‌های کنترل تراکم

از جمله موضوعات مورد بررسی در این زمینه می‌باشد. پس از آگاهی کامل از موضوعات تاثیرگذار بر روی کنترل تراکم به مطالعه رویکرد پروتکل TCP می‌پردازیم.

۳-۱ خدمات لایه‌ی انتقال

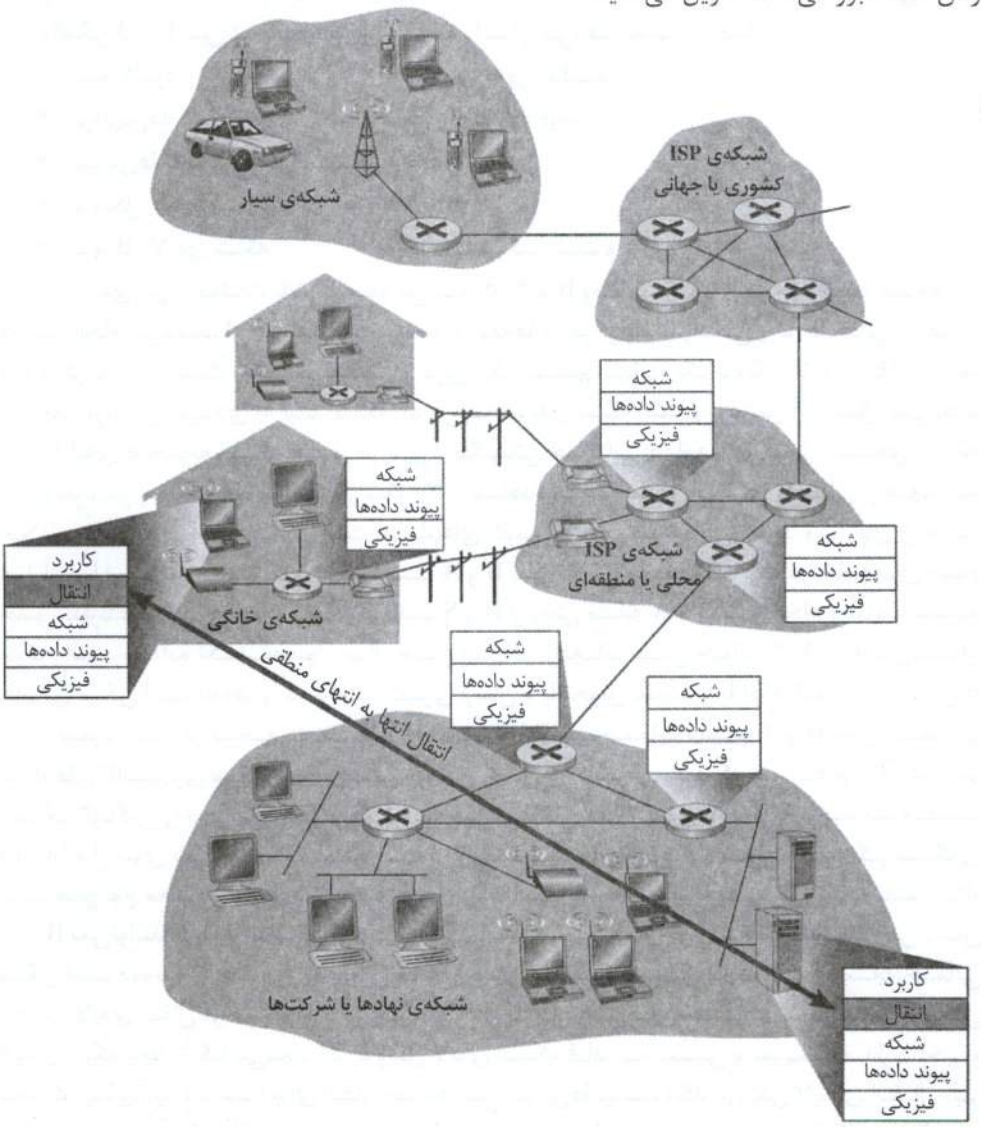
در دو فصل قبل مروری گذرا بر نقش لایه‌ی انتقال انجام شد. خدماتی که این لایه برای موجودیت‌های شبکه امکان‌پذیر می‌سازد، بررسی گردید. آنچه در دو فصل گذشته به آن اشاره شد به اختصار عبارت بود از آنکه: یک پروتکل لایه‌ی انتقال ارتباط منطقی بین پردازش‌های کاربردی در حال اجرا بر روی میزبان‌های مختلف را تسهیل می‌کند. از نقطه‌نظر یک برنامه‌ی کاربردی ارتباط منطقی به مفهوم تصور این موضوع است که هنگام اجرای پردازش‌ها دو میزبان مستقیماً با یکدیگر در ارتباط می‌باشند. در صورتی‌که در دنیای واقعی دو میزبان ممکن است در دو گوشه‌ی گیتی از طریق ده‌ها ابزار ارتباطی شبکه و پیوندهای مختلف به یکدیگر متصل شده باشند. پردازش‌های کاربردی از ارتباط‌های منطقی که از سوی لایه‌ی انتقال فراهم می‌شود جهت تبادل پیام‌ها، بین یکدیگر استفاده می‌کنند، بدون آنکه نگران جزئیات مربوط به زیر ساخت فیزیکی استفاده شده جهت حمل این پیام‌ها باشند. شکل ۳-۱ مفهوم ارتباط منطقی را نشان می‌دهد. همانگونه که از شکل ملاحظه می‌شود، پروتکل‌های لایه‌ی انتقال در سیستم‌های انتهایی پیاده‌سازی شده‌اند و نه در مسیریاب‌های شبکه. در سمت گُسیل، لایه‌ی انتقال پیام‌های دریافتی از یک برنامه کاربردی ارسال کننده را تبدیل به بسته‌های ویژه‌ی لایه‌ی انتقال می‌کند. این بسته‌ها اصطلاحاً در واژگان اینترنت **قطعه** نامیده می‌شوند. در فصل‌های گذشته اشاره گردید این قطعه ها علاوه بر آنکه حامل بخشی از پیام‌های کاربرد می‌باشند، در درون خود شامل سر پیام لایه‌ی انتقال نیز هستند. لایه‌ی انتقال قطعه‌ها را به لایه‌ی شبکه در سیستم انتهایی ارسال‌کننده، منتقل می‌کند و در این لایه قطعه در قالب داده گرام‌ها کپسول‌بندی* و به سوی مقصد گسیل می‌شوند. یادآوری این نکته ضروری است که مسیریاب‌ها فقط از اطلاعات بخش لایه‌ی شبکه استفاده می‌کنند به عبارت دیگر آن‌ها اطلاعات مربوط به محدوده‌های قطعه از لایه‌ی انتقال را که در درون داده گرام‌ها کپسول‌بندی شده‌اند پردازش نمی‌کنند. در سمت دریافت لایه‌ی شبکه اطلاعات مربوط به قطعه‌ی لایه‌ی انتقال را از داده گرام جداسازی نموده و آن را به لایه‌ی انتقال مقصد تحویل می‌دهد. لایه‌ی انتقال مقصد پس از پردازش قطعه‌های دریافتی، داده‌های درون قطعه‌ها را در اختیار پردازش برنامه‌ی کاربرد قرار می‌دهد. کاربردهای شبکه ممکن است از چندین پروتکل لایه‌ی انتقال استفاده نمایند. به‌عنوان مثال هر یک از دو پروتکل TCP و UDP در اینترنت مجموعه‌های گوناگونی از خدمات لایه‌ی انتقال را برای دسترسی به کاربردها فراهم می‌آورند.

• ارتباط بین لایه‌های انتقال و شبکه

یادآوری می‌شود لایه‌ی انتقال بدون درنگ بر فراز لایه‌ی شبکه در مجموعه‌ی پروتکلی قرار گرفته است. اگرچه هر پروتکل لایه‌ی انتقال ارتباط منطقی را بین پردازش‌های در حال اجرا بر روی میزبان‌های مختلف ایجاد می‌کند، اما یک پروتکل لایه‌ی شبکه همان ارتباط منطقی را بین میزبان‌ها نیز بوجود می‌آورد. تمایز بین این دو ارتباط اگرچه روشن است اما از اهمیت زیادی نیز برخوردار می‌باشد. برای درک دقت در تمایز آن‌ها به مثال ذیل توجه نمائید. در ذهن خود تصویر دو خانه یکی در ساحل شرقی و دیگری را در ساحل غربی که هر کدام از آن‌ها محل سکونت 12 نفر می‌باشند، مجسّم نمائید. افرادی

* - منظور از کپسول‌بندی نوعی بسته‌بندی در قالب تعریف یک لفاف یا پوشه است.

که در خانه ساحل شرقی اسکان دارند همگی با نفرات خانه‌ی ساحل غربی پیوند خویشاوندی دارند (به‌عنوان مثال خاله‌زاده‌ی یکدیگرند). این دو خانواده بسیار علاقه‌مند به تعامل از طریق نگارش با یکدیگر هستند. فرض کنید هر یک از این افراد هفته‌ای یکبار از طریق روش متداول در شبکه‌ی پست و استفاده از پاکت‌های نامه با یکدیگر نامه‌نگاری می‌کنند. در نتیجه هر خانه تعداد 144 مراسله را به خانه‌ی یکدیگر گسیل می‌کنند. در هر یک از خانه‌ها یک نفر به ترتیب به نام‌های A (ساحل غربی) و B (ساحل شرقی) مسئولیت جمع‌آوری نامه‌ها و پست کردن آن‌ها را بر عهده گرفته‌اند. بنابراین هر هفته A با مراجعه به خواهران و برادران خود نامه‌های آن‌ها را جمع‌آوری و به مامور شبکه‌ی پست که روزانه از آن خانه عبور می‌کند، تحویل می‌نماید.



شکل ۳-۱ ♦ ارتباط منطقی بین پردازش‌های کاربردی به‌وسیله پروتکل انتقال

هنگامی‌که نامه‌ها به همین خانه می‌رسد باز هم A وظیفه‌ی توزیع نامه‌های دریافتی را به برادران و خواهران خود بر عهده دارد. مشابه همین رفتار نیز بر عهده‌ی B در خانه‌ی واقع در ساحل شرقی است. در این مثال خدمات پست، ارتباط منطقی بین دو خانه را فراهم می‌آورد به بیان دیگر جا بجایی نامه‌ها از یک خانه به خانه‌ی دیگر از این طریق انجام می‌گیرد، بدون آنکه این وظیفه را بین یک شخص و دیگری انجام داده باشد. از طرف دیگر A و B هر یک نیز ارتباط منطقی در بین اقوام و بستگان خود را فراهم می‌آورند. هر یک از آن‌ها نامه‌هایی را دریافت و به خواهران و برادران خود تحویل می‌دهد. بدیهی است از نقطه‌نظر بستگان، این دو خدمات پست را ارئه می‌دهند، اگر چه A و B بخشی از یک فرآیند تبادل انتها به انتها مانند میزبان‌های انتهایی شبکه می‌باشند. مثال بالا می‌تواند روشنگر واقعیتی باشد که چگونگی ارتباط بین دو لایه‌ی انتقال و شبکه را نشان می‌دهد. به عبارت دیگر:

- پیام کاربرد = نامه‌های درون پاکت‌ها
- پردازش‌ها = خال‌زادگان – اقوام
- میزبان‌ها یا سیستم‌های انتهایی = خانه‌ها
- پروتکل لایه‌ی انتقال = A و B
- پروتکل لایه‌ی شبکه = خدمات شرکت پست و ترابری آن

در ادامه‌ی شرح جزئیات بالا، ملاحظه می‌شود که A و B وظایف خود را در چارچوب خانه‌های یکدیگر انجام می‌دهند. آن‌ها مسئول ذخیره‌سازی نامه‌ها در مرکز پست واسطی و یا جابجایی آن‌ها از یک مرکز به مرکز پست دیگر نمی‌باشند. در درون یک سیستم انتهایی یک پروتکل لایه‌ی انتقال پیام‌ها را از یک پردازش کاربردی به نبش شبکه که در اینجا لایه‌ی شبکه است و برعکس آن انتقال می‌دهد اما به ارایه‌ی هیچگونه اظهار نظری در مورد چگونگی جابجایی پیام‌ها در درون هسته‌ی شبکه نمی‌پردازد. در حقیقت همانگونه که از شکل ۳-۱ مشاهده می‌شود مسیریاب‌های واسطی از هیچگونه اطلاعات که لایه ی انتقال ممکن است به پیام‌های کاربرد افزوده باشد آگاهی ندارند و بر روی آن‌ها نیز عملیاتی را انجام نمی‌دهند. اکنون فرض کنید A و B بنابر دلایلی مانند مسافرت در خانه‌های خود حضور ندارند، در این حالت دو نفر دیگر مانند S و H از روش مشابه A و B برای انجام وظایف محوله ممکن است استفاده نکنند. به‌عنوان مثال چون این دو از لحاظ سنی جوانتر از A و B می‌باشند، بنابراین ممکن است نامه‌ها را در دفعات کمتری ارسال و یا تحویل نمایند و یا آنکه گاهی از اوقات آن‌ها را نیز مفقود کنند. در نتیجه دو نفر S و H قادر به ارایه‌ی خدمات مشابه A و B نمی‌باشند. در شبکه‌های کامپیوتری ممکن است چندین پروتکل انتقال در دسترس باشد که هر کدام از آن‌ها نیز خدمات گوناگون را برای کاربردها فراهم آورند. خدمات امکان‌پذیر از سوی A و B با توجه به خدمات قابل ارائه از سوی شبکه‌ی پست ممکن است دچار محدودیت‌هایی شود. به‌عنوان مثال اگر شبکه‌ی پست هیچ نوع محدودیت زمانی را جهت تحویل نامه‌ها بین دو خانه برای خود تعریف نکرده باشد، آنگاه A و B نمی‌توانند حداکثر تاخیری را برای تحویل نامه‌ها در بین اقوام خود مشخص نمایند. این زمان ممکن است دو، سه یا چند روز به طول انجامد و بستگی به شبکه‌ی پست دارد. با روش مشابه خدماتی که یک لایه‌ی انتقال فراهم می‌آورد در اغلب موارد از طریق روش‌های خدمات رسانی یک پروتکل لایه‌ی شبکه دچار تنگنا می‌شود. اگر پروتکل لایه‌ی شبکه قادر به تعیین و تعریف مقدار تاخیر و ضمانت پهنای باند (سرعت) برای انتقال قطعه‌ها بین میزبان‌ها نباشد، آنگاه پروتکل لایه‌ی انتقال نیز قادر نیست مقدار زمان تاخیر و یا ضمانت پهنای باند مورد نیاز پیام‌های برنامه‌ی کاربردی را بین پردازش‌های آن‌ها تعریف نماید.

برخی از خدمات ویژه از سوی لایه‌ی انتقال قابل واگذاری می‌باشد حتی هنگامی‌که پروتکل تاثیرگذار لایه‌ی شبکه خدمات مربوطه را در لایه‌ی شبکه فراهم نمی‌کند. به‌عنوان مثال یک پروتکل لایه‌ی انتقال می‌تواند سرویس انتقال مطمئن داده را به یک کاربرد پیشنهاد نماید حتی اگر پروتکل تاثیرگذار لایه‌ی شبکه دارای ویژگی غیرمطمئن باشد. به عبارت دیگر پروتکل شبکه سبب اتلاف، تحریف و یا تکرار بسته‌ها می‌شود. در مثال دیگری در فصل ۸ ملاحظه خواهد شد، چگونه پروتکل لایه‌ی انتقال با استفاده از رمزگذاری، ضمانت لازم برای آنکه پیام‌های برنامه‌های کاربردی از سوی عوامل غیر مجاز غیر قابل خواندن باشند را تعیین می‌کند. حتی در صورتی‌که لایه‌ی شبکه ضمانت محرمانه ماندن قطعه‌ها را برای لایه‌ی انتقال فراهم نکرده باشد.

بازنگری لایه‌ی انتقال در اینترنت

در بخش‌های مختلف اشاره گردید که شبکه‌ی اینترنت و به طور کلی شبکه‌های مبتنی بر TCP/IP از دو پروتکل متمایز TCP وUDP در لایه‌ی انتقال خود بهره می‌گیرند و ویژگی این دو پروتکل در آن است که یکی از آن‌ها یعنی TCP خدمات اتصال‌گرا و مطمئن و دیگری UDP خدمات غیر اتصال‌گرا و نامطمئن را برای پردازش‌ها در برنامه‌های کاربردی فراهم می‌آورند. علاوه بر این موارد همچنین توضیح داده شده است که پروتکل‌های لایه‌ی انتقال پیام‌های کاربردها را تبدیل به قطعه می‌کنند. در اینجا توضیح این نکته ضروری است که در اسناد RFC مربوط به لایه‌ی انتقال از واژه‌ی **قطعه** جهت پروتکل TCP و از واژه‌ی **داده گرام** جهت پروتکل UDP استفاده شده است، در صورتی‌که داده گرام‌ها در اصل در پروتکل لایه‌ی شبکه تشکیل می‌شوند. بدیهی است در اسناد RFC مرتبط با پروتکل‌های لایه‌ی شبکه به طور کلی تبدیل قطعه‌های پروتکل TCP و داده گرام‌های پروتکل UDP به بسته‌های کوچکتر به‌عنوان داده گرام تلقی شده است. بنابراین در ادامه‌ی مباحث این کتاب این تقسیم‌بندی رعایت می‌گردد. یعنی واژه‌ی داده گرام منحصر به عملیات پروتکل لایه‌ی شبکه است و واژه‌ی قطعه صرفاً مربوط به هر دو پروتکل TCP و UDP در لایه‌ی انتقال می‌باشد.

بدیهی است گزینش هر یک از این پروتکل‌ها در برنامه‌های کاربردی از سوی طراح کاربرد صورت می‌پذیرد و پیاده‌سازی و به کارگیری آن‌ها از طریق برنامه‌نویسی سوکت که در فصل ۲ توضیح داده شده، عملی می‌شود. قبل از بررسی پروتکل‌های لایه‌ی انتقال، ابتدا توضیحات مختصری پیرامون لایه‌ی شبکه در اینترنت و در حالت کلی شبکه‌های TCP/IP ارائه می‌شود. لایه ی پروتکل شبکه‌ی اینترنت دارای پروتکلی است به نام IP یا پروتکل اینترنت. این پروتکل یک ارتباط منطقی را بین میزبان‌ها برقرار می‌سازد. الگوی خدمات IP تحویل بر مبنای تلاش بهینه می‌باشد. به عبارت دیگر IP حداکثر کوشش لازم را در جهت تحویل قطعه‌ها بین میزبان‌های در حال ارتباط انجام می‌دهد اما ضمانتی را در راستای تحویل قطعی و سالم قطعه ها به مقصد متعهد نمی‌شود. به همین دلیل است که گفته می‌شود IP یک سرویس غیر مطمئن است. قبلاً اشاره گردید هر میزبان حداقل دارای یک آدرس لایه‌ی شبکه می‌باشد که آن را آدرس IP می‌نامند. در ادامه‌ی مباحث این فصل مبنا بر آن است که هر میزبان تنها دارای یک آدزس IP می‌باشد. اکنون تحلیل الگوی خدمات برای هر یک از دو پروتکل لایه‌ی انتقال یعنی TCP و UDP انجام می‌پذیرد. مهمترین وظیفه‌ی پروتکل‌های TCP و UDP گسترش خدمات تحویل IP از بین دو سیستم انتهایی به خدمات تحویل بین دو پردازش در حال اجرا بر روی آن‌ها می‌باشد. این گستردگی تحویل از میزبانی به میزبان دیگر و از پردازشی به پردازش دیگر اصطلاحاً

مالتی پلکس لایه انتقال[1] خوانده می‌شود. بدیهی است عکس عملیات مالتی پلکس در سوی دیگر مانند مقصد انجام می‌شود. در ادامه جزئیات بیشتری در مورد این موضوع ارایه خواهد شد. دو پروتکل TCP و UDP همچنین آزمایش یکپارچگی قطعه ها را از طریق قرار دادن اطلاعات آشکارسازی خطا در سر پیام آن‌ها امکان‌پذیر می سازند. حداقل خدمات لایه‌ی انتقال UDP عبارت است از دو سرویس تحویل داده‌ی پردازش به پردازش و آشکارسازی خطا که در بالا اشاره گردید. بنابراین می‌توان نتیجه‌گیری نمود که پروتکل UDP همانند IP یک سرویس غیر مطمئن است.

از طرف دیگر TCP علاوه بر موارد اشاره شده چندین امکان دیگر را نیز برای کاربردها فراهم می‌آورد. از جمله‌ی مهمترین این خدمات **انتقال مطمئن داده‌ها**[2] می‌باشد. با استفاده از موارد کنترل جریان، شماره گذاری ردیف، تائید و زمان‌بندی، پروتکل TCP ضمانت‌های لازم برای تحویل قطعه از سوی پردازش ارسال کننده به پردازش دریافت کننده را بدون هیچگونه تغییر در ترتیب و یکپارچگی آن‌ها فراهم می‌آورد. پروتکل TCP بنابراین سرویس غیر مطمئن IP را که بین سیستم‌های انتهایی در جریان است تبدیل به سرویس مطمئن انتقال قطعه‌ها بین پردازش‌های ارسال و دریافت می‌کند. علاوه بر این موارد TCP کنترل تراکم را نیز امکان‌پذیر می سازد. خدمات کنترل تراکم را نمی‌توان به‌عنوان یک سرویس جهت کاربردهای فراخوانی شده به‌شمار آورد بلکه این امکانات بیشتر در جهت استفاده‌ی کلی در شبکه‌ی اینترنت می باشد. به عبارت ساده‌تر کنترل تراکم TCP ، مانع انباشت بی‌مورد ترافیک بر روی پیوندها و مسیریاب‌های بین میزبان‌های در حال تعامل از سوی هر یک از ارتباط‌های در حال عبور از درون یک پیوند متراکم سهم برابری از پهنای باند را اختصاص دهد. انجام این مهم اینگونه پیاده‌سازی می‌شود که سرعت ارسال از سمت ارتباط TCP گسیل‌کننده داده‌ها را کنترل و تنظیم می‌کند. بدیهی است UDP از این مکانیزم‌های کنترل بهره‌ای نمی‌گیرد و بنابراین یک برنامه‌ی کاربردی که از UDP استفاده می‌کند با سرعت دلخواه خود قطعه‌ها را به درون پیوندها گسیل می‌کند. از آنجائیکه یک پروتکل با امکانات انتقال مطمئن داده ها و نیز کنترل تراکم لزوماً پیچیده می‌باشد، بنابراین چندین مبحث مجزا را بایستی به هر یک از این موارد اختصاص داد.

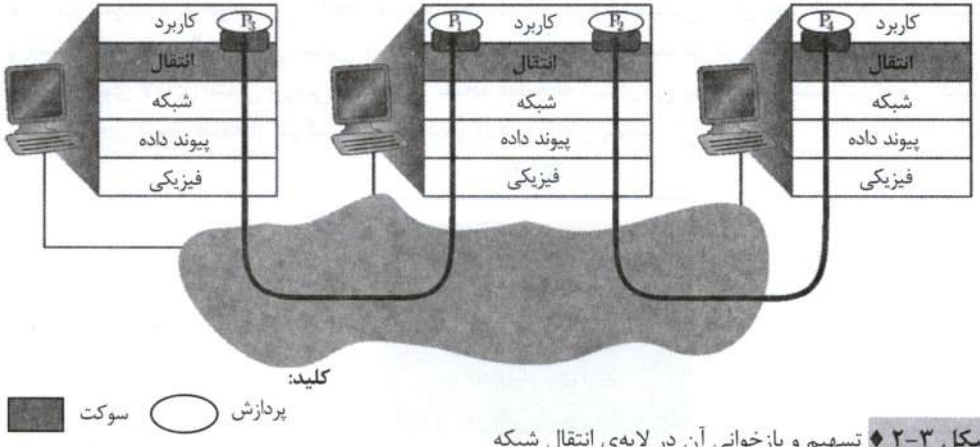
۳-۲ تسهیم (مالتی پلکس)

این بخش به بررسی مسایل مربوط به تسهیم و عکس آن اختصاص دارد. هر دو پروتکل لایه‌ی انتقال TCP و UDP از آن بهره می‌گیرند. قبلاً اشاره گردید که مفهوم تسهیم (مالتی پلکس و دی مالتی پلکس) گسترش خدمات تحویل از میزبانی به میزبان دیگر در لایه ی شبکه و پردازشی به پردازش دیگر در لایه ی انتقال شبکه می‌باشد. اگر چه در این بخش عملیات تسهیم صرفاً از نقطه‌نظر شبکه‌ی اینترنت بررسی می‌گردد اما توجه به این نکته ضروری است این خدمات به شکل دوگانه‌ی خود برای همه‌ی شبکه‌های کامپیوتری مورد نیاز می‌باشند. در میزبان مقصد لایه‌ی انتقال قطعه‌ها را از لایه‌ی شبکه که در زیر آن قرار گرفته است، دریافت می‌کند. مسئولیت لایه‌ی انتقال تحویل داده‌های درون این قطعه‌ها به پردازش‌های کاربردی مناسب در حال اجرا بر روی میزبان‌ها می‌باشد. برای شفافیت بیشتر موضوع به بیان یک مثال می‌پردازیم. فرض کنید در مقابل رایانه‌ی خود نشسته‌اید و در حال بارگیری صفحه‌های وب هستید در عین حالیکه که همزمان مشغول اجرای یک انتقال FTP و دو تعامل

1- Tran sport layer multi plexing
2- Reliable data transfer

Telnet نیز می‌باشید. به عبارت دیگر در حال اجرای چهار پردازش کاربردی هستید. دو پردازش Telnet، یک پردازش FTP و یک پردازش HTTP . هنگامی‌که لایه‌ی انتقال در رایانه‌ی شما داده‌ها را از لایه‌ی زیرین شبکه دریافت می‌کند بایستی داده‌های دریافتی را به سمت یکی از این چهار پردازش هدایت نماید. ضمن یادآوری مطالب بخش‌های ۲-۲۱ و ۲-۲۳ که در آن‌ها توضیح داده شد یک پردازش به‌عنوان بخشی از یک کاربرد شبکه می‌تواند یک و یا چند سوکت را اختیار نماید و از طریق آن‌ها داده‌ها را با استفاده از شبکه به پردازش ها تحویل داده و نیز از سمت پردازش‌ها همین عمل را به طرف شبکه انجام دهد. بنابراین بر مبنای آنچه در شکل ۳-۲ نشان داده شده است، لایه‌ی انتقال در میزبان گیرنده در حقیقت داده‌ها را مستقیماً به یک پردازش تحویل نمی‌دهد، بلکه آن‌ها را در اختیار یک سوکت میانی می‌گذارد. به دلیل آنکه در هر لحظه چندین سوکت در میزبان دریافت کننده وجود دارند. هر سوکت دارای یک شناسه‌ی منحصر به خود می‌باشد که قالب‌بندی آن شناسه وابسته به نوع سوکت است. به عبارت دیگر آن سوکت از نوع TCP و یا UDP است.

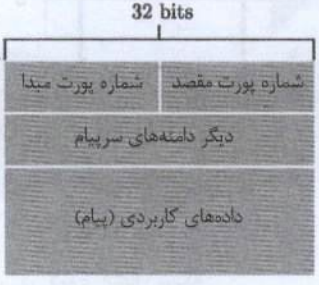
اکنون به بررسی چگونگی هدایت داده‌های ورودی از یک لایه‌ی انتقال به یک سوکت از سوی میزبان دریافت کننده‌ی قطعه‌ها می‌پردازیم. هر قطعه‌ی لایه‌ی انتقال حامل مجموعه‌ای از میدان‌ها در قطعه برای این منظور می‌باشد. در انتهای نقطه‌ی دریافت، لایه‌ی انتقال این میدان‌ها را ارزیابی و سوکت‌های دریافت کننده را شناسایی می‌کند، سپس قطعه‌ها را به آن سوکت هدایت می‌کند. این وظیفه‌ی تحویل داده‌های درون یک قطعه‌ی لایه‌ی انتقال به سوکت مناسب آن را اصطلاحاً **بازخوانی تسهیم**[1] نامیده‌اند. وظیفه‌ی جمع آوری گروه‌های مختلف داده از سوکت‌های مختلف میزبان مبدأ و کپسول‌بندی هر یک از این گروه‌ها با اطلاعات سر پیام جهت تولید قطعه‌ها و سپس گسیل قطعه‌ها به لایه‌ی انتقال را **تسهیم**[2] نامیده‌اند. بدیهی است لایه‌ی انتقال در میزبان میانی شکل ۳-۲ بایستی قطعه‌های ورودی از لایه‌ی زیرین شبکه را به یکی از پردازش‌های P_1 و P_2 در بالای آن، بازخوانی تسهیم نماید. این مهم از طریق هدایت داده‌ها در قطعه‌های ورودی به سوکت‌های پردازش مربوطه انجام می‌شود. لایه انتقال در میزبان میانی همچنین به جمع‌آوری داده‌های خروجی از این سوکت‌ها می‌پردازد.



شکل ۳-۲ ♦ تسهیم و بازخوانی آن در لایه‌ی انتقال شبکه

1- Demulti plexing
2- Multi plexing

به هنگام جمع‌آوری داده‌های خروجی لایه‌ی انتقال این میزبان، قطعه‌های مربوط به آن‌ها را تشکیل داده و سپس این قطعه‌ها را به سوی لایه‌ی شبکه گسیل می‌کند. اگرچه تسهیم و بازخوانی در چارچوب لایه‌ی انتقال شبکه اینترنت انجام می‌شود اما در حقیقت پیاده‌سازی این عملیات در هر نقطه‌ای از شبکه صورت می‌گیرد که یک پروتکل در یک لایه چه در لایه‌ی انتقال و یا مکان دیگری به‌وسیله چندین پروتکل در لایه‌ی بالاتر مورد استفاده قرار گرفته است. جهت توصیف عملکرد بازخوانی تسهیم بار دیگر مثال افراد دو خانواده یادآوری می‌شود که در آن مثال گفته شد هر یک از افراد خانواده به‌وسیله‌ی نام آن‌ها شناسایی می‌شوند. هنگامی‌که B یک گروه پیام پستی را از سوی مأمور پست تحویل می‌گیرد او در حقیقت یک عملیات بازخوانی تسهیم را انجام می‌دهد. زیرا وی مشخصات پاکت نامه را بررسی و از طریق آن دریافت کننده را شناسایی می‌نماید. در نقطه‌ی مقابل A عملیات تسهیم را انجام می‌دهد زیرا به جمع‌آوری نامه از بستگان خود می‌پردازد و پس از انجام این وظیفه گروه نامه‌ها را به مأمور پست تحویل می‌دهد. اکنون که تا حدودی نقش انجام و بازخوانی تسهیم به‌وسیله‌ی لایه‌ی انتقال مشخص گردید، چگونگی انجام این مهم عملاً در یک میزبان بررسی می‌شود. بنابر مطالب گذشته می‌توان نیازهای لایه‌ی انتقال جهت انجام عملیات تسهیم را در قالب برخورداری از : (۱) سوکت از یک شناسه‌ی منحصر به فرد و (۲) آنکه هر قطعه از محدوده‌های ویژه جهت تعریف سوکتی که قطعه بایستی به آن تحویل گردد، به‌شمار آورد. این میدان‌های ویژه در قطعه‌ها عبارتند از محدوده‌های شماره پورت‌های منبع و مقصد که در شکل ۳-۳ به تصویر کشیده شده‌اند. بدیهی است قطعه‌های TCP و UDP شامل میدان‌های دیگری نیز می‌باشند که در بخش‌های بعدی بررسی شده‌اند. هر شماره پورت یک ارزش عددی 16 بیت در محدوده‌ی صفر تا 65535 می‌باشد. از میان این ارزش‌ها، شماره پورت‌های از صفر تا 1023 اصطلاحاً **شماره پورت‌های مشهور**[1] نامیده شده‌اند که محدودیت‌هایی نیز برای آن‌ها تعریف شده است. به عبارت دیگر این پورت‌ها جهت کاربردهای ویژه مانند HTTP یا FTP و نظایر آن می‌باشند. صورت همه‌ی این شماره‌ها در RFC1700 و RFC3232 آمده است. هنگامی‌که یک کاربر جدید همانند آنچه در فصل دوم توضیح داده شد، طراحی می‌شود بایستی به هر یک شماره پورت ویژه اختصاص یابد. با استفاده از این اطلاعات است که لایه‌ی انتقال خدمات تسهیم و بازخوانی آن را انجام می‌دهد. هر سوکت در یک میزبان دارای یک شماره پورت است و هنگامی‌که یک قطعه به آن میزبان وارد می‌شود، شماره پورت مقصد که در درون قطعه تعریف شده است از سوی لایه‌ی انتقال بررسی شده و در نتیجه قطعه به سوی آن پورت هدایت می‌شود. آنگاه بخش داده‌ی درون قطعه از سوکت عبور نموده و وارد پردازش پیوست آن می‌شود.



شکل ۳-۳ ♦ محدوده‌های شماره‌ی پورت‌های مبدأ و مقصد قطعه‌ی لایه‌ی انتقال

1- Well known port numbers

- **تسهیم غیر اتصال‌گرا**

برنامه‌ی جاوا در بخش ۲-۲۳ که یک سوکت UDP را از طریق سطر ذیل ایجاد می‌نمود:

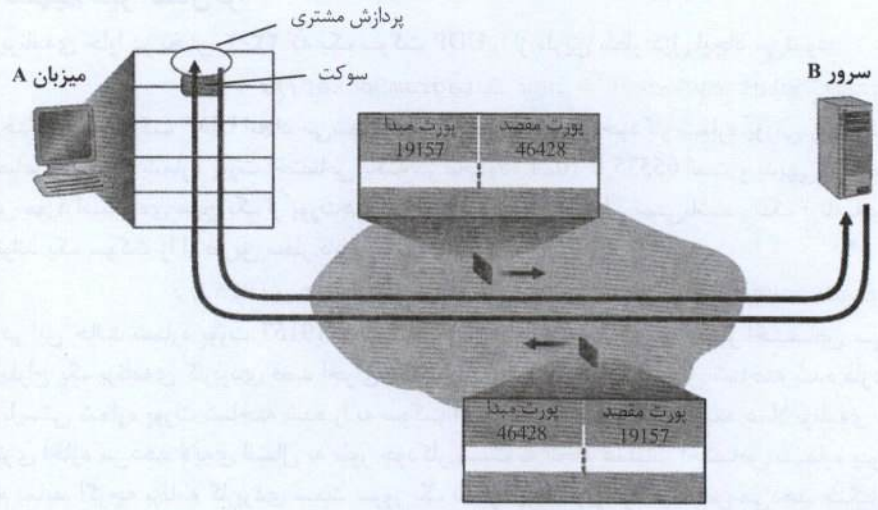
```
DatagramSocket mySocket = new DatagramSocket();
```

هنگامی‌که سوکت UDP ایجاد می‌شود، لایه‌ی انتقال به‌صورت خودکار شماره پورتی را به سوکت اختصاص می‌دهد. شماره پورت اختصاص یافته در محدوده 1024 تا 65535 است و بدیهی است که این ارزش مورد استفاده‌ی هیچ یک از پورت‌های UDP دیگر در آن میزبان نمی‌باشد. یک برنامه‌ی جاوا می‌تواند یک سوکت را از طریق سطر ذیل تعریف کند:

```
DatagramSocket mySocket = new DatagramSocket (19157);
```

در این حالت شماره پورت 19157 به سوکت UDP متعلق به کاربرد مورد نظر اختصاص می‌یابد. اگر طراح یک برنامه‌ی کاربردی قصد اجرای برنامه را در طرف سرور با پروتکل شناخته شده دارد، آنگاه وی بایستی شماره پورت شناخته شده را به سوکت آن برنامه اختصاص دهد. البته عملاً برنامه‌ی سمت مشتری اجازه می‌دهد لایه‌ی انتقال به طور خودکار نسبت به انجام عملیات اختصاص شماره پورت‌ها اقدام نماید. اگرچه برنامه کاربردی سمت سرور یک شماره پورت ویژه را اختصاص می‌دهد. هنگامی‌که یک شماره پورت به سوکت UDP اختصاص می‌یابد آنگاه می‌توان به تعریف عملیات تسهیم و بازخوانی آن برای UDP پرداخت. اگر فرض شود پردازشی در میزبان A از طریق یک پورت UDP به شماره‌ی 19157 مایل به ارسال گروه داده‌ای از یک کاربرد به پردازش دیگری در پورت UDP به شماره‌ی 46428 در میزبان B می‌باشد. لایه‌ی انتقال A یک قطعه را در قالب لایه‌ی انتقال، شامل داده‌های برنامه‌ی کاربردی ایجاد و در آن قطعه اطلاعات مربوط به پورت‌های مبدأ یعنی 19157 و مقصد 46428 را به همراه دو ارزش دیگر که در ادامه توضیح داده می‌شوند، پیوست می‌کند. لایه‌ی انتقال آنگاه قطعه‌ی نتیجه شده از این عملیات را به سمت لایه‌ی شبکه گسیل می‌دارد. این لایه قطعه‌ی دریافتی را در قالب یک داده گرام IP کپسول‌بندی نموده و تلاش بهینه‌ی خود را جهت تحویل آن به میزبان مقصد انجام می‌دهد. چنانچه قطعه در میزبان B دریافت شود آنگاه لایه‌ی انتقال آن، شماره پورت مقصد یعنی 46428 را ارزیابی و قطعه را به سوکت مربوطه که دارای چنین آدرسی است، تحویل می‌نماید. بدیهی است میزبان B ممکن است در حال اجرای چندین پردازش همزمان باشد که هر یک از آن‌ها سوکت ویژه‌ی خود را به همراه یک شماره پورت اختصاصی دارند. هنگامی‌که قطعه‌های UDP از شبکه دریافت می‌شوند، میزبان B از طریق عملیات بازخوانی تسهیم هر قطعه را به سوی سوکت مناسب آن از طریق بررسی شماره پورت مقصد هدایت می‌کند.

بدیهی است هر سوکت UDP با استفاده از دو بخش اطلاعات آدرس IP و شماره پورت مقصد شناسایی می‌شود. در نتیجه اگر دو قطعه‌ی UDP دارای آدرس‌های متفاوت IP و یا شماره پورت‌های گوناگون مبدا باشند اما آدرس‌های IP و یا شماره پورت‌های مقصد آن‌ها یکسان باشد، این دو قطعه از طریق یک سوکت مقصد مشابه به سوی همان پردازش در مقصد هدایت می‌شوند. ممکن است این دو پرسش را در ذهن خود مطرح نمائید که علت وجود شماره پورت در مبدا چیست؟ همانگونه که در تصویر ۳-۴ نشان داده شده است، در ارسال قطعه از A به B شماره پورت مبدا به‌عنوان بخشی از آدرس بازگشت عمل می‌نماید. هنگامی که B مایل به ارسال قطعه‌ای به A می‌باشد پورت مقصد در ارسال از قطعه‌ی B به A ارزش پورت مبدا قطعه ارسالی از A به B را اختیار می‌کند. بنابراین آدرس برگشت کامل عبارت است از آدرس IP متعلق به قطعه‌ی A و شماره پورت مبدا.



شکل ۳-۴ ♦ وارونه‌سازی شماره پورت‌های مبدا و مقصد

• تسهیم اتصال‌گرا

جهت درک عمیق تسهیم TCP یا به بیان دیگر تسهیم اتصال‌گرا نیاز به بررسی جامع سوکت‌های TCP و چگونگی ایجاد یک ارتباط TCP می‌باشد. یک تفاوت بنیادی که بین یک سوکت TCP و همتای UDP آن وجود دارد آن است که، یک سوکت TCP از طریق یک مجموعه‌ی چهار عنصری شامل آدرس IP مبدا، شماره پورت مبدا، آدرس IP مقصد و شماره پورت مقصد شناسایی می‌شود. بنابراین هنگامی‌که یک قطعه‌ی TCP از شبکه به یک میزبان وارد می‌شود از همه‌ی این چهار ارزش میزبان جهت بازخوانی تسهیم برای هدایت قطعه به سوی سوکت مناسب آن استفاده می‌کند. برای درک بهتر موضوع بار دیگر به برنامه‌نویسی مشتری – سرور TCP در بخش ۲-۲۱ اشاره می‌شود:

- برنامه‌ی سرور TCP در انتظار ایجاد ارتباطی می‌ماند که از سوی یک برنامه‌ی مشتری TCP بر روی پورت 6789 درخواست می‌گردید. فرآیند انتظار از یک سوکت خوش‌آمدگو بر اساس شکل ۲-۲۹ بهره می‌گرفت.
- با استفاده از سطر ذیل برنامه‌ی مشتری TCP به تعریف یک سوکت می‌پرداخت که ارتباط را ایجاد می‌کند.

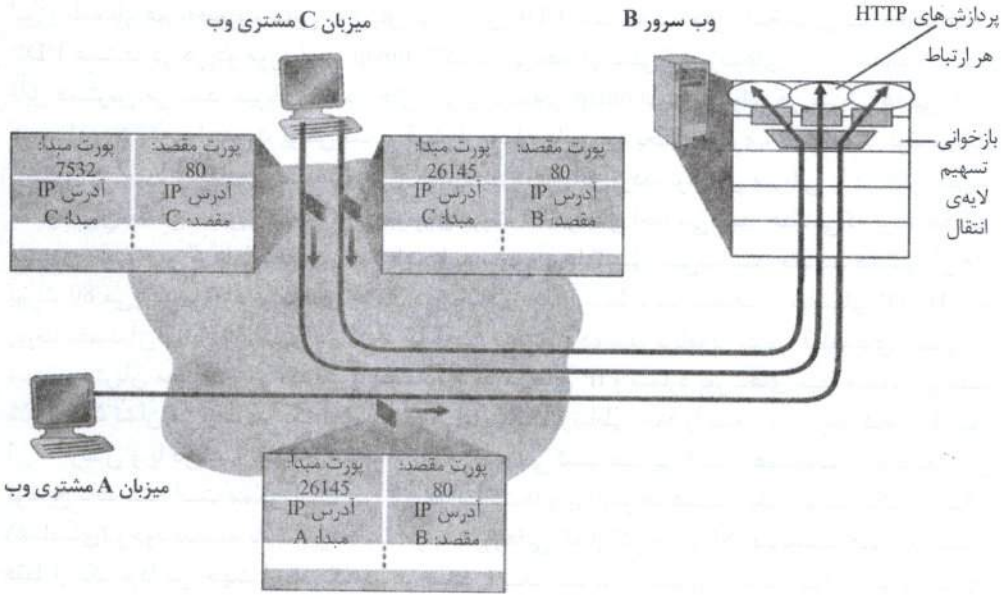
```
Socket clientSocket = new Socket ("server Host Name",6789);
```

- آنچه که یک قطعه‌ی درخواست ارتباط نام دارد عبارت است از یک قطعه‌ی TCP شامل شماره پورت مقصد 6789 و یک مجموعه بیت ایجاد ارتباط در سر پیام TCP. قطعه هم چنین حامل اطلاعات شماره پورت مبدا بر اساس گزینش برنامه‌ی مشتری می‌باشد. سطر بالا علاوه بر موارد اشاره شده به تعریف یک سوکت TCP می‌پردازد که از طریق آن پردازش مشتری قادر به ارسال و دریافت داده‌ها می‌شود.
- هنگامی‌که سیستم عامل یک میزبان در حال اجرای پردازش سرور قطعه‌ای را دریافت می‌کند که متقاضی ایجاد ارتباط با شماره پورت مقصد 6789 می‌باشد، برنامه‌ی سروری که منتظر پذیرش این ارتباط بر روی پورت 6789 می‌باشد را شناسایی می‌کند. برنامه‌ی سرور شناسایی شده آنگاه یک سوکت جدید را به شکل ذیل تعریف می‌کند:

```
Socket connectionSocket = welcomeSocket.accept();
```

- لایه‌ی انتقال در برنامه‌ی سرور، چهار ارزش ذیل را در قطعه‌ی متقاضی ارتباط ملاحظه می‌کند: (۱) شماره پورت مبدا در قطعه، (۲) آدرس IP میزبان مبدا، (۳) شماره پورت مقصد در قطعه و (۴) آدرس IP متعلق به خود. آنگاه سوکت ارتباطی تعریف شده‌ی جدید از طریق این چهار ارزش شناسایی می‌شود. تمامی قطعه‌های بعدی که پورت مبدا، آدرس مبدا، پورت مقصد و آدرس IP مقصد آن‌ها قابل نگاشت با چهار ارزش نامبرده باشند در درون این سوکت جدید بازخوانی می‌شوند. با ایجاد این ارتباط TCP ، اکنون برنامه‌های مشتری و سرور به مبادله‌ی داده‌ها بین یکدیگر می‌پردازند.

میزبان سرور ممکن است قادر به پشتیبانی تعداد قابل توجهی سوکت TCP همزمان باشد. هر یک از این سوکت‌ها نیز با یک پردازش پیوند دارند. البته هر سوکت نیز از طریق مجموعه‌ی چهارگانه‌ی خود قابل شناسایی می‌باشد. هنگامی‌که قطعه‌ی TCP وارد یک میزبان می‌شود، از تمامی چهار محدوده‌ی آن جهت هدایت قطعه به سوکت مناسب آن استفاده می‌شود. شکل ۳-۵ حالتی را نشان می‌دهد که میزبان C مایل به دو تعامل HTTP با سرور B می‌باشد و میزبان A نیز یک تعامل HTTP را با B برقرار می‌کند. هر سه میزبان : A ، C و نیز سرور B هر کدام آدرس IP انحصاری خود را به ترتیب در قالب A، C و B دارند. میزبان C دو شماره پورت مبدا متفاوت 26145 و 7532 را به دو ارتباط HTTP اختصاص داده است. از آنجائیکه میزبان A گزینش شماره پورت‌های مبدا را مستقل از C انجام داده است این میزبان ممکن است یک شماره پورت مبدا 26145 را نیز به ارتباط HTTP خود اختصاص دهد. این موضوع سبب بروز مشکلی نخواهد شد زیرا سرور B هنوز هم قادر به بازخوانی تسهیم دو ارتباط با شماره پورت مبدا یکسان می‌باشد، چرا که دو ارتباط نامبرده دارای آدرس‌های IP مبدا متفاوت می‌باشند.



شکل ۳-۵ ♦ دو برنامه‌ی مشتری از یک شماره پورت مشترک (80) در مقصد برای ارتباط با یک برنامه‌ی وب سرور مشابه استفاده می‌کنند.

۳-۳ پویش پورت

ملاحظه شد یک پردازش سرور صبورانه در انتظار تماسی می‌ماند که بر روی یک پورت آزاد از سوی یک برنامه مشتری انجام می‌شود. برخی از پورت‌های ویژه برای کاربردهایی مانند وب، FTP، DNS، SMTP می‌باشند و برخی دیگر به‌صورت قراردادی مورد استفاده‌ی کاربردهای پرطرفداری واقع می‌شوند مانند برنامه‌ی سرور SQL2000 ، که بر روی یک پورت UDP با آدرس 1434 عمل می‌کند. بنابراین اگر پورتی بر روی یک میزبان آزاد باشد این امکان وجود دارد که بتوان آن را به یک کاربرد ویژه‌ی در حال اجرا بر روی آن میزبان نگاشت نمود. این موضوع به‌ویژه برای مدیران سیستم بسیار سودمند است زیرا آن‌ها در اغلب موارد مایل هستند از وضعیت برنامه‌های کاربردی در حال اجرا بر روی میزبان‌های شبکه‌ی خود آگاهی داشته باشند. از طرف دیگر مهاجمان نیز مایل به شناسایی پورت‌های آزاد بر روی میزبان‌های هدف می‌باشند. بدیهی است اگر میزبانی در حال اجرای یک برنامه‌ی کاربردی با یک نقطه ضعف امنیتی باشد، به‌عنوان مثال اگر SQL در حال ارزیابی پورت 1434 است و در هنگام انجام این عملیات با یک سرریزی بافر روبه‌رو شود آنگاه این امکان را برای یک کاربر دور دست فراهم می‌آورد که یک برنامه‌ی دلخواهی را بر روی آن میزبان به اجرا درآورده و آنرا از این طریق در اختیار خود قرار دهد. اتفاقی که از طریق کرم اسلامر بر روی میزبان‌های آسیب‌پذیر در سال 2004 میلادی روی داد، دستیابی به اطلاعاتی از قبیل اینکه کدام برنامه‌ی کاربردی در حال گوش نمودن به کدام پورت است، بسیار ساده می‌باشد. برنامه‌های همه‌گانی متعددی در اینترنت وجود دارند که اصطلاحاً آن‌ها را **پویشگران پورت**[1] می‌نامند. آن‌ها این وظیفه را به خوبی انجام می‌دهند. یکی از محبوب‌ترین این برنامه‌ها nmap می‌باشد که به‌صورت آزاد از طریق سامانه‌ی: http://insecure.org/nmap قابل دستیابی است و در بسیاری از نسخه‌های لینوکس وجود دارد. در پروتکل TCP برنامه‌ی nmap به پویش پورت‌ها می‌پردازد و آن دسته از پورت‌هایی را شناسایی می‌کند که ارتباط‌های TCP را پذیرش می‌کنند. برای پروتکل UDP این برنامه باز هم به‌صورت پیاپی به دنبال پورت‌های UDP است که در حال پاسخگویی به قطعه‌های UDP هستند. در هر دو مورد آنچه nmap بازگشت می‌دهد فهرستی از پورت‌های آزاد، مسدود و یا غیر قابل دسترس می‌باشد. میزبانی که در حال اجرای برنامه‌ی nmap است می‌تواند هر میزبان هدفی را در هر نقطه‌ی شبکه‌ی اینترنت پویش نماید. قبل از پایان دادن به بحث جاری، لازم است توضیحات بیشتری در مورد استفاده‌ی وب سرورها از شماره پورت‌ها ارایه گردد. برنامه‌ی میزبانی را در نظر بگیرید که از طریق یک وب سرور مانند آپاچی بر روی پورت 80 در حال اجرا می‌باشد. هنگامی‌که برنامه‌های مشتری مانند مرورگرها، قطعه‌های اطلاعات را به سوی آن‌ها گسیل می‌نمایند. مقصد همگی آن‌ها پورت 80 می‌باشد به‌ویژه قطعه‌های حامل درخواست ایجاد ارتباط و نیز پیام‌های تقاضای HTTP که پورت مقصد آن‌ها نیز 80 است. همانگونه که اخیراً توضیح داده شد، برنامه‌ی سرور، قطعه‌های ارسالی از سوی مشتریان مختلف را از یکدیگر با استفاده از آدرس‌های IP و شماره پورت‌های مبدا متمایز می‌کند. شکل ۳-۵ نشان می‌دهد هر یک از این پردازش‌ها سوکت ارتباطی خود را شامل می‌شوند که از طریق آن به ارسال و یا دریافت پیام‌های تقاضا و پاسخ اقدام می‌کنند. بدیهی است همانگونه که قبلاً نیز توضیح داده شده است ممکن است بین ارتباط سوکت‌ها و پردازش‌ها همیشه یک پیوند یک به یک (همانندی) وجود نداشته باشد. در حقیقت وب سرورهایی که از کارآمدی بالا بهره‌مند هستند اغلب فقط از یک پردازش جهت ایجاد یک پیوند جدید با یک سوکت ارتباطی جدید برای هر برنامه‌ی

1- Port scanners

مشتری تازه وارد استفاده می‌کنند. برای هر برنامه‌ی سرور ممکن است در هر لحظه تعدادا قابل توجهی سوکت‌های ارتباطی با شناسه‌های مختلف وجود داشته باشد که همگی وابسته به یک پردازش مشابه می‌باشند. اگر هر دو برنامه‌ی مشتری و سرور از ارتباط HTTP پایا استفاده می‌کنند، آنگاه در تمامی طول ارتباط پایا هر دو برنامه مبادله‌ی پیام‌های HTTP را از طریق یک سوکت سرور مشابه انجام می‌دهند. اما اگر آنها از یک ارتباط HTTP نا پایا (غیر ماندگار) استفاده می‌کنند، آنگاه برای هر پیام درخواست و پاسخ یک ارتباط جدید TCP ایجاد می‌شود که حاصل آن تعریف یک سوکت جدید است که پس از انجام هر تبادل مسدود می‌گردد. واضح است اگر دایماً سوکت‌های جدید تعریف و سپس محدود شوند تاثیر آن بر روی کارایی وب سرورهایی که بسیار پرکار می‌باشند بسیار چشمگیر خواهد بود. برخی از سیستم‌عامل‌ها از راهکارهای ویژه‌ای جهت جلوگیری از این تاثیرگذاری بهره می‌جویند. اکنون که موضوعات مربوط به مالتی پلکس (تسهیم) و بازخوانی تسهیم (دی مالتی پلکس) در لایه‌ی انتقال بررسی گردید. در ادامه یکی دیگر از پروتکل‌های این لایه یعنی UDP تحلیل می‌شود.

۳-۴ انتقال غیراتصال‌گرا (UDP)

در این بخش چگونگی عملکرد پروتکل UDP بررسی می‌شود. در یک تصور کلی UDP واسط مابین دو لایه‌ی شبکه و کاربرد است. بنابراین وظیفه آن را می‌توان به‌عنوان یک واسط انتقال اینگونه ارزیابی نمود که یک پروتکل لایه انتقال مانند UDP در سمت ارسالی بایستی پیامی را از پردازشی در لایه کاربرد دریافت و سپس آن را مستقیماً به سوی لایه‌ی شبکه هدایت نماید و در جهت مقابل پیام‌ها را از پردازشی در لایه‌ی شبکه دریافت و آن‌ها را مستقیماً به سوی پردازشی در لایه‌ی کاربرد گسیل نماید. اما از مباحث گذشته چنین برمی‌آید که آنچه لایه‌ی انتقال بایستی انجام دهد فراتر از این موارد ساده است. حداقل انتظاری که از یک پروتکل لایه‌ی انتقال می‌رود آن است که خدمات تسهیم و بازخوانی تسهیم را برای گذر داده‌ها بین لایه‌ی شبکه و یک پردازش مطلوب در لایه‌ی کاربرد فراهم آورد. پروتکل UDP تعریف شده در مستند RFC768 حداقل وظایف لایه‌ی انتقال را تعریف نموده است. آنچه در این سند در رابطه با لایه‌ی انتقال تعریف شده است، علاوه بر عملیات تسهیم و بازخوانی و نیز حداقل امکانات جهت آشکارسازی خطا اطلاعات اضافی دیگری را به بسته‌های IP اضافه نمی‌نماید. در حقیقت اگر یک طراح به دنبال گزینش UDP جهت استفاده در یک برنامه‌ی کاربردی به جای TCP می‌باشد، بایستی آگاه به این موضوع باشد که برنامه‌ی کاربردی وی تقریباً به‌صورت مستقیم با IP تعامل برقرار می‌نماید. به بیان دیگر UDP پیام‌ها را از پردازش کاربردی دریافت و محدوده‌های متعلق به شماره پورت‌های مبدا و مقصد را جهت خدمات تسهیم و بازخوانی آن به همراه دو میدان کوچک دیگر به پیام‌ها اضافه می‌نماید و سپس قطعه‌ی حامل را به سمت لایه‌ی شبکه گسیل می‌کند. لایه‌ی شبکه قطعه‌های ارسالی از سوی لایه‌ی انتقال را در قالب داده گرام‌های IP کپسول‌بندی نموده و حداکثر تلاش خود را معطوف تحویل آن‌ها به میزبان مقصد می‌کند. چنانچه قطعه از سوی میزبان مقصد دریافت گردید، پروتکل UDP از شماره پورت مقصد برای تحویل قطعه به پردازش کاربردی مورد نظر بهره می‌گیرد. بدیهی است با استفاده از UDP عملیات دست‌دهی بین پردازش‌های دو لایه‌ی انتقال، ارسال و دریافت کننده انجام نمی‌شود. به همین دلیل است که UDP را پروتکل غیر اتصال‌گرا نامیده‌اند.

پروتکل DNS نمونه‌ای از یک پروتکل لایه‌ی کاربرد است که از UDP استفاده می‌کند. هنگامی‌که کاربرد DNS در یک میزبان مایل به تعریف یک درخواست می‌باشد. آن کاربرد پیام درخواست را تولید

و آن را به UDP منتقل می‌کند، بدون آنکه هیچگونه عملیات دست‌دهی را با UDP در حال اجرا در میزبان مقصد انجام دهد. پروتکل UDP سمت میزبان فقط سر پیام‌ها را به پیام اضافه می‌نماید و سپس قطعه‌ی حاصل شده را به لایه‌ی شبکه گسیل می‌کند. لایه‌ی شبکه نیز قطعه را در قالب داده گرام‌های IP بسته‌بندی نموده و آن‌ها را به برنامه‌ی سرور DNS تحویل می‌نماید. پس از آن برنامه‌ی کاربرد DNS میزبان مبدا منتظر دریافت پاسخ می‌شود. چنانچه احتمالاً در اثر اتلاف داده گرام‌ها پاسخی را دریافت ننمایند آنگاه ممکن است یکی از دو راهکار ذیل را انجام دهد: (۱) تلاش در جهت ارسال درخواست به سرورهای DNS دیگر و (۲) آگاه نمودن برنامه‌ی کاربردی مورد پرسش از این موضوع که پاسخی را از جانب وی دریافت نکرده است. اکنون ممکن است این پرسش مطرح گردد که چرا یک طراح برنامه‌ی کاربردی به جای UDP از TCP استفاده نمی‌کند؟ و آیا TCP به دلیل آنکه انتقال مطمئن را امکان‌پذیر می‌سازد در مقابل UDP مناسب‌تر نمی‌باشد؟ پاسخ به این پرسش‌ها منفی است زیرا بنا به دلایل ذیل برای بسیاری از کاربردها استفاده از UDP مناسب‌تر است:

۱- **کنترل بهینه در سطح کاربرد از لحاظ زمان و نوع داده‌ی ارسالی**. در شرایط استفاده از UDP بلافاصله پس از آنکه یک پردازش کاربردی داده‌ها را به UDP تحویل نماید. این پروتکل داده‌ها را تبدیل به قطعه‌های UDP نموده و بی‌درنگ آن‌ها را به سوی لایه‌ی شبکه گسیل می‌کند. اما پروتکل TCP از سوی دیگر از مکانیزم کنترل تراکم بهره می‌گیرد که سرعت لایه‌ی انتقال TCP در سمت ارسال کننده را هنگامی‌که یک یا چندین پیوند بین میزبان‌های مبداو مقصد دچار تراکم شدید شده‌اند، کند می‌کند. پروتکل TCP هم چنین به ارسال مجدد قطعه ها تا دریافت پاسخ تائید از سوی مقصد بدون توجه به طول زمان انتقال مطمئن آن‌ها ادامه می‌دهد. کاربردهای بی‌درنگ غالباً نیازمند یک حداقل سرعت و نیز حداقل زمان تاخیر انتقال برای قطعه‌ها می‌باشند و از طرف دیگر تا حدودی اتلاف داده‌ها برای آن‌ها قابل تحمل است.

۲- **عدم ایجاد ارتباط**. در ادامه‌ی این فصل توضیح داده شده است که TCP از روش دست‌دهی سه جانبه قبل از شروع انتقال داده‌ها استفاده می‌کند در حالیکه UDP بی‌درنگ به انجام این عملیات اقدام می‌کند. در نتیجه UDP سبب معرفی هیچگونه تاخیری برای ایجاد ارتباط نمی‌شود. شاید علت اصلی آنکه چرا DNS بر روی UDP به جای TCP اجرا می‌شود نیز همین موضوع باشد. سرعت اجرای DNS بر روی TCP به مراتب کندتر خواهد شد. از طرف دیگر از آنجائیکه انتقال مطمئن، نیاز اصلی صفحات وب متنی می‌باشد بنابراین پروتکل HTTP مجبور است از TCP به جای UDP استفاده نماید. در نتیجه یک ارتباط TCP سبب معرفی تاخیر در HTTP می‌شود که تاثیر قابل توجهی را به هنگام انجام عملیات بارگیری اسناد وب از خود به جای می‌گذارد.

۳- **موقعیت بدون ارتباط**. پروتکل TCP موقعیت‌های انتقال را در سیستم‌های انتهایی نگهداری می‌کند. این وضعیت ارتباط‌ها شامل بافرهای ارسال و دریافت، عوامل کنترل تراکم و زنجیره و شماره‌ی تائیدیه‌ها می‌باشند. در بخش ۳-۱۳ ملاحظه خواهد شد که این وضعیت‌های اطلاعاتی جهت پیاده‌سازی خدمات انتقال مطمئن داده‌ی TCP و نیز فراهم آوری مکانیزم‌های کنترل تراکم حیاتی هستند. از طرف دیگر UDP به نگهداری وضعیت‌های ارتباطی نمی‌پردازد و هیچ یک از عوامل نامبرده را نیز رهگیری نمی‌کند. به همین دلیل یک برنامه‌ی سرور اختصاص یافته به یک کاربرد ویژه می‌تواند بسیاری از برنامه‌های فعال مشتری را که در حال اجرا بر روی UDP می‌باشند، پشتیبانی نماید.

۴- **سرآیند اندک سر پیام بسته**. قطعه‌های TCP دارای 20 بایت سرآیند سر پیام به ازای هر قطعه می‌باشند در حالی که این سرآیند برای قطعه‌های UDP برابر 8 بایت است. شکل ۳-۶ فهرستی از کاربردهای محبوب اینترنتی و پروتکل‌های لایه‌ی انتقال مورد استفاده‌ی آن‌ها را نشان می‌دهد. همانگونه که انتظار می‌رود کاربردهایی مانند پست الکترونیکی، دستیابی راه دور، وب و انتقال فایل بر روی TCP پیاده‌سازی می‌شوند و همگی آن‌ها نیازمند انتقال مطمئن داده از سوی TCP می‌باشند. بدیهی است بسیاری از کاربردهای مهم دیگری نیز وجود دارند که از طریق UDP اجرا می‌شوند. از این پروتکل همچنین برای به روز رسانی جداول مسیریابی نیز استفاده می‌شود. از آنجائیکه اطلاعات به روزرسانی شده‌ی RIP به‌صورت متناوب هر 5 دقیقه یکبار ارسال می‌شوند در نتیجه اطلاعات به روز رسانی شده‌ای که تلف می‌شوند با نوع جدید آن‌ها جایگزین می‌گردند و عملاً تاثیر اتلاف آن‌ها بی اثر می‌شود. از UDP همچنین برای حمل داده‌های مدیریت شبکه (SNMP) استفاده می‌شود. علت ترجیح UDP نسبت به پروتکل TCP از آن جهت است که اغلب در مواردی که شبکه در وضعیت فشار قرار دارد و انتقال مطمئن و کنترل تراکم با دشواری رو به رو می‌شود، نیاز به کاربردهای مدیریت شبکه بیشتر می‌شود. در شکل ۳-۶ نشان داده شده است که از هر دو پروتکل TCP و UDP امروز در کاربردهای چند رسانه‌ای مانند تلفن اینترنتی، کنفرانس‌های ویدیویی بی‌درنگ و بارگیری و بارگذاری فایل‌های حامل صدا و تصویر استفاده می‌شود. این کاربردها در فصل ۷ بیشتر مورد بررسی قرار می‌گیرند.

کاربرد	پروتکل لایه‌ی کاربرد	پروتکل انتقال اصلی
پست الکترونیکی (ایمیل)	SMTP	TCP
دستیابی راه دور	Telnet	TCP
وب (web)	HTTP	TCP
انتقال فایل (FTP)	FTP	TCP
سرور راه دور	NFS	به ویژه UDP
جریان پخش تصاویر چند رسانه‌ای	خصوصی	UDP یا TCP
تلفن اینترنتی	خصوصی	UDP یا TCP
مدیریت شبکه	SNMP	به ویژه UDP
پروتکل مسیریابی	RIP	به ویژه UDP
ترجمه‌ی نام	DNS	به ویژه UDP

شکل ۳-۶ ♦ کاربردهای اینترنت پرطرفدار و پروتکل انتقال آن‌ها

اخیراً توضیح داده شد، همگی این کاربردها قادرند که اتلاف تعداد اندکی از بسته‌های خود را تحمل کنند. در نتیجه انتقال مطمئن داده برای انجام مطلوب وظایف این کاربردها چندان حیاتی نیست. علاوه بر همه‌ی این موارد کاربردهای زمان واقعی مانند تلفن اینترنتی و یا کنفرانس تصویری در برابر اعمال کنترل تراکم ترافیک TCP دارای عملکرد بسیار ضعیف می‌باشند. به همین منظور طراحان برنامه‌های کاربردی چندرسانه‌ای پیاده‌سازی کاربردهای خویش را بر روی UDP ترجیح می‌دهند. یادآوری این نکته نیز ضروری است که به گونه‌ی فرآینده‌ای از TCP جهت انتقال یکپارچه‌ی اطلاعات رسانه‌ای نیز

استفاده می‌شود. به‌عنوان مثال بر اساس مطالعه‌ی انجام شده در سال 2004 میلادی نزدیک به 75% اطلاعات رسانه‌ای زنده و بر اساس تقاضا از پروتکل TCP جهت انتقال اطلاعات خود استفاده نموده‌اند. هنگامی‌که نرخ اتلاف بسته‌ها اندک باشد و برخی از نهادهای تجاری و غیرتجاری نیز ترافیک جاری از UDP را به دلیل مسایل امنیتی مسدود نمایند آنگاه پروتکل TCP برای انتقال اینگونه اطلاعات از دید طراحان برنامه‌های کاربردی جذاب جلوه می‌نماید. پروتکل UDP از مکانیزم‌های کنترل تراکم بی‌بهره است اما از طرف دیگر وجود این ویژگی برای جلوگیری از پدیدار شدن وضعیت تراکم در شبکه حائز اهمیت می‌باشد. اگر فرض شود هر برنامه‌ی کاربردی چند رسانه‌ای جریان اطلاعات خود را با سرعت بالا و بدون استفاده از کنترل تراکم وارد شبکه نماید آنگاه تعداد اندکی از بسته‌های UDP موفق به عبور از مسیر بین مبدا تا مقصد خواهند شد. علت این عدم موفقیت به واسطه‌ی سرریزی بوجود آمده از روترها می‌باشد. نرخ قابل توجه اتلاف که در نتیجه‌ی عدم اعمال کنترل UDP در سمت ارسال کننده‌ها وجود دارد سبب می‌شود فرستنده‌های TCP در هنگام روبه‌رو شدن با تراکم، به شدت از سرعت انتقال خود کاسته و در نتیجه‌ی عدم کنترل تراکم در UDP نرخ اتلاف بسته‌ها به شدت بین فرستنده و گیرنده‌ی UDP افزایش یابد. این موضوع خود سبب می‌شود که سرعت انتقال TCP تقریباً به صفر نزدیک شود که از نقطه‌نظر شبکه بسیار ناخوشایند است. به همین دلیل بسیاری از پژوهشگران روش‌های سازگاری کنترل تراکم را برای همه‌ی موارد از جمله UDP پیشنهاد داده‌اند. این پیشنهاد در مستند RFC 4340 به تفصیل توضیح داده شده است.

قبل از ورود به بحث ساختار قطعه در UDP یادآوری این نکته ضروری است که قبلاً عنوان گردید یک برنامه‌ی کاربردی می‌تواند به هنگام استفاده از UDP از خدمات انتقال مطمئن داده نیز بهره‌مند باشد، در صورتی‌که ویژگی قابلیت اطمینان در بافت آن کاربرد تعریف شده باشد. به‌عنوان مثال با اضافه نمودن مکانیزم تائید و ارسال مجدد این مهم قابل دستیابی است. با بهره‌مندی از این ویژگی یک برنامه‌ی کاربردی می‌تواند با اطمینان و بدون تحمل محدودیت سرعت انتقال که از سوی روش‌های کنترل تراکم TCP اعمال می‌شود، به انتقال خود ادامه دهد.

۳-۵ ساختار قطعه‌ی UDP

ساختار قطعه‌ی UDP در شکل ۳-۷ نشان داده شده است. این ساختار بر اساس مستند RFC 768 است. بخش اطلاعات کاربردی در میدان داده در هر قطعه‌ی UDP قرار می‌گیرد. به‌عنوان مثال برای DNS این بخش شامل پیام‌های پرسش و پاسخ آن‌ها می‌باشد و برای کاربردهای یکپارچه مانند صدا و تصویر، بخش‌هایی از این اطلاعات در محدوده‌ی داده در هر قطعه جایگذاری می‌شود. سر پیام UDP حامل چهار میدان است که طول هر کدام نیز 2 بایت است.

همانگونه که در بخش قبل توضیح داده شد، شماره پورت‌ها اجازه‌ی انتقال داده‌های کاربردی را از سمت میزبان مقصد به پردازش مورد نظر در حال اجرا بر روی سیستم‌های انتهایی مقصد می‌دهند. به بیان دیگری عملیات مربوط به بازخوانی تسهیم را انجام می‌دهند. از یک کد **مجموع مقابله**[1] میزبان دریافت کننده‌ی داده جهت تشخیص خطا در قطعه استفاده می‌کند. این کد در واقع علاوه بر UDP از سوی چندین محدوده‌ی دیگر در سر پیام IP نیز محاسبه می‌شود که در این مقطع به تحلیل آن‌ها

1- Checksum

نمی‌پردازیم. **محدوده‌ی طول**[1] یا اندازه در سر پیام UDP، مجموع طول قطعه را به همراه سر پیام آن در قالب بایت مشخص می‌کند.



شکل ۳-۷ ♦ ساختار قطعه‌ی UDP

- **کد مجموع مقابله‌ای**

قبلاً اشاره گردید این کد جهت آشکارسازی خطا می‌باشد. برخی از بیت‌های درون قطعه‌ی UDP ممکن است در اثر اختلال‌های موجود در پیوندها مانند نویز و یا هنگام ذخیره‌سازی آن‌ها در مسیریاب‌ها دچار تغییر شوند. این تغییر معمولاً به هنگام عبور قطعه از مبدا به مقصد روی می‌دهد. پروتکل UDP برای محاسبه‌ی کد مجموع مقابله در سمت فرستنده نتیجه‌ی حاصل از مجموع تمامی کلمه‌های 16 بیت موجود در یک قطعه ی UDP را از طریق مکمل یک آن محاسبه می‌کند و در صورتی‌که نتیجه‌ی حاصل جمع سبب سرریزی شود، حاصل جمع مجدداً تکرار می‌شود. حاصل نهایی عملیات در میدان کُد مجموع مقابله در قطعه‌ی UDP جایگذاری می‌شود. در ادامه مثالی از چگونگی انجام محاسبه‌ی کُد مجموع مقابله آمده است. جزئیات مربوط به پیاده‌سازی مؤثر محاسبه در مستند RFC 1071 توضیح داده شده است. اگر فرض شود سه کلمه‌ی 16 بیت به‌صورت ذیل در درون یک قطعه‌ی UDP وجود دارد. آنگاه چگونگی محاسبه‌ی کد مجموع مقابله چنین است:

0110011001100000 سطر (۱)

0101010101010101 سطر (۲)

1000111100001100 سطر (۳)

مجموع دو کلمه‌ی 16 بیت سطرهای 1 و 2 برابر است با:

1011101110110101

و حاصل جمع این نتیجه و سطر 3 برابر است با:

```
1011101110110101
1000111100001100
----------------
0100101011000010
```

ملاحظه گردید که آخرین عملیات جمع سرریزی در پی داشت که مجدداً عملیات جمع تکرار می‌شود. برای بدست آوردن کد مجموع مقابله اکنون بایستی حاصل جمع نهایی تبدیل به مکمل یک

1- Length field

شود به عبارت دیگر تمامی 16 بیت حاصل جمع وارونه شوند. نتیجه‌ی حاصل از این وارونه‌سازی در قالب عبارت ذیل خلاصه می‌شود:

1011010100111101

این عبارت همان کد مجموع مقابله (Checksum) می‌باشد که در میدان ویژه‌ی خود در قطعه‌ی UDP قرار می‌گیرد. در سمت گیرنده همگی 4 کلمه‌ی 16 بیت موجود در قطعه‌ی UDP به همراه کد مجموع مقابله با یکدیگر جمع شده و چنانچه خطایی در قطعه ایجاد نشده باشد آنگاه حاصل جمع بدست آمده برابر عبارت ذیل خواهد شد:

1111111111111111

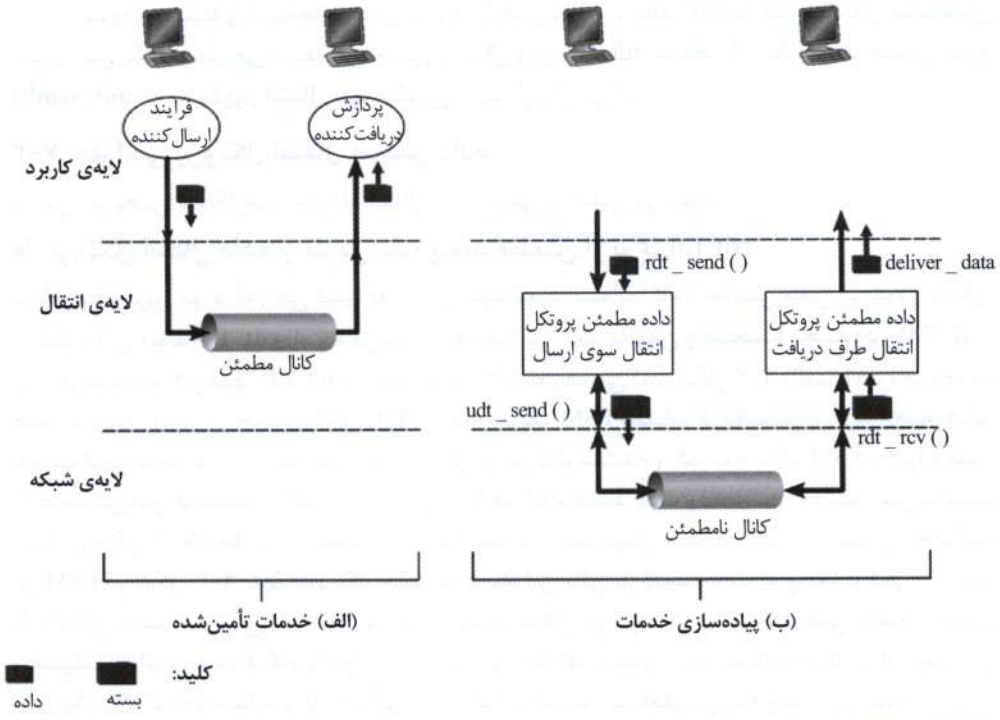
چنانچه یکی از بیت‌ها برابر (0) شود مفهوم آن وقوع خطا در بسته‌ی UDP است. ممکن است برای شما این پرسش مطرح شود چرا علیرغم آنکه اکثر پروتکل‌های لایه‌ی پیوند داده مانند بسیاری از پروتکل‌های **اترنت** کُد مجموع مقابله را فراهم می‌آورند باز هم پروتکل UDPاین کد را در بسته‌های خود تعریف می‌کند؟ پاسخ این پرسش آن است که همه‌ی پیوندهای بین مبدا و مقصد که از پروتکل لایه پیوند استفاده می‌کنند ضمانت اجرای عملیات تشخیص خطا را در خود ندارند. ممکن است یکی از این پیوندها از پروتکلی استفاده نماید که این امکانات را پشتیبانی نمی‌کند. علاوه بر آن اگر قطعه‌های UDP در سراسر یک پیوند بدون خطا انتقال یابند این احتمال وجود دارد که خطای انجام شده بر روی بیت‌ها در هنگام ذخیره‌ی بسته‌های UDP در حافظه‌ی یک مسیریاب اتفاق افتد. بنابراین با توجه به این مهم که ضمانت آشکارسازی خطا بین پیوندها و یا در حافظه وجود ندارد، پروتکل UDP ناگزیر است روش‌هایی را برای آشکارسازی خطا در لایه‌ی انتقال تعریف نماید. این روش‌ها بایستی بر اساس انتها به انتها باشد. به عبارت دیگر انتقال داده به گونه‌ی انتها به انتها بایستی همراه خود روش‌های آشکارسازی خطا را نیز شامل شود. این خدمات را در طراحی برنامه‌های کاربردی اصطلاحاً **اصل انتها به انتها** می‌نامند. مفهوم این اصل آن است که برخی روش‌های ویژه مانند آشکارسازی خطا بایستی از نقطه‌نظر عملکردی در لایه‌های پائین تعریف گردد.

بنابراین از آنجائیکه انتظار می‌رود پروتکل IP در اطراف هر پروتکل لایه‌ی دوم عمل نماید، بدیهی است لایه‌ی انتقال روش‌های آشکارسازی خطا را در درون خود جای داده است اما قادر به تصحیح و بازیابی موارد به خطا رفته نمی‌باشد. برخی از پیاده‌سازی‌های UDP از قطعه‌هایی که دچار خطا شده‌اند به سادگی صرف‌نظر می‌کنند و برخی دیگر آن‌ها را به همراه یک پیام هشدار به برنامه‌ی کاربردی ارسال کننده، برگشت می‌دهند. در ادامه‌ی مطلب، TCP و اصول آن به‌عنوان یک پروتکل مطمئن انتقال داده بررسی خواهد شد اما قبل از آن اصول مربوط به مفهوم انتقال مطمئن داده‌ها تحلیل می‌شود.

۳-۶ اصول انتقال مطمئن داده

در این بخش اصول مربوط به انتقال داده و نیز چالش‌های پیش روی آن بررسی می‌شود. چالش مربوط به پیاده‌سازی انتقال مطمئن داده‌ها تنها متعلق به لایه انتقال نمی‌باشد بلکه در دو لایه‌ی پیوند و کاربرد نیز وجود دارد. اگر قرار باشد فهرستی از عنوان‌های مهم و بنیادی 10 چالش عمده‌ی شبکه تهیه گردد. چالش مربوط به انتقال مطمئن داده‌ها قطعاً در ابتدای آن فهرست قرار خواهد گرفت. در بخش بعدی که TCP مورد ارزیابی قرار می‌گیرد بسیاری از اصولی که در این بخش بررسی شده است از نقطه‌نظر کاربرد در TCP نشان داده می‌شوند. شکل ۳-۸ چارچوب مطالعه پیرامون انتقال مطمئن داده

را نشان می‌دهد. به اختصار خدمات فراهم آوری شده برای موجودیت‌های لایه‌های بالاتر عبارت است از یک پیوند مطمئن جهت عبور و انتقال داده‌ها. با در اختیار داشتن یک کانال مطمئن، احتمال تغییر بیت‌های اطلاعاتی از ارزش 0 به 1 و یا برعکس آن که نوعی وقوع خطا به‌شمار می‌آید و نیز مفقودی آن‌ها رخ نمی‌دهد و همه‌ی بیت‌های تشکیل دهنده‌ی داده به ترتیب وقوع خویش از یک نقطه به نقطه‌ی دیگر منتقل می‌شوند. این الگوی خدماتی است که TCP به کاربردهای اینترنت که آن را احضار می‌نمایند، ارائه می‌دهد. این مهم را می‌توان وظیفه‌ی یک **پروتکل انتقال مطمئن داده** جهت پیاده‌سازی این خدمات دانست. این وظیفه هنگامی دچار چالش فزاینده‌تری می‌شود که لایه‌ی زیرین آن حالت غیر مطمئن را اختیار نماید. به‌عنوان مثال TCP یک پروتکل انتقال مطمئن داده است که بر روی پروتکل IP که دارای وضعیت غیرمطمئن است و در دو سوی لایه‌ی شبکه قرار گرفته است، پیاده‌سازی می‌شود.



شکل ۳-۸ ♦ انتقال مطمئن داده: الگوی خدمات و پیاده‌سازی آن

از نقطه‌نظر کلی لایه‌ی واقع در ذیل دو نقطه‌ی ارتباطی مطمئن، ممکن است شامل یک پیوند لایه‌ی فیزیکی مانند پروتکل پیوند داده در یک شبکه‌ی معمولی باشد و یا بین شبکه‌های جهانی یک پروتکل انتقال قرار گیرد. در این بخش این چالش در قالب کلی به گونه‌ی یک کانال انتقال نقطه به نقطه‌ی غیر مطمئن ارزیابی می‌شود. شکل ۳-۸ (ب) واسط‌های مورد نیاز پروتکل انتقال داده را نشان می‌دهد. در طرف ارسال این پروتکل از بالا از سوی یک تابع فراخوان () rdt_send احضار می‌شود.

داده‌های مورد نظر جهت انتقال به لایه‌های بالاتر سمت گیرنده تحویل می‌شوند. بدیهی است عبارت rdt به مفهوم پروتکل **انتقال مطمئن داده**[1] است و عبارت send_ به مفهوم آن است که سمت ارسال rdt فراخوانی شده است. در سمت دریافت تابع **rdt_rcv()** هنگامی فراخوانی می‌شود که یک بسته در حال دریافت بر روی پیوند است. هنگامی‌که پروتکل rdt قصد تحویل داده را به لایه‌ی بالاتر دارد این عمل از طریق فراخوانی تابع deliver_data() انجام می‌گیرد. لازم به یادآوری است که در ادامه‌ی مطالب این بخش از واژه‌ی بسته به‌عنوان جایگزین قطعه برای لایه‌ی انتقال استفاده شده است زیرا که نظریه‌ی اعمال شده در این بخش جهت اعمال به شبکه‌های کامپیوتری است و فقط شامل شبکه‌ی اینترنت نخواهد شد و بنابراین استفاده از واژه‌ی بسته مناسب‌تر است.

در این بخش تنها انتقال **داده‌ی یک‌طرفه** که عبارت است از ارسال داده از سوی یک فرستنده به یک گیرنده بررسی می‌شود. بدیهی است انتقال **مطمئن داده‌ی دوطرفه** دشوارتر از انتقال یک‌طرفه نمی‌باشد اما از نقطه‌نظر تشریح عوامل آن پیچیده‌تر است. اگرچه انتقال داده به گونه‌ی یک‌طرفه بررسی می‌شود اما در عمل دوطرف ارسال و دریافت پروتکل بنابر شکل ۳-۸ بایستی به‌صورت دوطرفه عمل نمایند. علاوه بر مبادله‌ی بسته‌های حامل داده دوطرف ارسال و دریافت rdt نیازمند مبادله‌ی بسته‌های کنترل بین یکدیگر نیز می‌باشند. هر دو سوی ارسال و دریافت rdt بسته‌ها را از طریق فراخوانی تابع **udt_send()** که به مفهوم **انتقال غیرمطمئن**[2] است ارسال می‌کنند.

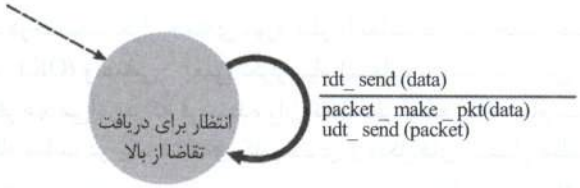
۳-۷ طراحی پروتکل انتقال مطمئن داده

در این زیر بخش پروتکل‌های مختلف انتقال داده معرفی و تحلیل می‌شوند.

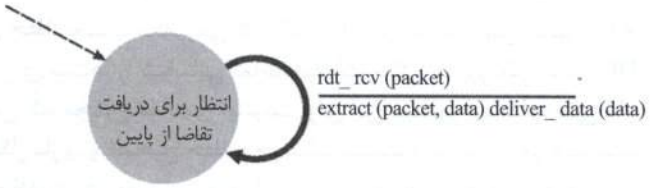
• پروتکل انتقال داده از طریق یک پیوند مطمئن از نوع: rdt 1.0

ابتدا از ساده‌ترین مورد آغاز می‌کنیم که در آن پیوند مورد استفاده کاملاً مطمئن تلقی می‌شود. پروتکل مرتبط با این پیوند **rdt 1.0** نامیده می‌شود و به طور کلی یک **ماشین وضعیت محدود**[3] (FSM) را برای فرستنده و گیرنده‌ی rdt 1.0 در قالب شکل ۳-۹ تعریف می‌کند. شکل ۳-۹ (الف) FSM مربوط به عملیات ارسال است در حالیکه شکل ۳-۹ (ب) به تعریف FSM عملیات دریافت می‌پردازد. آنچه حائز اهمیت است توجه به این نکته است که برای هر دو مورد فرستنده و گیرنده یک FSM مجزا وجود دارد. ماشین‌های فرستنده و گیرنده در شکل ۳-۹ هر کدام فقط یک وضعیت را اختیار می‌نمایند. انتقال پروتکل از یک حالت به وضعیت دیگر با استفاده از نماد پیکان مشخص شده است. از آنجائیکه هر FSM در شکل ۳-۹ تنها فقط یک حالت دارد بنابراین برای بازگشت حالت به وضعیت قبلی آن نیز یک انتقال بیشتر نیاز نمی‌باشد. رخدادهایی که سبب انتقال می‌شوند در بالای خط افقی به‌عنوان انتقال و عملیات انجام شده به هنگام وقوع رویدادها در زیر خط افقی نشان داده شده‌اند. هنگامی‌که عملیاتی برای یک رخداد وجود ندارد و یا رخدادی انجام نگرفته است اما عملیاتی به وقوع پیوسته است از علامت ∧ به ترتیب در ذیل یا بالای خط افقی استفاده شده است تا معرّف عدم استفاده از یک عملیات و یا رخداد باشد. وضعیت آغازین ماشین FSM از طریق یک پیکان خط‌چین مشخص می‌شود. اگرچه FSM شکل ۳-۹تنها فقط یک حالت دارد اما ماشین‌های FSM در مراحل بعدی دارای حالت‌های بیشتری هستند. بنابراین آگاهی از وضعیت اولیه هر FSM دارای اهمیت قابل توجهی می‌باشد.

1- Reliable data transfer
2- Unreliable data tranfer
3- Finite state machine



(الف) پروتکل rdt 1.0 سمت ارسال



(ب) پروتکل rdt 1.0 سمت دریافت

شکل ۳-۹ ♦ یک پروتکل کاملاً مطمئن rdt 1.0 برای یک کانال

در سمت ارسال، rdt داده‌ها را از لایه‌ی بالاتر با استفاده از رخداد rdt_send (data) پذیرش می‌نماید و سپس به تعریف یک بسته شامل داده‌ها از طریق انجام عملیات make_pkt (data) می‌پردازد و آن را به درون پیوند گسیل می‌نماید. در عمل رخداد rdt_send (data) از طریق یک تابع فراخوان مانند rdt_send () به‌وسیله کاربرد لایه بالاتر نتیجه می‌شود. در سمت دریافت rdt یک بسته را از طریق پیوندی با استفاده از رخداد rdt_rcv (packet) تحویل می‌گیرد و سپس داده‌ی موجود در بسته را از طریق تابع عملیاتی extract (packet,data) خارج می‌نماید و آن را از طریق تابع عملیات deliver_data (data) به لایه‌ی بالاتر منتقل می‌کند. در عمل رخداد rdt_rcv (packet) از تابع فراخوانی مانند rdt_rcv () نتیجه می‌شود که از پروتکل لایه‌ی زیرین نتیجه شده است. در این پروتکل ساده تفاوت چشمگیری بین یک جزء داده و یک بسته وجود ندارد. اگرچه همه‌ی جریان بسته از فرستنده به گیرنده و از طریق یک کانال مطمئن انتقال می‌باشد. در این حالت گیرنده نیازمند فراهم آوری هیچگونه بازخوردی به فرستنده نمی‌باشد زیرا احتمال اشکال در انتقال وجود ندارد. بدیهی است فرض در نظر گرفته شده بر آن است که گیرنده با همان سرعت فرستنده داده‌ها را دریافت می‌کند و بنابراین گیرنده تقاضای کاهش سرعت از سوی فرستنده را نخواهد نمود.

پروتکل انتقال داده از طریق پیوند با بیت‌های خطا از نوع rdt 2.0

الگوی واقعی در عمل ممکن است سبب تخریب برخی از بیت‌های داده در یک بسته گردد. چنین خطا در بیت‌های بسته به‌وسیله عناصر فیزیکی شبکه به هنگام انتقال، انتشار یا ذخیره‌ی بسته رخ می‌دهد. در این زیر بخش مانند گذشته فرض بر آن است که همه‌ی بسته‌های انتقال یافته با همان ترتیب ارسال، دریافت خواهند شد اگرچه برخی از بیت‌های آن‌ها ممکن است دچار خطا شده باشد. قبل از تکمیل یک پروتکل جهت انتقال مطمئن اطلاعات بر روی یک کانال، نخست به بررسی چگونگی برخورد با چنین وضعیتی می‌پردازیم. حالتی را تجسم نمایید که چگونه یک پیغام طولی را از طریق خط تلفن ممکن است پیمایش نماید. در یک الگوی ویژه ممکن است دریافت کننده‌ی پیام پس از دریافت هر جمله از پیام با بیان واژه‌ای مانند **ok** این نکته را اعلام نماید که جمله‌ی ارسالی شنیده، فهمیده و ذخیره گردید. چنانچه دریافت کننده جمله‌ای را به‌صورت نامفهوم یا بهم ریخته دریافت کند

ممکن است فرستنده درخواست تکرار جمله‌ی مورد نظر را نماید. در این حالت چنین پروتکلی از هر دو وضعیت **تائید مثبت**[1] (OK) و **منفی**[2] یعنی تکرار پیام استفاده نموده است. این پیام‌های کنترل برای گیرنده این امکان را فراهم می‌آورند که فرستنده را از نقطه‌نظر اینکه کدام پیام سالم و کدام غیر سالم دریافت شده است آگاه نماید. در یک شبکه‌ی کامپیوتری پروتکل‌های انتقال مطمئن داده بر مبنای چنین تدبیری پایه‌ریزی می‌شوند. روش تکرار مجدد پیام دچار اشکال را اصطلاحاً **پروتکل تکرار مجدد پیام بدون خطا**[3] (ARQ) می‌نامند. معمولاً در پروتکل‌هایی نظیر ARQ سه نوع توانمندی دیگر جهت آشکارسازی بیت‌هایی که دچار خطا شده‌اند نیز بایستی تعریف گردد. این سه عبارتند از:

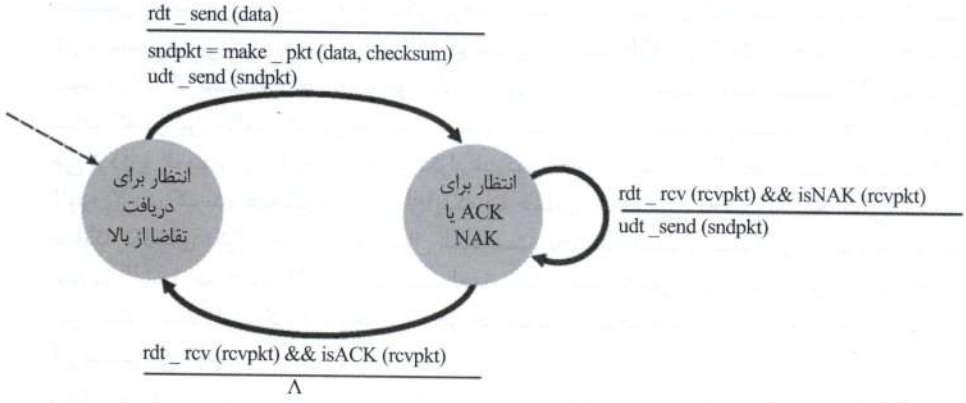
۱- **آشکارسازی خطا:** نخست مکانیزمی لازم است تا برای گیرنده این امکان را فراهم آورد که وقوع خطا بر روی بیت‌ها را شناسایی نماید. قبلاً اشاره گردید پروتکلی مانند UDP با استفاده از اطلاعات میدان کُد مجموع مقابله‌ی اینترنت برای این منظور استفاده می‌نماید. در فصل ۵ روش‌های آشکارسازی و تصحیح خطا با جزئیات بیشتری بررسی خواهد شد. این روش‌ها به بیت‌های اضافه‌تری علاوه بر بیت‌های اصلی در بسته‌های داده جهت انتقال از سوی فرستنده به گیرنده نیازمند می‌باشند. این بیت‌های اضافی در محدوده‌ی کُد مجموع مقابله‌ی بسته یعنی rdt 2.0 جمع‌آوری و نگهداری می‌شوند.

۲- **بازخورد گیرنده**[4]: از آنجائیکه فرستنده و گیرنده ممکن است بر روی سیستم‌های انتهایی مختلف اجرا گردند و این سیستم‌ها هزاران کیلومتر از یکدیگر فاصله داشته باشند. تنها روشی که فرستنده می‌تواند از چگونگی وضعیت دریافت بسته‌های خود به‌وسیله گیرنده آگاهی یابد آن است که گیرنده بازخورد ویژه‌ای را برای وی فراهم آورد. پاسخ‌های از قبیل مثبت **(ACK)**[5] و یا منفی **(NAK)**[6] می‌تواند نمونه‌ای از این بازخوردها باشد. در پروتکل rdt 2.0 ارسال ACK و NAK از سوی گیرنده به فرستنده به گونه‌ی مشابه می‌تواند روی دهد. این بسته‌ها می‌توانند تنها فقط طولی معادل یک بیت را داشته باشند. به‌عنوان مثال یک ارزش 0 به مفهوم NAK و ارزش 1 به مفهوم ACK باشد.

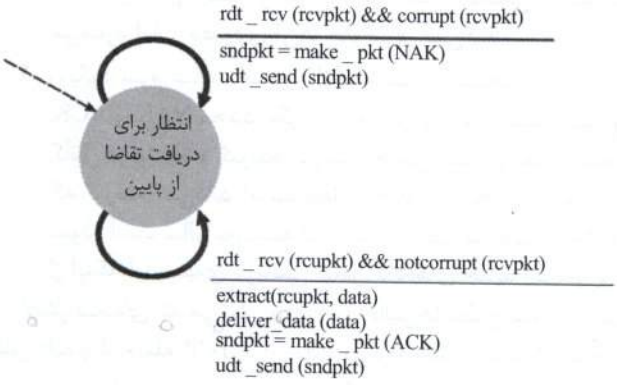
۳- **تکرار مجدد:** یک بسته که در آن خطایی به وقوع پیوسته است مجدداً از سوی فرستنده به گیرنده گسیل می‌گردد. شکل ۳-۱۰ بیان کننده FSM پروتکل rdt 2.0 جهت انتقال داده با روش آشکارسازی خطا و با هر دو تائید مثبت و منفی آن می‌باشد. در سمت ارسال rdt 2.0 دارای دو وضعیت است. در وضعیت سمت چپ پروتکل ارسال در انتظار دریافت داده‌ها از لایه‌ی بالاتر است. هنگامی‌که که یک رخداد rdt_send (data) روی می‌دهد فرستنده به خلق یک بسته‌ی sendpkt می‌پردازد که این بسته نه تنها شامل داده‌ی مورد نظر جهت ارسال می‌باشد بلکه به همراه خود کُد مجموع مقابله را نیز مشابه آنچه در بخش ۳-۵ توضیح داده شد حمل می‌کند. این پروتکل بسته‌ی ایجاد شده را از طریق عملیات udt_send (sendpkt) گسیل می‌کند. در وضعیت سمت راست FSM پروتکل ارسال در انتظار دریافت یک بسته‌ی ACK یا NAK از گیرنده می‌باشد. اگر یک بسته‌ی ACK دریافت گردد آنگاه در شکل ۳-۱۰ عبارت

1- Positive acknowledgments
2- Negative acknowledgments
3- Automatic Repeat Request
4- Receiver Feedback
5- Acknowledgment
6- Not acknowledgment

rdt_rcv (rcvpkt) && iSAK (rcvpkt) در ارتباط با این رخداد قلمداد می‌گردد و در نتیجه فرستنده آگاه می‌شود که بسته‌ای که اخیراً ارسال نموده است بدون اشکال از سوی گیرنده دریافت شده است و بنابراین پروتکل به وضعیت انتظار جهت دریافت داده از لایه‌ی بالاتر می‌رود. اگر یک پیام NAK دریافت گردد آنگاه پروتکل نسبت به انتقال مجدد بسته‌ی مفقود شده اقدام می‌کند و در انتظار بسته‌ی ACK یا NAK از سوی گیرنده قرار می‌گیرد. توجه به این نکته ضروری است هنگامی‌که فرستنده‌ی در انتظار پیام تائید ACK و یا NAK می‌باشد قادر به دریافت داده‌های بیشتری از سوی لایه بالاتر نمی‌باشد. به عبارت دیگر رخداد rdt_send ( ) نمی‌تواند روی دهد. این رخداد هنگامی به وقوع می‌پیوندد که فرستند پیام ACK را دریافت و از این وضعیت خارج گردد. فرستنده بسته‌های جدیدی را تا زمانیکه اطمینان از دریافت صحیح آن‌ها از سوی گیرنده نداشته باشد، ارسال نخواهد نمود. به همین دلیل پروتکل‌هایی مانند rdt 2.0 اصطلاحاً **توقف و انتظار**[1] نامیده می‌شوند.



(الف) طرف ارسال: rdt 2.0



(ب) طرف دریافت: rdt 2.0

شکل ۳-۱۰ ♦ پروتکل rdt 2.0 برای یک کانال با بیت‌های خطا

ماشین FSM سمت گیرنده برای rdt 2.0 هنوز هم حالت یک وضعیتی را به دنبال دارد. به هنگام

1- Stop and Waite

ورود بسته پاسخ گیرنده در قالب ACK و یا NAK می‌باشد و بستگی به وضعیت سلامت بسته‌ی دریافتی دارد.

در شکل ۳-۱۰ پیام rdt_rcv (rcvpkt) && corrupt (rcvpkt) در ارتباط با رخدادی است که بسته‌ی دریافت شده دچار خطا می‌باشد. پروتکل rdt 2.0 دارای یک ضعف عمده از لحاظ عملکرد است و آن هنگامی است که پیام‌های ACK و یا NAK خود دستخوش خطا گردند. جهت آشکارسازی وقوع خطا بر روی بسته‌های NAK و یا ACK بایستی بیت‌های مربوط به کُد مجموع مقابله (checksum) را به آن‌ها اضافه نمود. چالش این موضوع مربوط به نوع عملیاتی است که یک پروتکل بایستی جهت بازیابی این بسته‌ها (یعنی ACK و NAK) از خطاهای احتمالی انجام دهد. دشواری قضیه از آن جهت است که اگر یک بسته‌ی ACK یا NAK آلوده گردد، فرستنده به هیچ روی نمی‌تواند از دریافت صحیح آخرین داده‌های ارسالی به‌وسیله‌ی گیرنده آگاه شود. سه راهکار را می‌توان در ذیل برای بازیابی بسته‌های آلوده ACK وNAK در نظر گرفت:

۱- نخستین راهکار مربوط است به چگونگی برخورد انسان‌ها در مورادی که انتقال پیام بین آن‌ها دچار دشواری می‌شود. به‌عنوان مثال اگر گوینده پیام تائید **OK** و یا "تکرار مجدد" از جانب شونده را درک نکند این احتمال وجود دارد که وی با عبارت "منظور شما چیست؟" پرسش نماید که در این حالت یک نوع جدید از بسته‌ی ارسالی و دریافت به‌وسیله پروتکل معرفی می‌گردد و گوینده نیز پاسخ را مجدداً تکرار می‌کند. حال اگر عبارت "منظور شما چیست؟" آلوده گردد، گیرنده هیچگونه ایده‌ای از اینکه آیا جمله‌ی آلوده شده بخشی از پیام اصلی است و یا آنکه تقاضای تکرار آخرین پیام ارسال شده است، نخواهد داشت و بنابراین ممکن است با عبارت "شما چه گفتید؟" از خود عکس‌العمل نشان دهد. بدیهی است پاسخ این پرسش آخر نیز ممکن است آلوده گردد. نتیجه آنکه هر چه مسیر به طرف جلوتر حرکت می‌کند دشواری آن بیشتر می‌شود.

۲- راهکار جایگزین دوم آن است که به اندازه‌ی کافی بیت‌های کُد مجموع مقابله را به بسته‌های ACK و NAK اضافه نمود که اجازه دهد فرستنده نه تنها به آشکارسازی خطا بلکه به بازیابی بیت‌های دچار خطا نیز بپردازد. این مشکل کانالی را که بسته‌ها را آلوده می‌کند برطرف می‌سازد اما از مفقودی آن‌ها جلوگیری نمی‌کند.

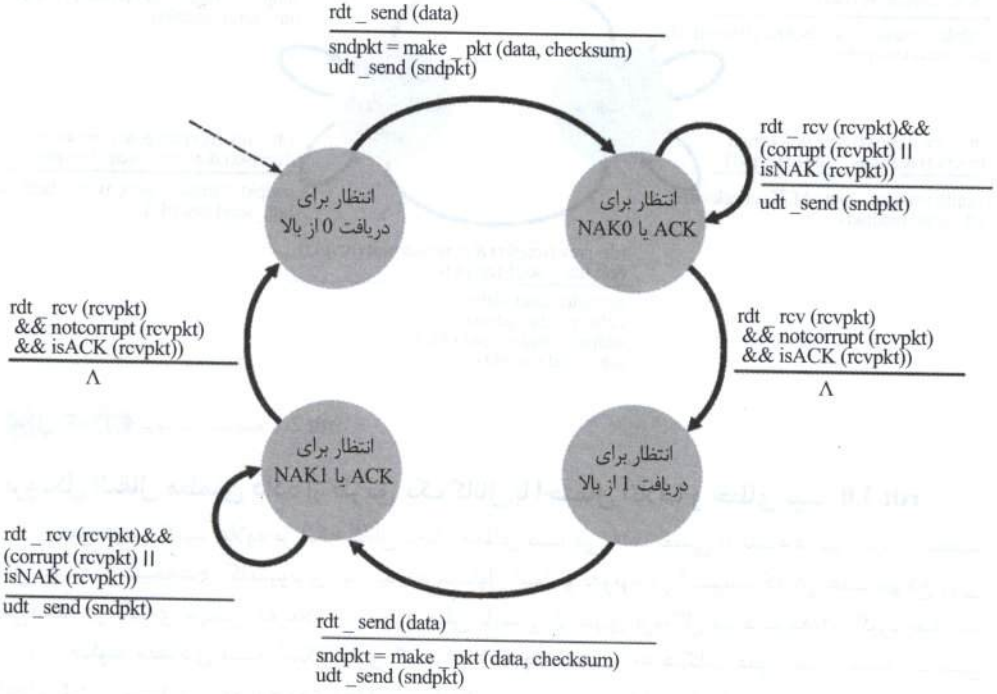
۳- رویکرد سوم عبارت است از آنکه فرستنده بسته‌های جاری را به هنگام دریافت پیام‌های آلوده‌ی ACK و NAK مجدداً تکرار نماید. این رویکرد سبب تکثیر و یا دوباره‌سازی بسته‌ها در درون کانال فرستنده و گیرنده می‌شود. چالش بنیادی تکرار بسته‌ها در کانال برای گیرنده آن است که گیرنده نمی‌داند آیا بسته‌های ACK و یا NAK ارسالی از سوی وی که آخرین مرتبه ارسال نموده است سالم به‌وسیله فرستنده دریافت شده‌اند یا نه؟ به عبارت دیگر هیچگونه آگاهی قبلی از اینکه آیا بسته‌ی دریافتی شامل داده‌های جدید است و یا انکه تکراری می‌باشند، ندارد.

راهکار ساده‌ای که می‌توان برای این چالش‌ها مطرح نمود و بدیهی است تقریباً همه‌ی پروتکل‌های انتقال داده و از جمله TCP از آن بهره می‌برند عبارت است از اضافه نمودن یک محدوده‌ی جدید به بسته‌ی داده و اجبار فرستنده به شماره‌گذاری بسته‌های داده‌ی ارسالی و قرار دادن این **شماره ترتیب**[1] در محدوده‌ی جدید تعریف شده. در این حالت گیرنده فقط لازم است شماره ترتیب یک بسته را برای آگاهی از اینکه آیا بسته تکراری است و یا نه ارزیابی نماید. برای این مورد ساده‌ی پروتکل توقف و

1- Sequence number

انتظار تنها یک ارزش یک بیت شماره‌ی ترتیب کافی می‌باشد، زیرا از طریق آن گیرنده می‌تواند ملاحظه نماید آیا فرستنده بسته‌ی قبلی را مجدداً ارسال نموده است و یا آنکه در حال ارسال یک بسته‌ی جدید می‌باشد. در حالت نخست یعنی تکرار بسته، شماره ترتیب بسته‌ی دریافتی همانند شماره ترتیب بسته‌ی دریافت شده‌ی اخیر می‌باشد و در حالت دوم این شماره تغییر یافته و چگونگی تغییر آن نیز به جلو و به‌صورت تقسیم پیمانه‌ی 2 (modulo-2) می‌باشد. از آنجائیکه در حال حاضر کانال انتقال مورد نظر به گونه‌ای است که بسته‌ها را از دست نمی‌دهد بنابراین بسته‌های ACK و NAK نیازی به بیان شماره ترتیب بسته‌ای که در حال تائید آن می‌باشند، ندارند. فرستنده می‌داند که بسته‌ی ACK و NAK دریافتی چه سالم و یا آلوده به واسطه‌ی بسته‌ای که اخیراً انتقال یافته، تولید شده است.

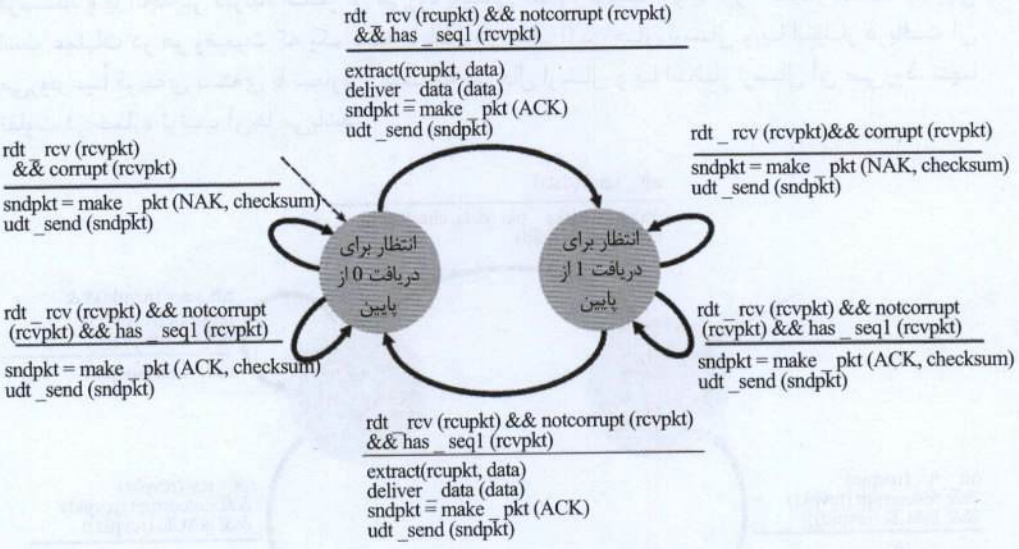
شکل‌های ۳-۱۱ و ۳-۱۲ ماشین‌های FSM را برای پروتکل rdt 2.1 و نوع تغییر یافته‌ی rdt 2.0 نشان می‌دهند. ماشین‌های FSM فرستنده و گیرنده‌ی پروتکل rdt 2.1 هر یک از لحاظ تعداد وضعیت‌ها دو برابر حالت‌های قبلی می‌باشند. این موضوع به دلیل آن است که وضعیت هر پروتکل اکنون بایستی انعکاس دهنده‌ی این موضوع باشد که آیا بسته‌ی جاری در حال ارسال به‌وسیله‌ی فرستنده و یا آنچه در گیرنده انتظار آن می‌رود بایستی شماره ترتیب 0 و یا 1 را اختیار نماید. بدیهی است عملیات در هر وضعیت که یک بسته با شماره ردیف 0 در حال ارسال و یا انتظار دریافت آن می‌رود عیناً قرینه‌ی بسته‌ی با شماره 1 است که در حال ارسال و یا انتظار ارسال آن می‌رود. تنها تفاوت در شماره ترتیب آن‌ها می‌باشد.



شکل ۳-۱۱ ♦ پروتکل فرستنده rdt 2.1

پروتکل rdt 2.1 از هر دو تائید مثبت و منفی از جانب فرستنده و گیرنده بهره می‌جوید. هنگامی‌که

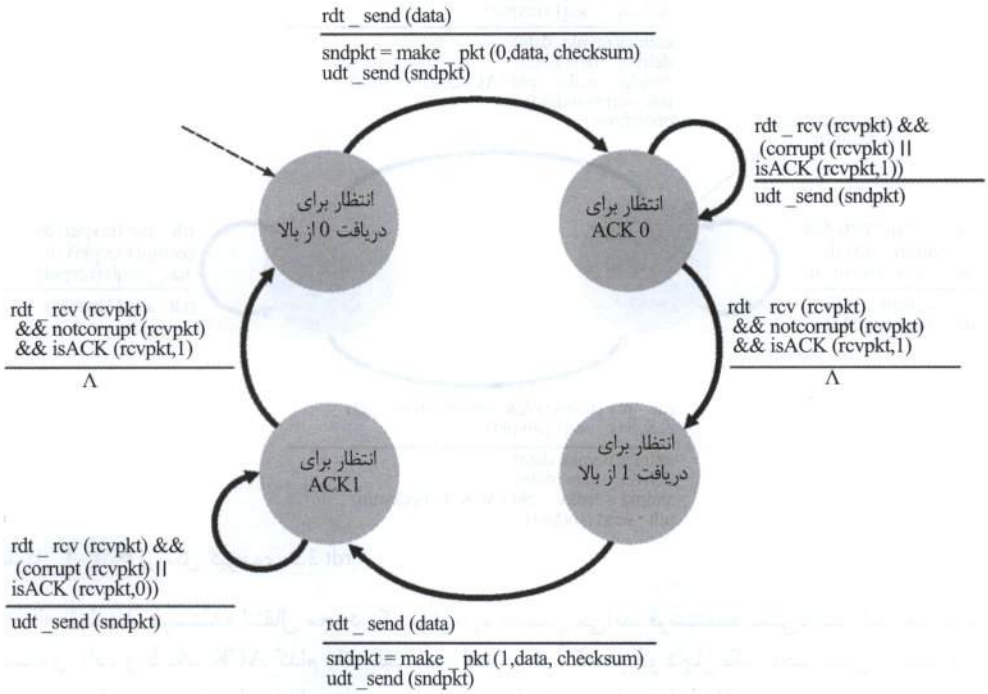
بسته‌ای خارج از نوبت دریافت می‌شود. گیرنده تائید مثبت را برای آن بسته‌ی دریافتی ارسال می‌کند. هنگامی‌که بسته‌ای آلوده دریافت می‌شود گیرنده یک تائید منفی را گسیل می‌نماید. برای آخرین بسته ی سالم دریافتی ارسال ACK و NAK تاثیر NAK را دارد. فرستنده‌ای که دو ACK را برای بسته‌ی مشابهی دریافت نماید می‌داند که گیرنده، بسته را سالم دریافت ننموده است زیرا در پی بسته دو مرتبه بسته‌ی ACK دریافت شده است. پروتکل rdt 2.2 یک پروتکل انتقال مطمئن داده‌ی آزاد NAK قلمداد می‌شود و در ارتباط با یک کانال با خطای بیت است که به‌وسیله‌ی شکل ۳-۱۳ و ۳-۱۴ نشان داده شده است. تفاوت ظریف بین دو پروتکل rdt 2.1 و rdt 2.2 آن است که گیرنده بایستی شماره ترتیب بسته‌ای را که به‌وسیله یک پیام ACK تائید می‌شود پیوست نماید. این مهم از طریق افزودن ACK0 یا ACK1 به‌عنوان آرگومان تابع ( ) make_pkt در FSM گیرنده انجام می‌شود. فرستنده نیز بایستی شماره ترتیب بسته ای را که در حال تائید از سوی یک پیام ACK دریافتی است، ارزیابی نماید. این ارزیابی از طریق یکی از دو آرگومان 0 و یا 1 در تابع iSACK در ماشین FSM فرستنده صورت می‌پذیرد.



شکل ۳-۱۲ ♦ پروتکل گیرنده rdt 2.1

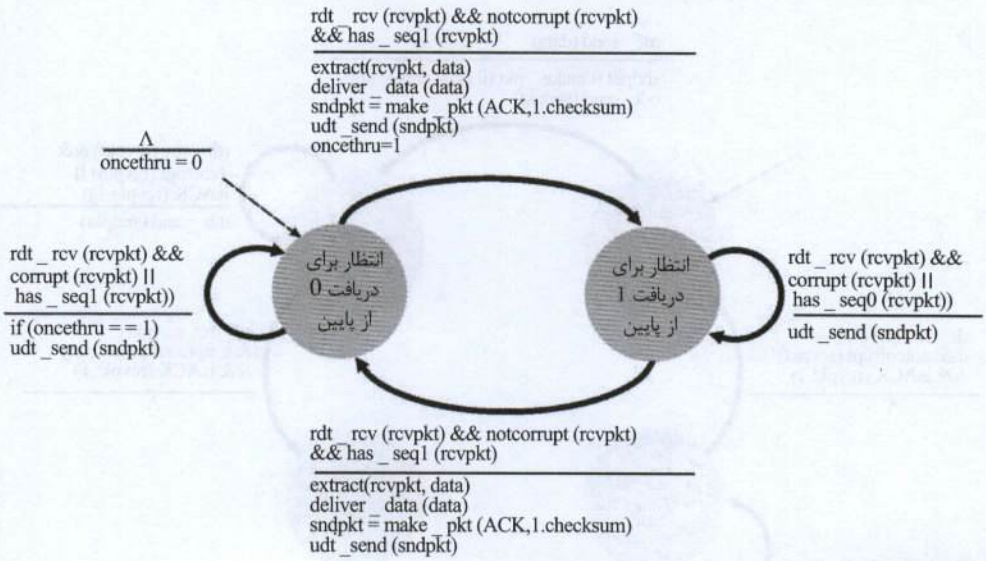
پروتکل انتقال مطمئن داده از طریق یک کانال با احتمال اتلاف و خطای بیت rdt 3.0

اکنون فرض کنید علاوه بر آنکه کانال دچار خطای بیت می‌باشد بعضی از بسته‌ها نیز در آن مفقود می‌شوند. در شبکه‌های کامپیوتری این پدیده متداول است و به‌ویژه در اینترنت که در اغلب مواقع روی می‌دهد. دو نکته‌ی مهمی که علاوه بر موارد قبلی بایستی از سوی پروتکل مورد توجه قرارگیرد عبارتند از: (۱) چگونه مفقودی بسته آشکارسازی گردد؟ و (۲) چه تدبیری به هنگام مفقود شدن بسته بایستی اتخاذ گردد. استفاده از کُد مجموع مقابله، شماره‌های ترتیب، بسته‌های ACK و انتقال مجدد اصولی هستند که در پروتکل rdt 2.2 بررسی شدند و به پاسخگویی مورد دوم یعنی برخورد با مفقودی بسته می‌پردازند. برای آشکارسازی مفقودی بسته مکانیزم‌های جدیدتری بایستی به پروتکل اضافه گردد.



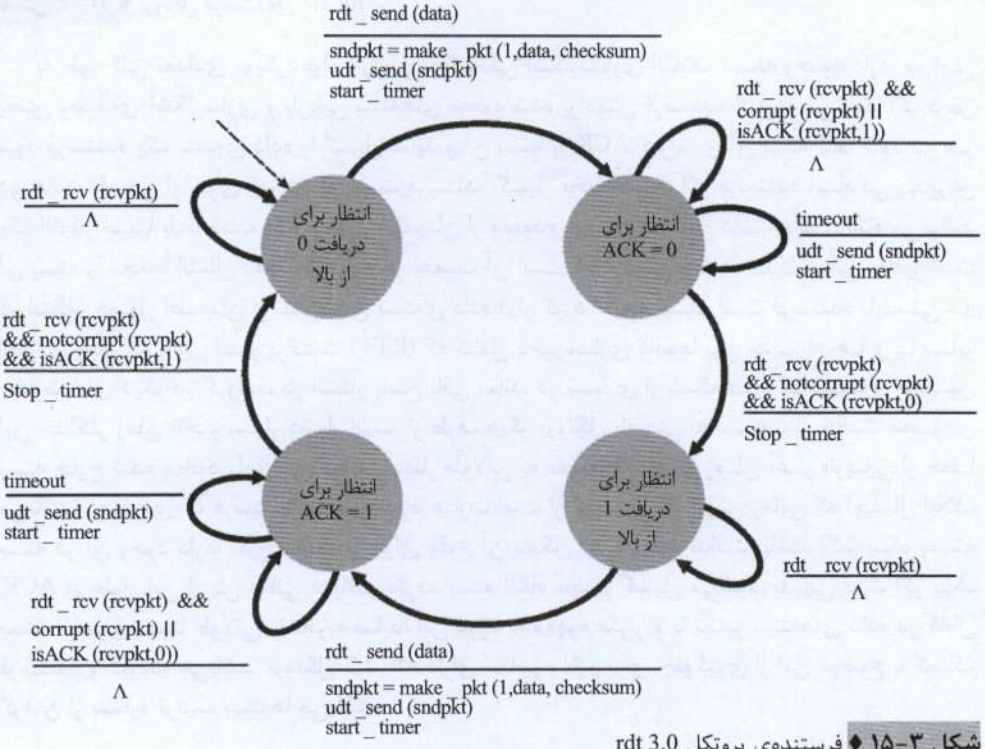
شکل ۳-۱۳ ♦ پروتکل فرستنده‌ی rdt 2.2

به طور کلی تعدادی رویکرد برای ارزیابی چگونگی آشکارسازی اتلاف بسته وجود دارد. در این بخش وظیفه‌ی آشکارسازی و بازیابی بسته‌های مفقود شده بر دوش فرستنده گذارده می‌شود. اگر فرض شود فرستنده یک بسته‌ی داده را گسیل نماید و آن بسته یا ACK گیرنده‌ی آن بسته تلف شود در هر دو حالت پاسخی از سوی گیرنده به سمت فرستنده گسیل نخواهد شد. اگر فرستنده آمادگی پذیرش یک انتظار نسبتاً بلند مدت را تا حصول اطمینان از مفقودی بسته‌ی ارسالی داشته باشد آنگاه می‌تواند آن بسته را مجدداً انتقال دهد. نکته‌ی حائز اهمیت آن است که فرستنده تا چه مدت زمانی حاضر است در انتظار حصول اطمینان از تلف شدن بسته‌ی داده قرار گیرد؟ آنچه مسلم است فرستنده بایستی به اندازه‌ی زمان دوره‌ی رفت و برگشت (RTT) که شامل ذخیره‌سازی داده‌ها بین مسیریاب‌ها و یا سایر پردازش‌های دیگر در گیرنده، در انتظار پاسخ باقی بماند. در بسیاری از شبکه‌های کامپیوتری تخمین این حداکثر زمان تاخیر بسیار دشوار است. از طرف دیگر پروتکل بایستی به سرعت از حالت مفقودی بسته خارج شده و خود را بازیابی نماید. انتظار طولانی به مفهوم فاصله تا زمان آغاز بازیابی از خطا می‌باشد. رویکردی که فرستنده در عمل دارد عبارت است از گزینش یک ارزش زمانی که احتمال اتلاف بسته در آن وجود دارد. اگرچه ضمانتی برای وقوع آن ممکن است وجود نداشته باشد. اگر یک بسته ACK در طول این ارزش زمانی دریافت نگردد بسته آنگاه مجدداً گسیل می‌شود. بدیهی است اگر یک بسته تاخیری نسبتاً طولانی را تجربه نماید، این مورد به مفهوم تکرار و یا تکثیر بسته‌های داده در کانال فرستنده و گیرنده می‌باشد. پروتکل rdt 2.2 دارای مکانیزم لازم برای جلوگیری از این موضوع با کمک گرفتن از شماره ترتیب بسته‌ها می‌باشد.

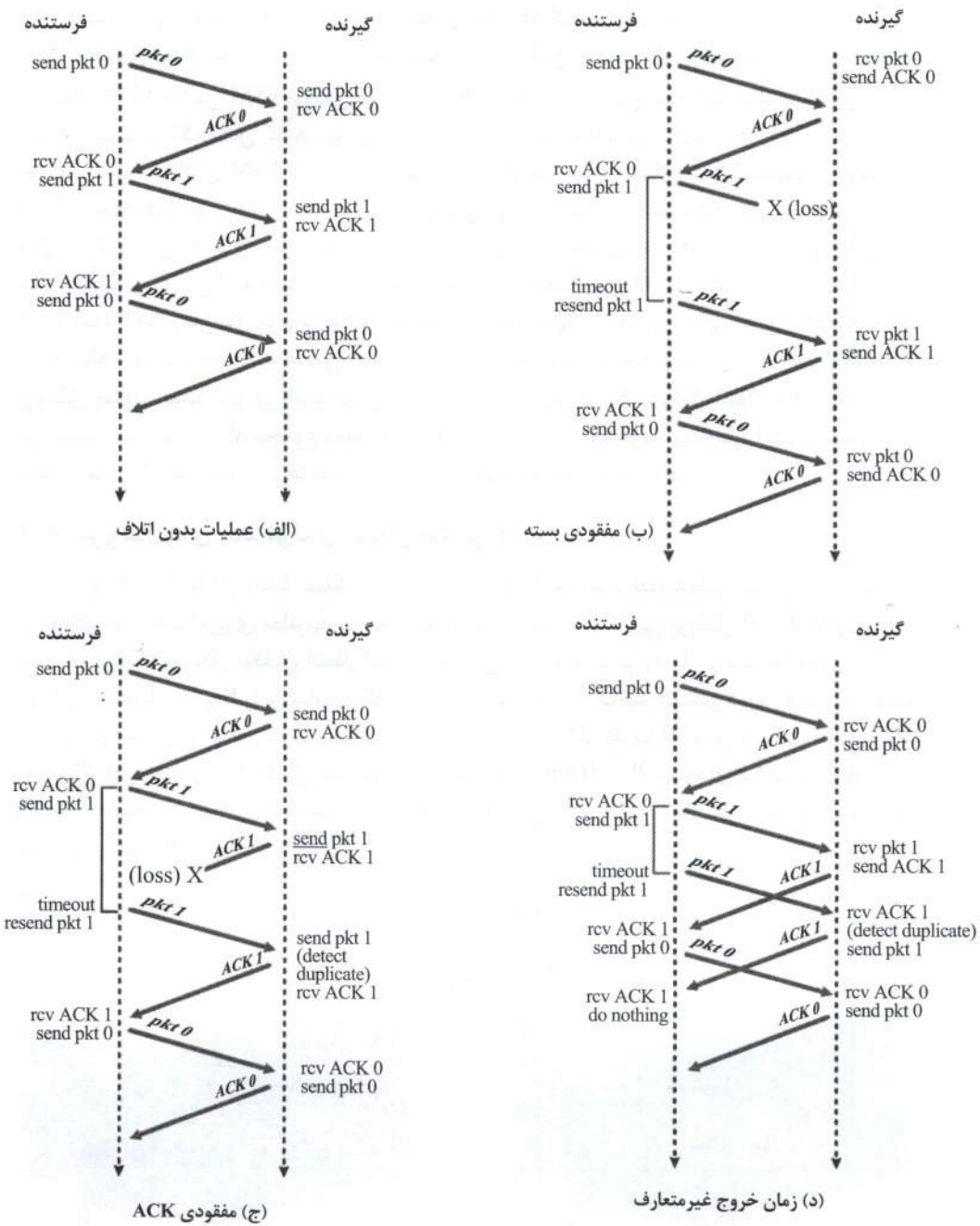


شکل ۳-۱۴ ♦ پروتکل گیرنده‌ی rdt 2.2

از نقطه‌نظر فرستنده انتقال مجدد یک نوش‌دارو به‌شمار می‌آید. فرستنده نمی‌داند که آیا یک بسته‌ی داده و یا یک ACK کدام یک تلف شده است و یا آنکه هر دو دچار یک تاخیر بیش از اندازه‌ی معین شده‌اند. در همه‌ی این موارد تاثیر مشابه است و آن عبارت است از انتقال مجدد بسته‌ها.



شکل ۳-۱۵ ♦ فرستنده‌ی پروتکل rdt 3.0



شکل ۳-۱۶ ♦ عملیات پروتکل rdt 3.0 و نیز پروتکل تناوب بیت

بنابراین فرستنده نیازمند انجام مراحل ذیل است:

۱- فعال‌سازی یک زمان‌سنج (timer) به هنگام ارسال یک بسته چه از لحاظ آنکه بسته مجدداً

ارسال می‌گردد و یا آنکه نخستین مرتبه‌ای است که گسیل می‌شود. ۲- واکنش نسبت به وقفه‌ای ایجاد شده از سوی یک زمان‌سنج و اتخاذ پاسخ مناسب. ۳- توقف زمان‌سنج.

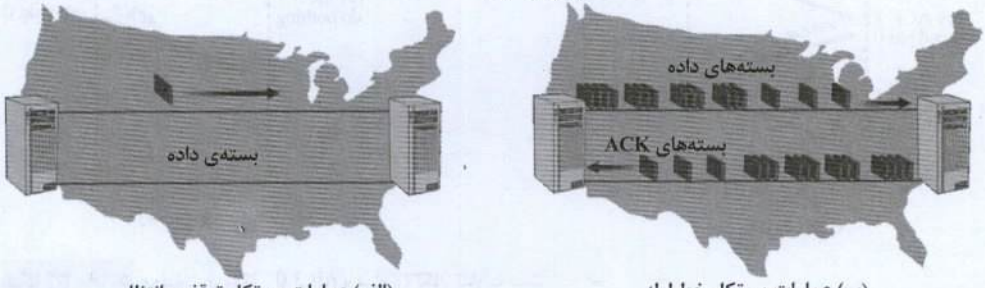
شکل ۳-۱۵ ماشین FSM پروتکل فرستنده‌ی rdt 3.0 را نشان می‌دهد که یک پروتکل انتقال مطمئن است. در یک کانال که هر دو ویژگی آلوده نمودن بسته‌ها و نیز اتلاف آن‌ها را دارد، بر عهده‌ی خواننده است که الگوی FSM گیرنده‌ی این پروتکل را طراحی نماید. شکل ۳-۱۶ عملکرد پروتکل را در قالب عدم مفقودی و تاخیر بسته‌ها و نیز چگونگی بازیابی بسته‌های تلف شده نشان می‌دهد در این شکل حرکت زمان به طرف جلو از بالا به پائین شکل می‌باشد. بدیهی است که زمان دریافتی برای یک بسته پس از ارسال آن لازم است. این فاصله‌ی زمانی در حقیقت زمان تأخیر انتشار است. در شکل‌های ۳-۱۶ (ب) تا (د) زمان‌های مربوط به آغاز عملیات و سپس انتهای پنجره‌ی زمانی نشان داده شده است. از آنجائیکه شماره ترتیب بسته‌ها بین 0 و 1 در حال تغییر می‌باشند، پروتکل rdt 3.0 را اصطلاحاً **پروتکل تناوب بیت**[1] نیز می‌نامند. در این بخش با عناصر کلیدی یک پروتکل انتقال داده آشنا شدیم. این عناصر عبارتند از: کُد مجموع مقابله (checksum) شماره ترتیب، زمان‌سنج‌ها و اعتبارسنجی‌های مثبت و منفی که هر یک از آن‌ها نقش حیاتی را در عملیات یک پروتکل ایفا می‌نمایند.

۳-۸ پروتکل‌های خط لوله‌ی انتقال مطمئن داده

اگرچه پروتکل rdt 3.0 از لحاظ عملکرد کاملاً بدون اشکال است اما از نقطه‌نظر کارایی با توجه به شبکه‌های پرسرعت امروزی مطلوب نمی‌باشد. نقطه ضعف اصلی در کارایی پروتکل rdt 3.0 این واقعیت است که آن یک پروتکل توقف و انتظار است. جهت پی بردن به تاثیر رفتار توقف و انتظار در یک پروتکل دو میزبان A و B را بر اساس شکل ۳-۱۷ در نظر بگیرید. تاخیر انتشار دوره رفت و برگشت بین این دو میزبان بر اساس سرعت حرکت نور در یک مسافتی اگر تقریباً 30 میلی‌ثانیه در نظر گرفته شود و اگر این دو میزبان از طریق یک پیوند با سرعت برابر R = 1Gbps بسته‌های با طول L = 1000 بایت را که شامل سرپیام و داده است انتقال دهند آنگاه زمان لازم جهت انتقال هر بسته به درون این پیوند برابر است با:

$$d_{trans} = \frac{L}{R} \qquad (۱-۳)$$

$$d_{trans} = \frac{1000 \times 8bit}{10^9 bit/s} = 8\mu s$$



(الف) عملیات پروتکل توقف و انتظار (ب) عملیات پروتکل خط لوله

شکل ۳-۱۷ ♦ عملکرد پروتکل توقف و انتظار در برابر پروتکل خط لوله

1- Alterneting bit protocol

شکل ۳-۱۸ (الف) نشان می‌دهد که در پروتکل توقف و انتظار اگر فرستنده در زمان $t = 0$ آغاز به ارسال بسته‌ای نماید آنگاه در زمان $t = \frac{L}{R} = 8\mu s$ آخرین بیت در سمت فرستنده وارد کانال می‌شود. بسته آنگاه یک پویش 15 میلی‌ثانیه‌ای را از مبدا به مقصد انجام می‌دهد به این گونه که آخرین بیت بسته در زمان 15.008 میلی‌ثانیه بر اساس محاسبه‌ی ذیل در گیرنده ظاهر می‌شود:

$$t = \frac{RTT}{2} + \frac{L}{R} \qquad (۲-۳)$$

$$t = \frac{30}{2} + 0.008 \Rightarrow t = 30.008ms$$

اگر به منظور آسان‌سازی موضوع فرض شود بسته‌های ACK از لحاظ طول بسیار کوچک بوده و بنابراین از زمان انتقال آن‌ها می‌توان صرف نظر نمود و نیز اگر گیرنده یک ACK را بی‌درنگ پس از دریافت آخرین بیت بسته گسیل نماید. آنگاه بسته‌ی ACK در نقطه‌ی ارسال نیز در زمانی معادل آنچه در ذیل محاسبه می‌شود به فرستنده عودت داده می‌شود:

$$t = RTT + \frac{L}{R} \Rightarrow t = 30.008ms$$

در این نقطه از زمان است که فرستنده می‌تواند پیام بعدی خود را گسیل نماید. بنابراین در زمان 30.008 میلی‌ثانیه فرستنده در حقیقت فقط معادل زمان 0.008 میلی‌ثانیه در حال ارسال اطلاعات بوده است. اگر بهره‌وری فرستنده یا پیوند را به‌عنوان جزیی از زمانی که فرستنده مشغول گسیل بیت‌ها به درون پیوند می‌باشد در نظر بگیریم آنگاه تحلیل شکل ۳-۱۸ (الف) نشان می‌دهد که پروتکل توقف و انتظار از نقطه‌نظر بهره‌وری فرستنده بسیار ناچیز می‌باشد:

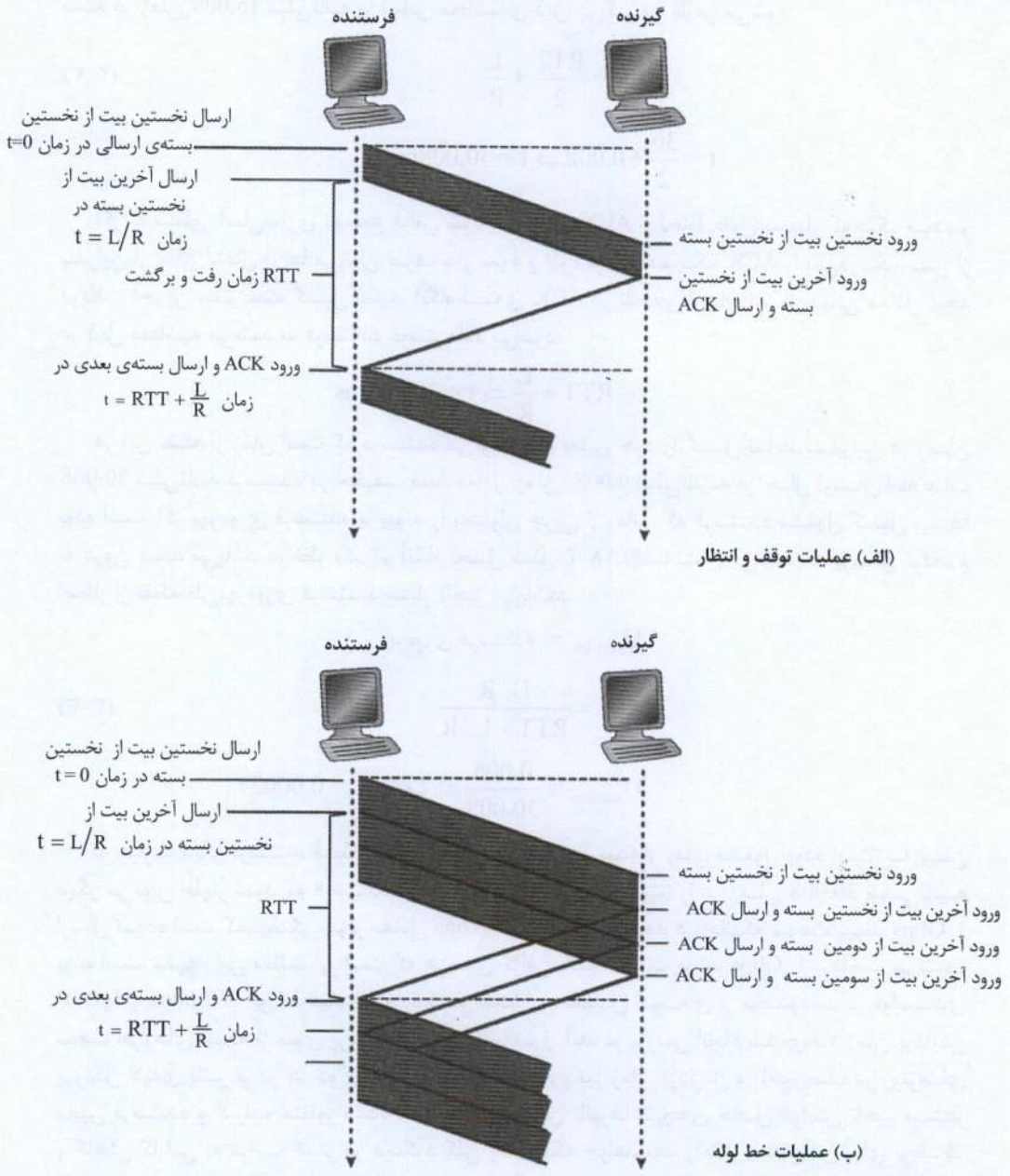
$$U_{sender} = \text{بهره‌وری فرستنده}$$

$$= \frac{L/R}{RTT + L/R} \qquad (۳-۳)$$

$$U_{sender} = \frac{0.008}{30.008} \Rightarrow U_{sender} = 0.00027$$

به عبارت دیگر فرستنده فقط 0.027 درصد (2.7 صدم درصد) از زمان مشغول بوده است. به بیان دیگر می‌توان اظهار نمود که فرستنده تعداد 1000 بایت (8000 بیت) را در زمان 30.008 میلی‌ثانیه ارسال نموده است که نهفتگی مؤثر معادل 267 kbps را نشان می‌دهد در حالیکه سرعت پیوند 1 Gbps بوده است. مفهوم این مطلب آن است که هزینه‌ی گزافی بابت ظرفیت پیوند 1 Gbps پرداخت می‌شود اما تنها از 267 kbps این ظرفیت استفاده شده است! این تحلیل نمونه‌ای از محدودیت در توانمندی سخت افزارهای شبکه از سوی پروتکل‌ها می‌تواند به‌شمار آید. در بررسی انجام شده‌ی بالا زمان پردازش پروتکل لایه‌ی پائین‌تر در هر دو سوی فرستنده و گیرنده و نیز زمان پردازش و تاخیر صف در روترهای مابین فرستنده و گیرنده منظور نشده است. با افزودن این تاثیرها نتیجه‌ی حاصل افزایش تاخیر بیشتر و کاهش کارایی به مراتب کمتر در عملکرد کلی یک شبکه خواهد بود. راهکار پیشنهادی برای برطرف نمودن این دشواری ساده می‌باشد. عملکرد به گونه‌ی توقف و انتظار به فرستنده اجازه می‌دهد چندین بسته بدون انتظار برای دریافت بسته‌ی ACK گسیل گردند. شکل ۳-۱۷ (ب) نشان می‌دهد اگر فرستنده مجاز به ارسال سه بسته قبل از انتظار برای تائید آن‌ها باشد آنگاه بهره‌وری فرستنده با ضریب

افزایش 3 مواجه خواهد شد. از آنجائیکه تعداد قابل توجهی از بسته‌ها در شرایط ترانزیت فرستنده به گیرنده قرار دارند که می‌توانند یک خط لوله را پر نمایند، این روش را اصطلاحاً **الگوی خط لوله‌ای**[1] می‌نامند.



شکل ۳-۱۸ ♦ ارسال به‌صورت خط لوله و روش توقف و انتظار

1- Pipelining approach

تأثیر الگوی خط لوله بر روی پروتکل‌های انتقال مطمئن داده را می‌توان به‌صورت ذیل عنوان نمود:

۱- محدوده‌ی شماره ردیف نیازمند گسترش است زیرا هر بسته در ترانزیت بدون در نظر گرفتن انتقال مجدد آن بایستی دارای یک شماره ردیف انحصاری باشد. این احتمال می‌رود که همواره تعدادی بسته در ترانزیت و به‌صورت تایید نشده وجود داشته باشند.

۲- سمت ارسال و دریافت پروتکل‌ها ممکن است به ذخیره‌سازی بیش از یک بسته اقدام نمایند. حداقل تعداد بسته‌هایی که فرستنده همواره بایستی ذخیره (بافر) نماید برابر آن تعداد بسته‌ای است که تائید آن‌ها را دریافت نکرده است. ذخیره‌سازی بسته‌های سالم دریافت شده ممکن است در گیرنده نیز روی دهد.

۳- محدوده‌ی ارزش شماره‌های ترتیب و نیز نیازهای ذخیره‌سازی بسته‌ها به چگونگی برخورد پروتکل با مواردی از قبیل مفقودی بسته، آلوده شدن آن و نیز تاخیرهای ناشی از انتقال دارد.

دو رویکرد جهت بازیابی خطا در روش خط لوله به کار گرفته می‌شود، این دو عبارتند از: (۱) روش بازگشت به عقب Go-Back-N و (۲) روش **تکرار گزینشی**[1].

۳-۹ روش بازگشت به عقب Go-Back-N (GBN)

در یک پروتکل GBN به فرستنده اجازه داده می‌شود چندین بسته را بدون انتظار برای تائید آن‌ها گسیل نماید. تعداد بسته‌های تائید نشده در خط لوله در هر زمان نبایستی از یک حداکثر مجاز تعداد N بسته فراتر رود. شکل ۳-۱۹ نگرش فرستنده را از نقطه‌نظر محدوده‌ی ارزش شماره ترتیب در یک پروتکل GBN نشان می‌دهد. اگر مبنا شماره ترتیب قدیمی‌ترین بسته‌ی تائید نشده در نظر گرفته شود و نیز کوچکترین شماره ردیف استفاده نشده یعنی شماره‌ی بسته‌ی بعدی جهت ارسال نیز با عنوان nextseqnum تعریف گردد آنگاه تعداد 4 تناوب در محدوده‌ی شماره‌های ردیف بسته می‌توان تعریف نمود. این چهار تناوب عبارتند از:

الف) تناوب [0,base-1] که به مفهوم بسته‌هایی است که گسیل شده‌اند و تائید آن‌ها نیز دریافت شده است.

ب) تناوب [base,nextseqnum] که به مفهوم بسته‌هایی است که گسیل شده‌اند اما تائید آن‌ها هنوز دریافت نشده است.

ج) تناوب [nextseqnum,base N-1] که از آن می‌توان جهت گسیل بی‌درنگ در صورت دریافت داده‌ها از لایه‌ی بالاتر استفاده نمود.

د) تناوب base + N ≤ **شماره ردیف** که از آن تا لحظه‌ای که تائید یک بسته‌ی جاری در خط لوله دریافت نشده است، نمی‌توان استفاده نمود. این حالت به‌ویژه مربوط به بسته‌ای می‌شود که شماره ردیف آن **مبنا** (base) باشد.

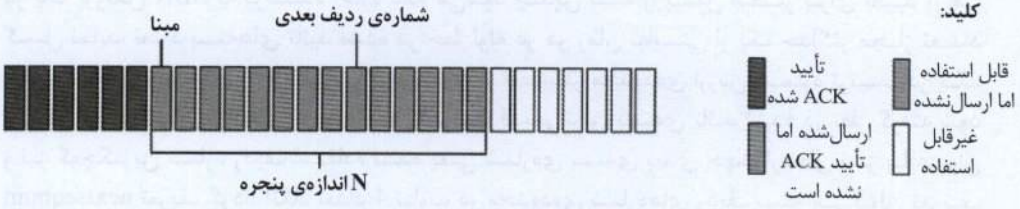
همانگونه که از شکل ۳-۱۹ می‌توان استنباط نمود محدوده‌ی ارزش قابل قبول شماره‌های ترتیب برای بسته‌های ارسال شده‌ای که تائید آن‌ها دریافت نشده است را می‌توان پنجره‌ای با اندازه‌ی N در نظر گرفت. هنگامی که پروتکل آغاز به فعالیت می‌کند این پنجره دچار لغزش به جلو از لحاظ فضای شماره ردیف می‌شود. به همین منظور N را اصطلاحاً **اندازه‌ی پنجره**[2] و پروتکل GBN را **پروتکل**

1- Selective repeat

2- Window size

پنجره‌ی لغزنده[1] نامیده‌اند. برای خوانندگان ممکن است این پرسش مطرح شود که چرا تعداد بسته‌های تائید نشده در ابتدا با یک ارزش ثابت N تعریف می‌شود و چرا تعداد چنین بسته‌هایی نامحدود در نظر گرفته نمی‌شود؟ پاسخ این پرسش‌ها را می‌توان در بحث کنترل جریان که در ادامه‌ی همین فصل آمده است جستجو نمود. علت دیگر این محدودیت به واسطه مسایل مربوط به کنترل تراکم در TCP می‌باشد که در ادامه‌ی موضوعات همین فصل بازهم مورد کنکاش قرار می‌گیرد.

در عمل شماره ردیف هر بسته در یک دامنه‌ی ثابت از نظر طول در سر پیام بسته حمل می‌شود. اگر تعداد بیت‌های دامنه‌ی شماره ردیف K فرض شود آنگاه محدوده‌ی این شماره برابر $[0, 2^K - 1]$ خواهد بود. در یک محدوده‌ی معین از این شماره‌ها آنگاه همگی ارزش‌های شماره ردیف‌ها از طریق محاسبه‌ی ریاضی تقسیم به پیمانه‌ی ($\text{modulo } 2^K$) بدست می‌آید. به عبارت دیگر فضای شماره‌های ردیف را می‌توان به گونه‌ی حلقه‌ای تصور نمود که اندازه‌ی آن 2^K می‌باشدو شماره ردیف $2^K - 1$ بی‌درنگ پس از شماره ردیف 0 قرار می‌گیرد. لازم به یادآوری است در پروتکل rdt 3.0 تنها از یک بیت برای این منظور استفاده می‌شد و بنابراین محدوده‌ی شماره‌های ترتیب در قالب [0,1] قرار می‌گرفت. در بخش ۳-۱۱ ملاحظه خواهد شد که پروتکل TCP دارای دامنه‌ی 32 بیت جهت این شماره‌ها می‌باشد. بنابراین در TCP شماره ردیف بایت‌ها در یک جریان بایت به جای بسته‌ها شمارش می‌شوند.



شکل ۳-۱۹ ♦ چشم‌انداز فرستنده از شماره‌های ردیف در پروتکل Go-Back-N

دو شکل ۳-۲۰ و ۳-۲۱ توصیفی گسترش یافته از ماشین‌های FSM دو سوی فرستنده و گیرنده را برای هر یک از پروتکل‌های ACK ، NAK آزاد و GBN نشان می‌دهند. این ماشین FSM اصطلاحاً FSM گسترش یافته نامیده شده است زیرا متغیرهایی مشابه آنچه در زبان‌های برنامه‌نویسی وجود دارد به شماره‌های base و nextseqnum اضافه شده‌اند. علاوه بر آن عملیاتی به گونه‌ی ساده و شرطی به متغیرها نیز افزوده شده است. با اندکی دقت ملاحظه می‌شود ویژگی‌های اضافه شده به FSM شباهت بسیاری به مشخصات یک زبان برنامه‌نویسی دارد. هر فرستنده‌ی GBN بایستی در برابر هر یک از سه رخداد زیر واکنش نشان دهد:

۱- **فراخوانی از بالا:** هنگامی‌که () rdt_send از لایه‌ی بالاتر فراخوانی می‌شود فرستنده ابتدا به ارزیابی اینکه آیا پنجره کامل (پُر) است می‌پردازد. به عبارت دیگر آیا تعداد بسته‌های تائید نشده برابر است با N یا خیر؟ اگر پنجره کامل نباشد، بسته‌ای خلق شده و سپس با ارسال آن متغیرهای مربوط به آن به روزرسانی می‌شوند. اگر پنجره پُر باشد فرستنده داده را به لایه‌ی بالاتر بازگشت می‌دهد و از این طریق به آن اعلام می‌دارد که پنجره در حال حاضر کامل است. لایه‌ی بالاتر مجدداً در زمانی دیگر تلاش جدیدی را در گسیل بسته‌ی داده به سمت پنجره

1- Sliding window protocol

می‌نماید. بدیهی است در یک پیاده‌سازی واقعی، فرستنده این بسته‌ی داده را ذخیره و یا از یک مکانیزم همزمانی مانند یک پرچم (نشان) بهره می‌گیرد. هنگامی‌که پنجره از یک حالت پُر بودن خارج گردید به لایه‌ی بالاتر اجازه می‌دهد تابع () rdt_send را فراخوانی نماید.

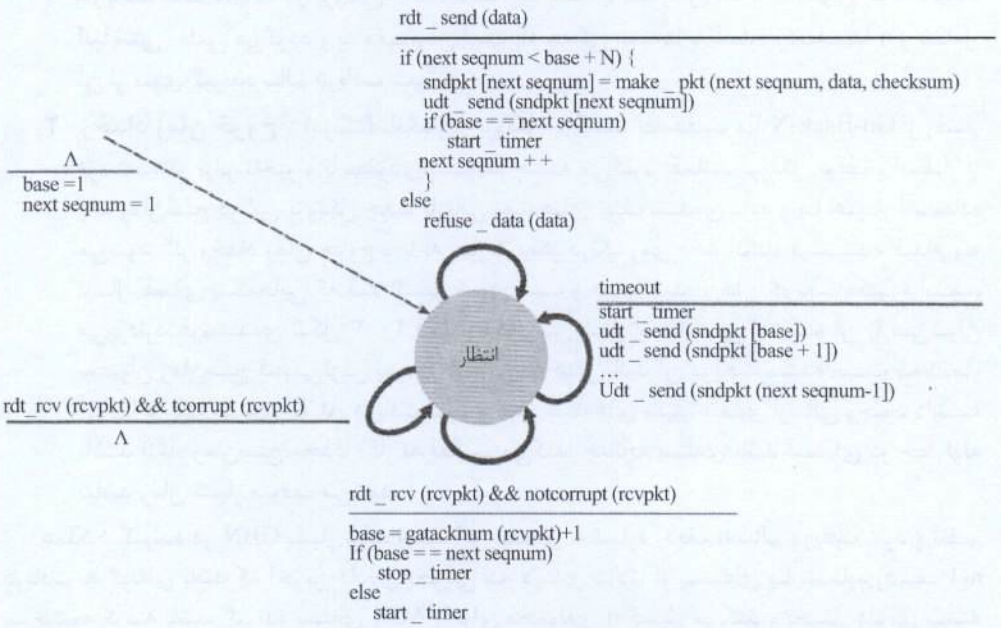
۲- **دریافت یک ACK:** در پروتکل GBN تائید یک بسته با شماره ردیف n به‌عنوان یک **تائید انباشتی**[1] تلقی می‌گردد و به مفهوم آن است که همگی بسته‌ها با شماره ردیف تا n و شامل آن از سوی گیرنده سالم دریافت شده‌اند.

۳- **رخداد زمان خروج**[2]: (درنگ) نامگذاری پروتکل بازگشت به عقب یا Go-Back-N از رفتار فرستنده در برابر تاخیر و یا مفقودی بسته سرچشمه می‌گیرد. همانند پروتکل توقف و انتظار از یک زمان‌سنج در این پروتکل جهت بازیابی بسته‌های تلف شده‌ی داده و یا اعتبار استفاده می‌شود. اگر رخداد زمان خروج و یا به عبارت دیگر درنگ روی دهد آنگاه فرستنده اقدام به ارسال همه‌ی بسته‌هایی که قبلاً گسیل نموده است و هنوز تائید آن‌ها را دریافت نکرده است، می‌پردازد. فرستنده‌ی شکل ۳-۲۰ فقط از یک زمان شمار استفاده می‌کند که آن را می‌توان به‌عنوان زمان‌سنج قدیمی‌ترین بسته‌ی ارسالی که هنوز تائید آن دریافت نشده است به‌شمار آورد. اگر یک بسته‌ی ACK دریافت شود و هنوز بسته‌های تائید نشده‌ی ارسالی وجود داشته باشند آنگاه زمان‌سنج مجدداً آغاز به فعالیت می‌کند. چنانچه بسته‌ی تائید نشده‌ای در خط لوله نباشد زمان شمار متوقف می‌شود.

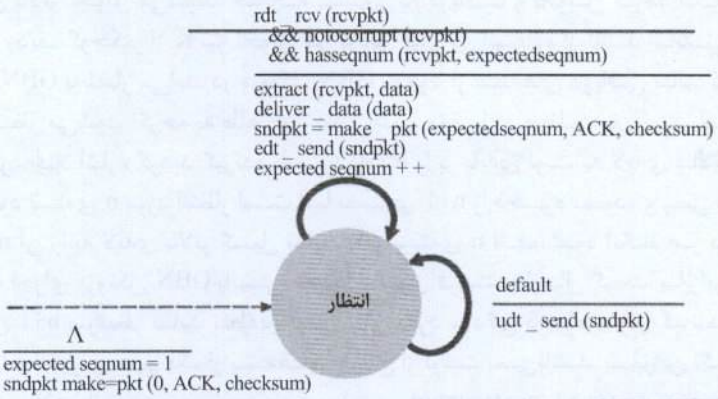
عملکرد گیرنده در GBN بسیار ساده است. اگر بسته‌ای با شماره ردیف n سالم دریافت شود و نظم دریافت به گونه‌ای باشد که آخرین داده‌ی تحویلی به لایه‌ی بالاتر از بسته‌ای با شماره ردیف n-1 سرچشمه گرفته باشد، گیرنده بسته‌ی ACK را برای محموله‌ی n گسیل می‌کند و بخش داده‌ی بسته را به لایه بالاتر تحویل می‌دهد. در بقیه‌ی موارد گیرنده از بسته صرفنظر نموده و یک پیام ACK را مجدداً برای جدیدترین بسته‌ی دریافتی گسیل می‌کند. بدیهی است از آنجائیکه بسته‌ها به گونه‌ی یک بسته در هر زمان به لایه‌ی بالاتر تحویل می‌شوند، چنانچه بسته‌ی K دریافت و تحویل گردد آنگاه همه‌ی بسته‌های با شماره ردیف کوچکتر از K نیز تحویل می‌گردند. بنابراین استفاده از تائید انباشتی یک گزینش طبیعی برای GBN به‌شمار می‌آید. در پروتکل GBN معمولاً از بسته‌های دریافتی سالم که خارج از نوبت هستند صرفنظر می‌شود. اگرچه به ظاهر این عمل غیر معقول به نظر می‌رسد، اما توجیهی برای آن وجود دارد. قبلاً اشاره گردید گیرنده بایستی داده‌ها را بر پایه‌ی نوبت به لایه‌ی بالاتر تحویل نماید. اگر فرض شود بسته‌ی n مورد انتظار است اما بسته‌ی n+1 را ذخیره نموده و پس از دریافت و تحویل بسته‌ی n آن را به لایه‌ی بالاتر گسیل نماید. واگر بسته‌ی n تلف شود آنگاه هر دو بسته‌ی n و n+1 در نتیجه اجرای پروتکل GBN بایستی مجدداً از سوی فرستنده ارسال گردند. بنابراین گیرنده می‌تواند از بسته ی n+1 صرفنظر نماید. نقطه‌ی قوت این رویکرد سادگی ذخیره‌سازی گیرنده است. دریافت کننده نیازمند ذخیره‌ی هیچ یک از بسته‌های خارج از نوبت نمی‌باشد. بنابراین اگر فرستنده بایستی محدوده‌ی بالا و پائین پنجره‌ی خود و نیز موقعیت nextseqnum را در پنجره حفظ نماید. جهت این هدف تنها قطعه‌ای از اطلاعات که گیرنده نیازمند نگهداری آن می‌باشد عبارت است از شماره ردیف متعلق به بسته‌ای که در نوبت بعد قرار دارد. این ارزش در متغیر expectedseqnum

1- Cummulative acknowledgment
2- Time out

نگهداری می‌شود که در ماشین FSM گیرنده‌ی شکل ۳-۲۱ نشان داده شده است. بدیهی است نقطه ضعف حذف یک بسته‌ی دریافتی سالم آن است که در انتقال مجدد، آن بسته ممکن است دچار خطا شده و یا مفقود گردد و بنابراین نیاز به ارسال مجدد افزون‌تری باشد.



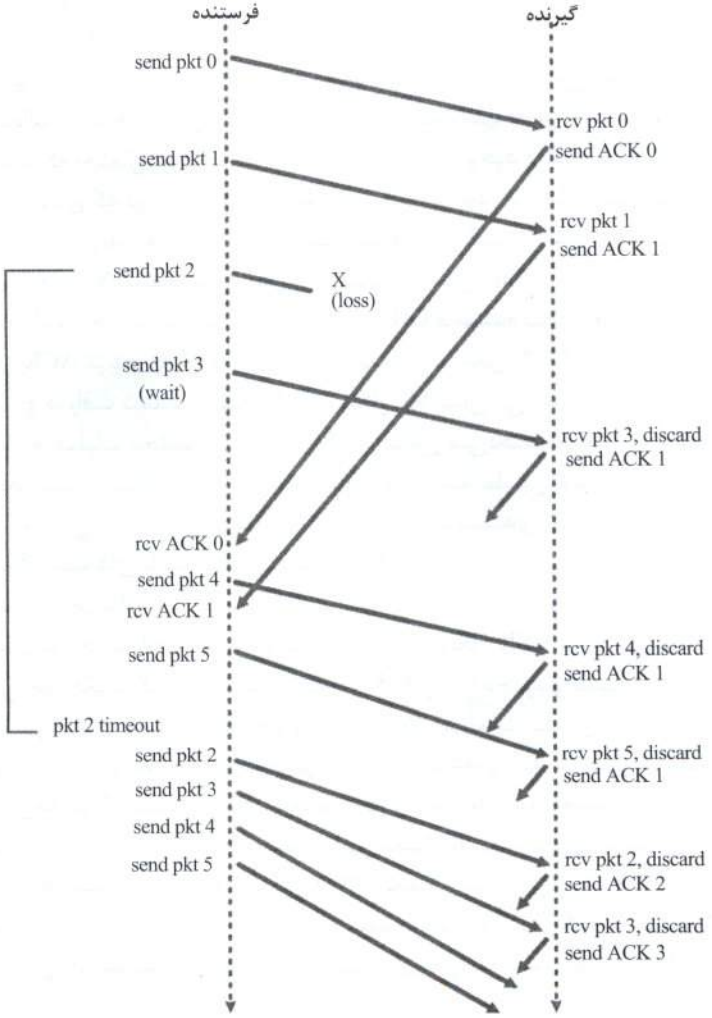
شکل ۳-۲۰ ♦ توصیف FSM گسترش یافته‌ی گیرنده پروتکل GBN



شکل ۳-۲۱ ♦ توصیف FSM گسترش یافته‌ی گیرنده پروتکل GBN

شکل ۳-۲۲ عملیات پروتکل GBN را برای حالتی نشان می‌دهد که یک پنجره با تعداد چهار بسته موردنظر می‌باشد. به علت این محدودیت در اندازه‌ی پنجره، فرستنده بسته‌های از 0 تا 3 را ارسال می‌نماید اما سپس بایستی در انتظار تائید بیش از یک بسته قبل از ادامه‌ی عملیات باشد. هنگامی‌که ACKهای پی در پی مانند ACK0 و ACK1 دریافت می‌شوند پنجره به طرف جلو لغزش می‌یابد و

فرستنده بسته‌های جدید pkt4 و pkt5 را به ترتیب گسیل می‌نماید. در سمت گیرنده بسته‌ی دوم مطابق شکل مفقود شده است و بنابراین بسته‌های 3، 4، 5 خارج از نوبت تلقی شده و از آن‌ها صرفنظر می‌شود. یادآوری این نکته ضروری است که به هنگام پیاده‌سازی این پروتکل در یک پشته‌ی پروتکلی ساختار آن مشابه FSM گسترش یافته‌ی شکل ۳-۲۰ می‌باشد. پیاده‌سازی آن در قالب انواع مراحل و توابع که رخدادهای مختلف را بایستی پوشش دهند انجام می‌شود. توابع مختلف درون این پروتکل از طریق مراحل موجود در پروتکل‌های دیگر و یا وقفه‌ها فراخوانی می‌شوند. در سمت فرستنده‌ی این رخدادها ممکن است: (۱) فراخوانی از سوی یک موجودیت لایه‌ی بالاتر برای احضار تابع () rdt_send باشد، (۲) یک وقفه‌ی ناشی از یک زمان‌سنج باشد و (۳) فراخوانی از لایه‌ی پائین‌تر جهت احضار () rdt_rcv به هنگام ورود یک بسته. نتیجه‌گیری حاصل از این بخش را می‌توان در پیاده‌سازی عوامل انتقال مطمئن داده در پروتکل TCP به‌شمار آورد که این عوامل شامل استفاده از شماره‌های ترتیب (ردیف)، اعتبارسنجی انباشتی، کُدهای مجموع مقابله و عملیات درنگ و ارسال مجدد می‌شوند.



شکل ۳-۲۲ ♦ عملیات پروتکل Go-Back-N

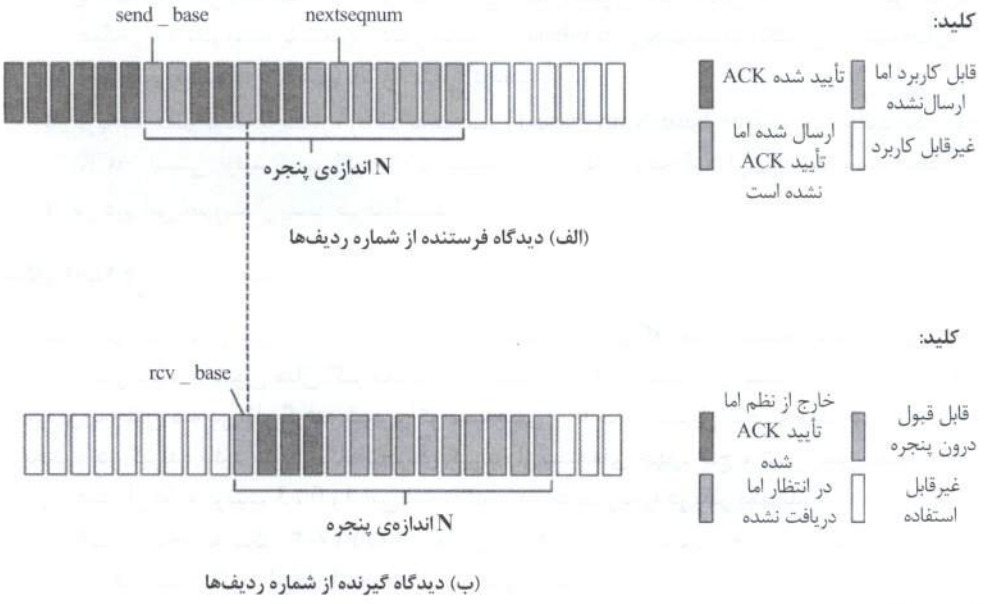
۳-۱۰ تکرار گزینشی[1] (SR)

پروتکل GBN بر اساس شکل ۳-۱۷ این امکان را به فرستنده می‌دهد که خط لوله را از بسته‌های داده تکمیل (پُر) نماید و در نتیجه مانع از اشکالات ایجاد شده به‌وسیله‌ی پروتکل‌های توقف و انتظار گردد. بدیهی است مواردی نیز وجود دارد که پروتکل GBN دچار اشکالات از این دست می‌شود به‌ویژه هنگامی‌که اندازه‌ی پنجره و عامل حاصل‌ضرب پهنای باند و زمان تاخیر که در فصل نخست از آن سخن به میان آمد دچار افزایش قابل توجهی شود. نتیجه‌ی رشد این دو عامل برای پروتکل GBN ماندگاری بسیاری از بسته‌های داده در خط لوله می‌باشد. بنابراین با وجود خطا بر روی تنها یک بسته، پروتکل GBN اقدام به ارسال مجدد تعداد قابل توجهی از بسته‌ها می‌نماید که نیازی به بسیاری از آن‌ها نمی‌باشد. هر چقدر که احتمال خطای کانال افزایش یابد خط لوله می‌تواند از این بسته‌های تکرار شده‌ی غیر ضروری پُر گردد. اگر یک پنجره با حجم 1000 کلمه در نظر گرفته شود هر مرتبه که یک کلمه دچار خطا گردد تمامی 1000 کلمه مجدداً انتقال داده می‌شوند. این موضوع خود سبب کاهش سرعت و سرانجام توقف مبادله خواهد شد.

همانگونه که از عنوان این بخش استنباط می‌شود پروتکل‌های تکرار گزینشی جهت جلوگیری از این موارد طراحی شده‌اند. اینگونه پروتکل‌ها سبب می‌شوند فرستنده تنها اقدام به تکرار انتقال آن گروه از بسته‌های داده بنماید که احتمال بروز خطا به هنگام دریافت آن‌ها وجود داشته است. به عبارت دیگر آن دسته از بسته‌های داده‌ای که در گیرنده مفقود شده‌اند و یا آنکه آلوده دریافت شده‌اند. این تکرار انتقال نیازمند آن است که گیرنده به‌صورت منفرد تائید بسته‌های دریافتی را اعلام نماید. باز هم در اینجا از یک پنجره‌ی با حجم N استفاده می‌شود که به کمک آن بتوان تعداد بسته‌های تائید نشده و در انتظار ارسال را در خط لوله محدود نمود. بر خلاف پروتکل GBN فرستنده برای بعضی از بسته‌های درون پنجره پیام‌های ACK مربوط به آن‌ها را دریافت داشته است. شکل ۳-۲۳ (الف) و (ب) به ترتیب موارد مربوط به ارسال و دریافت شماره ردیف‌ها را در پروتکل SR نشان می‌دهد. از طرف دیگر شکل ۳-۲۴ جزئیات مربوط به عملیات مختلف مورد نیاز SR را نشان می‌دهد. در این پروتکل همواره گیرنده‌ی SR تائید یک بسته‌ی سالم دریافت شده را بدون توجه به نظم و ترتیب بسته‌ها ارسال می‌نماید. بسته‌هایی که خارج از نوبت می‌باشند تا زمان دریافت یک بسته‌ی تلف شده در بافر باقی می‌مانند به عبارت دیگر بسته‌های با شماره‌های ردیف کوچکتر در بافر می‌مانند. نتیجه‌ی رویکرد آن است که یک گروه بسته را می‌توان با توجه به رعایت ترتیب آن‌ها به لایه‌ی بالاتر گسیل نمود. شکل ۳-۲۵ فهرستی از عملیات مختلف انجام شده به‌وسیله‌ی گیرنده پروتکل SR را نشان می‌دهد در حالیکه شکل ۳-۲۶ نشان می‌دهد چگونه گیرنده در آغاز بسته‌های 3، 4 و 5 را ذخیره و سپس آن‌ها را به همراه بسته ی شماره 2 که سرانجام دریافت می‌کند تحویل لایه‌ی بالاتر می‌نماید. نکته‌ی مهم آن است که در مرحله‌ی دوم از شکل ۳-۲۵ گیرنده به جای صرفنظر نمودن از بسته‌هایی که تا کنون دریافت نموده است و شماره ردیف آن‌ها نیز کوچکتر از مبنای پنجره‌ی جاری است به تائید مجدد آن‌ها می‌پردازد. با اندکی تامل می‌توانید خود را متقاعد نمائید که یقیناً این تکرار تائید ضروری است. با توجه به محدوده‌ی شماره ردیف فرستنده و گیرنده در شکل ۳-۲۳ به‌عنوان مثال اگر پیام ack برای بسته‌ی send-base که از سمت فرستنده به گیرنده گسیل شده است دریافت نگردد، آنگاه فرستنده سرانجام این بسته را مجدداً ارسال خواهد نمود اگر چه ممکن است برای ما واضح باشد که گیرنده آن را دریافت

1- Selective Repeat

نموده است اما برای فرستنده این حقیقت آشکار نمی‌باشد. اگر گیرنده به تائید بسته نمی‌پرداخت پنجره‌ی فرستنده هرگز به جلو حرکت نمی‌نمود. این مثال بیان کننده‌ی یک جنبه از پروتکل‌های SR و بسیاری از پروتکل‌های دیگر است. فرستنده و گیرنده همیشه دارای دیدگاهی مشترک از نقطه‌نظر آنچه که صحیح و یا نادرست دریافت نموده‌اند، می‌باشند. برای پروتکل‌های SR این به مفهوم آن است که پنجره‌های فرستنده و گیرنده همیشه با یکدیگر در انطباق نیستند.



شکل ۳-۲۳ ♦ پروتکل تکرار گزینشی فرستنده و گیرنده‌ی بسته‌های داده از نقطه‌نظر شماره ردیف

۱- **دریافت داده از بالا:** هنگامی‌که داده از بالا دریافت می‌شود. فرستنده‌ی SR شماره ردیف بعدی که در دسترس می‌باشد را برای بسته بررسی می‌کند. چنانچه شماره ردیف در محدوده‌ی پنجره‌ی فرستنده باشد آنگاه داده بسته‌بندی شده و گسیل می‌شود در غیر اینصورت داده بافر و یا به لایه‌ی بالاتر جهت انتقال در زمان دیگری مشابه با پروتکل GBN عودت داده می‌شود.

۲- **زمان خروج (Time out):** از زمان‌سنج‌ها برای مقابله‌ی مفقودی بسته‌ها استفاده می‌شود. بدیهی است هر بسته بایستی دارای زمان‌شمار منطقی خود باشد زیرا تنها یک بسته به هنگام فرا رسیدن زمان خروج گسیل خواهد شد. از یک زمان‌سنج سخت‌افزاری می‌توان برای شبیه‌سازی عملیات چندین زمان‌شمار منطقی بهره گرفت.

۳- **دریافت پیام ACK:** به هنگام دریافت یک پیام ACK، پروتکل SR فرستنده بسته‌ی در حال دریافت را مشروط بر آنکه در پنجره موجود باشد نشان‌گذاری می‌نماید. اگر شماره ردیف بسته برابر send-base در بسته باشد آنگاه ارزش مبنای پنجره به سمت بسته‌ی تائید نشده‌ای که حامل کوچکترین شماره ردیف است حرکت داده می‌شود. چنانچه پنجره حرکت نماید و بسته‌های انتقال نیافته‌ای با شماره ردیف‌های در محدوده‌ی پنجره وجود داشته باشند آنگاه این بسته‌ها منتقل می‌شوند.

شکل ۳-۲۴♦ رخدادها و عملیات پروتکل SR فرستنده

۱- دریافت صحیح بسته‌ها با شماره ردیف موجود در [rcv-base,rcv-base+N-1] . در این حالت بسته‌ی دریافت شده در محدوده‌ی گیرنده قرار می‌گیرد و یک بسته‌ی ACK گزینشی به فرستنده عودت داده می‌شود. اگر بسته قبلاً دریافت نشده باشد آنگاه ذخیره می‌شود. اگر این بسته دارای شماره ردیف معادل مبنای پنجره‌ی دریافتی یعنی rcv-base در شکل ۳-۲۲ باشد آنگاه این بسته و هر بسته‌ی ذخیره شده‌ی قبلی با رعایت نوبت شماره ردیف آن‌ها که آغاز آن با rcv-base می‌باشد به لایه بالاتر گسیل می‌شوند. به‌عنوان مثال شکل ۳-۲۶ را در نظر بگیرید. هنگامی‌که یک بسته با شماره ردیفی مانند rcv-base=2 دریافت شود، آنگاه آن و بسته‌های 3، 4، و 5 می‌توانند به لایه‌های بالاتر منتقل گردند.

۲- دریافت سالم بسته با شماره ردیف به‌صورت [rcv-base-N,rcv-base-1] در این حالت یک پیام ACK بایستی تولید گردد، اگر چه این بسته‌ای باشد که گیرنده قبلاً آن را تائید نموده است.

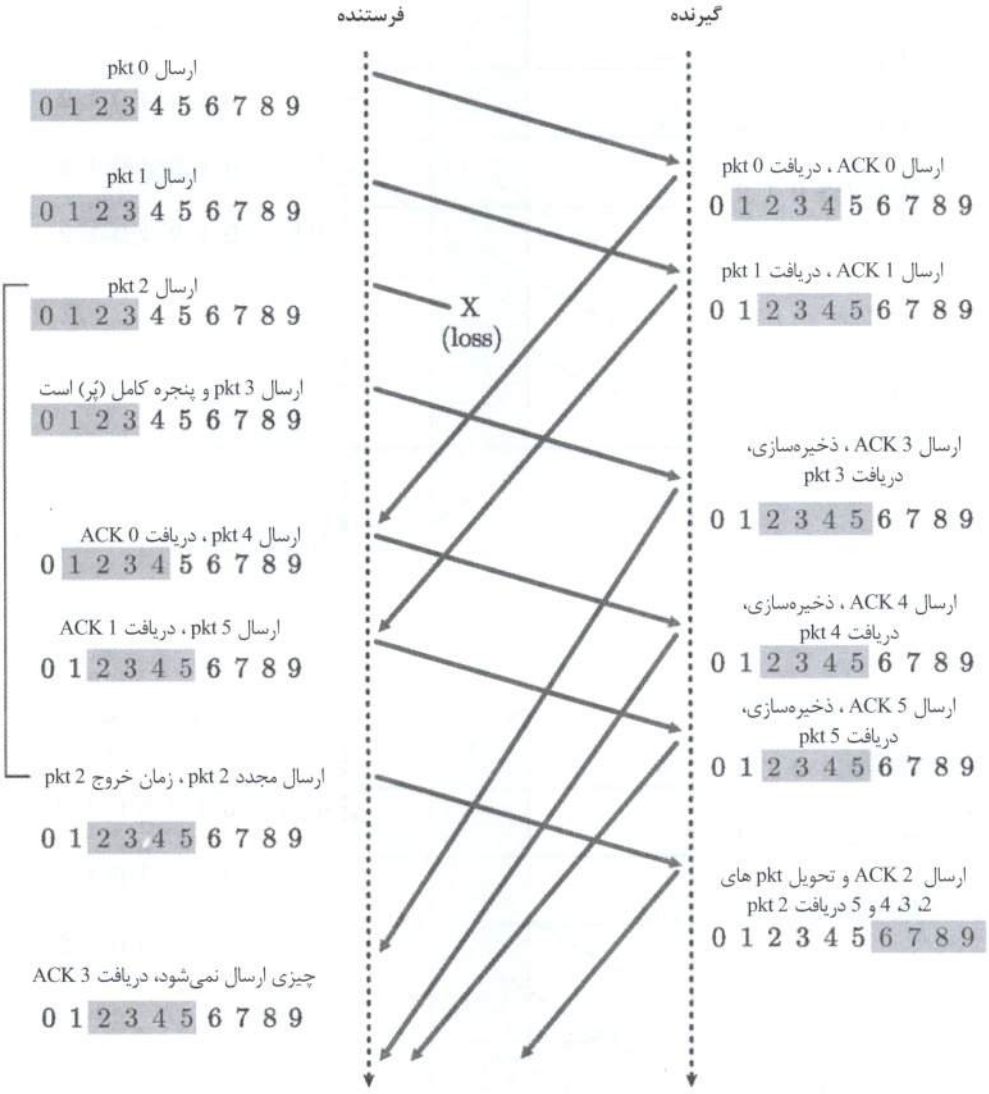
۳- در غیر این صورت: از بسته صرفنظر شود.

شکل ۳-۲۵ ♦ رخدادها و عملیات پروتکل SR گیرنده

عدم وجود همزمانی بین پنجره‌های فرستنده و گیرنده هنگامی‌که تعداد ردیف‌ها محدود می‌باشد بسیار اثرگذار است. به‌عنوان مثال اگر محدوده‌ی چهار بسته با شماره ردیف‌های 0 ، 1 ، 2 ، 3 و پنجره‌ای به طول سه در نظر گرفته شود و اگر بسته‌های از 0 تا 2 گسیل گردند و به‌صورت سالم دریافت و در گیرنده تائید شوند آنگاه پنجره‌ی گیرنده از بسته‌های چهار، پنج و شش عبور نموده زیرا شماره ردیف آن‌ها به ترتیب 3 ، 0 و1 می‌باشد. اکنون دو حالت زیر را در نظر بگیرید:

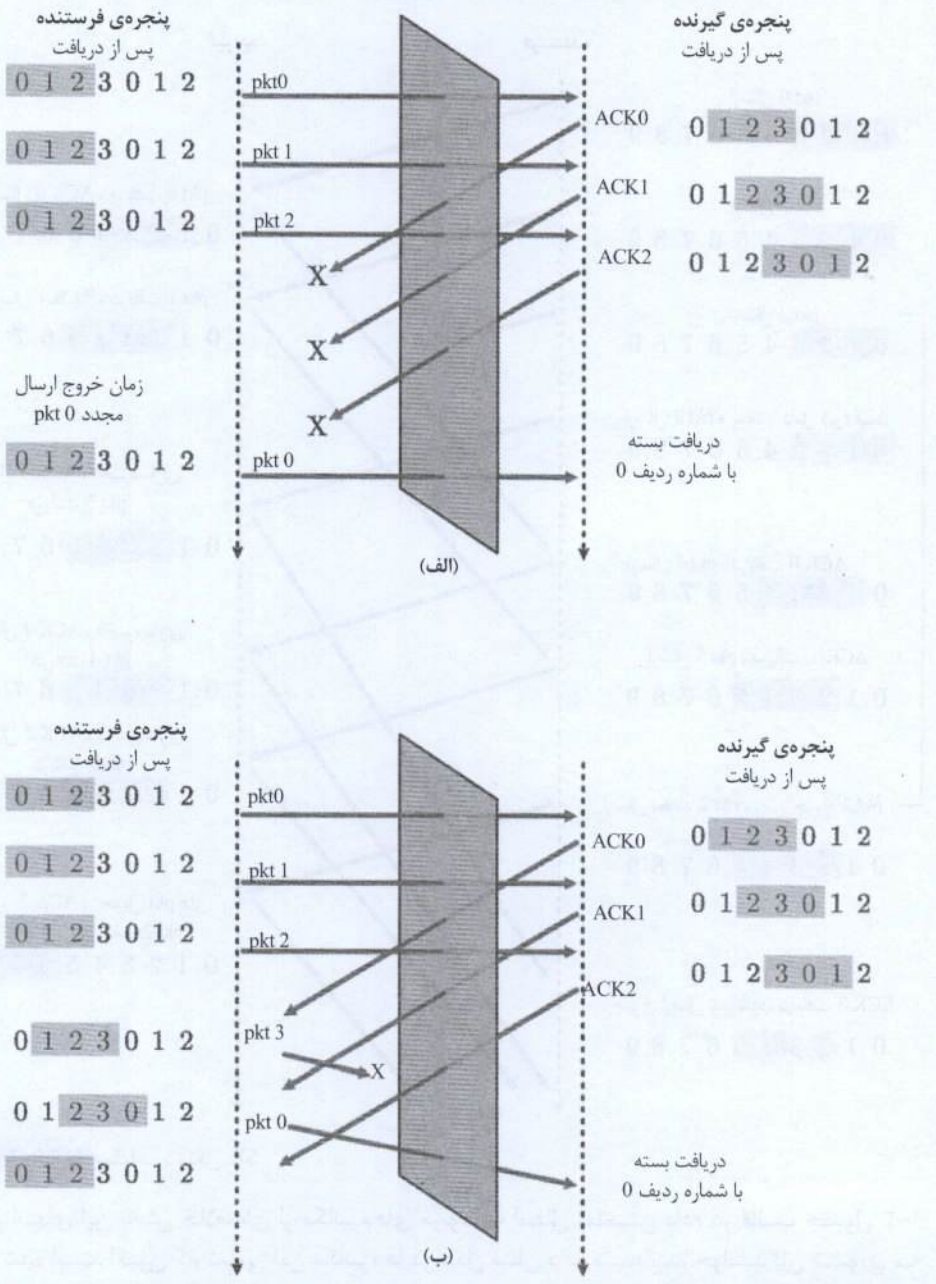
الف) با توجه به شکل ۳-۲۷ (الف) پیام‌های ACK برای سه بسته‌ی نخستین مفقود شده‌اند و فرستنده ناگریز آن‌ها را مجدداً ارسال می‌نماید. پس از آن گیرنده بسته‌ای را دریافت می‌کند که شماره ردیف آن 0 می‌باشد به عبارت دیگر نسخه‌ای را از نخستین بسته‌ی ارسال شده دریافت می‌کند.

ب) با توجه به شکل ۳-۲۷ (ب) پیام‌های ACK برای سه بسته ی نخستین سالم تحویل گیرنده شده‌اند. فرستنده در این حالت پنجره را به جلو رانده و بسته‌های چهار، پنج و شش را گسیل می‌کند که هر یک از آن‌ها به ترتیب دارای شماره ردیف‌های 3 ، 0 و 1 می‌باشند. بسته ی با شماره ردیف 3 مفقود شده است اما بسته‌ی با شماره ردیف 0 به همراه داده‌های جدید وارد می‌شود. اکنون شکل ۳-۲۷ را از نظر نقطه‌نظر گیرنده بررسی نمائید که دارای یک پرده‌ی تصویری مشابه آنچه در شکل نشان داده شده است بین گیرنده و فرستنده می‌باشد. از آنجائیکه گیرنده قادر به رویت عملیات انجام شده توسط فرستنده نمی‌باشد بنابراین آنچه گیرنده مشاهده می‌کندردیفی از پیام‌هایی است که از سوی کانال دریافت و یا به درون آن گسیل می‌نماید. بنابراین از نقطه‌نظر گیرنده دو حالت (الف) و (ب) در شکل ۳-۲۷ هر دو مشابه یکدیگر می‌باشند. به هیچ روی نمی‌توان تمایزی را بین انتقال مجدد نخستین بسته و انتقال پنجمین بسته در یک انتقال اصلی تشخیص داد. بدیهی است پنجره‌ای که اندازه‌ی آن به مقدار 1 (یک) کمتر از اندازه‌ی دامنه‌ی شماره ردیف‌ها باشد بی‌حاصل است و عمل نمی‌کند. اکنون این پرسش مطرح می‌شود که اندازه‌ی پنجره بایستی حداقل کوچک‌تر و یا برابر نیمی از تعداد شماره ردیف‌ها برای پروتکل SR باشد. آیا می‌توانید این واقعیت را برای خود توجیه نمائید.



شکل ۳-۲۶ ♦ عملیات پروتکل SR

در انتهای این بخش خلاصه‌ای از مکانیزم‌های مربوط به انتقال مطمئن داده در قالب جدول ۳-۱ آورده شده است. اکنون که تمامی این مکانیزم‌ها در عمل نشان داده شده‌اند، خوانندگان تشویق به بازنگری مجدد این بخش شده، این عمل به قصد شناسایی مرحله به مرحله‌ی افزایش پیچیدگی اعمال‌شده در الگوهایی می‌باشد که فرستنده و گیرنده را به یکدیگر پیوند می‌دهند. نکته‌ی آخر آنکه در ابتدای مطالب این بخش فرض در نظر گرفته شده مبنی بر آن بود که بسته‌ها را در بین یک پیوند انتقال بین فرستنده و گیرنده نمی‌توان به نظم درآورد. هنگامی‌که فرستنده و گیرنده از طریق یک رشته سیم (کابل) فیزیکی به یکدیگر متصل شده‌اند عموماً این فرض قابل قبول است اما هنگامی‌که پیوند ارتباطی این دو یک شبکه است به نظم درآوردن بسته‌ها امکان‌پذیر می‌باشد.



شکل ۳-۲۷ ♦ پروتکل گیرنده SR با پنجره‌های بسیار بزرگ. چالش در اینجا یک بسته‌ی جدید یا انتقال مجدد است.

یکی از روش‌های به ترتیب درآوردن بسته‌ها از این حقیقت ناشی می‌شود که نسخه‌های قدیمی از یک بسته با یک شماره ردیف و یا تائید ACK مانند x می‌تواند ظهور نماید، اگرچه پنجره‌های فرستنده و گیرنده هیچکدام حامل x نباشند. فرض امکان به نظم درآوردن بسته‌های یک کانال را می‌توان به گونه‌ی ذخیره‌ساز بسته‌ها که در هر لحظه می‌تواند آن‌ها را در نقطه‌ای در آینده خارج نماید، تصور

نمود. از آنجائیکه شماره ردیف‌ها ممکن است مجدداً به کار گرفته شوند دقت کافی جهت حفاظت در برابر تکثیر بسته‌ها لازم می‌باشد. در عمل رویکرد به کار گرفته شده بر اساس این واقعیت پایه‌ریزی می‌شود که شماره ردیف تا حصول اطمینان فرستنده از اینکه بسته‌های ارسال شده‌ی قبلی با شماره ردیف x دیگر در شبکه حضور ندارد، بسته‌ها مجدداً مورد استفاده قرار نمی‌گیرند. مکانیزم انجام این مهم حصول اطمینان از ادامه‌ی حیات یک بسته بیش از یک حداکثر زمان معین در شبکه می‌باشد. در پروتکل TCP به منظور دستیابی به حداکثر سرعت در شبکه، زمان در نظر گرفته شده برای حیات یک بسته ی 3 دقیقه می‌باشد. سند RFC1323 به توصیف روش استفاده از شماره ردیف‌ها جهت ممانعت از دشواری‌های حاصل در اثر به نظم درآوردن بسته‌ها می‌پردازد.

جدول ۳-۱ ♦ خلاصه‌ای از مکانیزم‌های انتقال مطمئن داده و کاربرد آن‌ها

مکانیزم	توضیح کاربرد
کُد مجموع مقابله‌ای (checksum)	کاربرد جهت آشکارسازی بیت‌های خطا در یک بسته‌ی ارسال شده است.
زمان‌سنج (Timer)	استفاده جهت زمان خروج و یا انتقال مجدد بسته به دلایل مختلف از جمله : مفقود شدن بسته و یا پیام ACK در کانال می‌باشد. علاوه بر این زمان خروج می‌تواند به هنگام تاخیر یک بسته نیز روی دهد که اصطلاحاً به آن زمان خروج زود هنگام گفته می‌شود و یا آنکه بسته به‌وسیله‌ی گیرنده دریافت شده است اما پیام ACK آن در طول مسیر از گیرنده به فرستنده مفقود گردیده است و سرانجام آنکه نسخه‌های تکثیر شده‌ی بسته از سوی گیرنده دریافت شده است.
شماره ردیف	استفاده جهت شماره‌گذاری ردیفی بسته‌های جاری از فرستنده به گیرنده فاصله‌ی ایجاد شده در شماره ردیف بسته‌ها سبب می‌شود گیرنده از مفقود شدن بسته‌ها آگاه گردد. بسته‌هایی که شماره ردیف آن‌ها تکرار گردد به گیرنده این فرصت را می‌دهند که نسخه‌های تکراری یک بسته را تشخیص دهد.
پیام تائید (ACK)	گیرنده از آن به هدف آگاه نمودن فرستنده از دریافت بدون عیب یک بسته‌ی دریافتی استفاده می‌کند. پیام‌های ACK حامل شماره ردیف بسته و یا بسته‌هایی هستند که تائید شده‌اند. این پیام‌ها می‌توانند به گونه‌ی منفرد و یا انباشتی باشند و وابسته به نوع پروتکل کاربردی هستند.
پیام تائید منفی (Negative ACK)	گیرنده از این پیام جهت آگاه‌سازی فرستنده از عدم دریافت سالم یک بسته استفاده می‌کند. اینگونه پیام‌ها حامل شماره ردیف بسته‌ای هستند که سالم دریافت نشده است.
پنجره و خط لوله	در این روش فرستنده ممکن است محدود به ارسال بسته‌هایی باشد که شماره ردیف آن‌ها در یک محدوده‌ی ویژه قرار می‌گیرند. با ارسال همزمان چندین بسته بدون انتظار جهت تائید، بهره‌وری فرستنده در مقابل روش توقف و انتظار افزایش می‌یابد. اندازه‌ی پنجره بر اساس توانمندی گیرنده در دریافت و یا ذخیره‌ی پیام‌ها و یا سطح تراکم شبکه قابل تنظیم می‌باشد.

۳-۱۱ انتقال اتصال‌گرای (TCP)

قبلاً اشاره گردید شبکه‌ی اینترنت از پروتکل اتصال‌گرای TCP به‌عنوان مکانیزم انتقال مطمئن داده‌ها استفاده می‌کند. اصول بنیادی این پروتکل در اسناد RFC به شماره‌های 793، 1122، 1323، 2018 و 2581 با جزئیات کامل تشریح شده است. پروتکل TCP از آن جهت اتصال‌گرا خوانده می‌شود زیرا یک فرآیند کاربردی قبل از آغاز ارسال داده‌ها به فرآیند دیگر، بایستی اقدام به عملیات دست‌دهی که در فصل‌های گذشته به آن اشاره رفت، بنماید. پس از انجام این مهم است که اصطلاحاً یک ارتباط TCP بین دو کاربرد برقرار می‌گردد. بدیهی است ارتباط TCP را نمی‌توان همانند یک مدار انتها به انتها در TDM و یا FDM مانند شبکه‌های راهیابی مدار به‌شمار آورد. علاوه بر آن یک ارتباط TCP یک مدار مجازی VCP نیز نمی‌تواند قلمداد گردد. زیرا نقاط ارتباط در دو سیستم انتهایی قرارگرفته است. از آنجائیکه پروتکل TCP فقط در سیستم‌های انتهایی شبکه اجرا می‌شود و عناصر واسطی شبکه مانند مسیریاب‌ها و راهیاب‌های لایه پیوند از آن بهره نمی‌برد در نتیجه عناصر واسطی شبکه نمی‌توانند به نگهداری حالت ارتباط TCP بپردازند. در حقیقت مسیر یاب‌های واسطی به طور کامل نسبت به ارتباط‌های TCP بیگانه می‌باشند. آنها فقط داده گرام‌ها را و نه ارتباط‌ها را مشاهده می‌کنند. آنچه یک ارتباط TCP تأمین می‌کند، یک سرویس دو طرفه‌ی همزمان است. چنانچه یک ارتباط TCP بین فرآیند A در یک میزبان و فرآیند B در میزبان دیگری وجود داشته باشد آنگاه، داده‌های یک لایه‌ی کاربرد می‌تواند از سوی فرآیند A به فرآیند B همزمان با جاری شدن داده‌هایی از B به سوی A میسر گردد. در نتیجه یک ارتباط TCP همواره حالت نقطه به نقطه را اختیار می‌کند به عبارت دیگر داده‌ها بین یک فرستنده و گیرنده‌ی منفرد جاری می‌شوند. آنچه قبلاً به نام فرآیند چند پخشی که عبارت است از انتقال داده در یک عملیات گسیل از یک فرستنده به تعداد قابل توجهی گیرنده بود در پروتکل TCP امکان‌پذیر نمی‌باشد، به بیان ساده‌تر در TCP دو میزبان همراه، اما سه میزبان مزاحم یکدیگر می‌باشند. قبلاً اشاره گردید چگونه پروتکل TCP نسبت به ایجاد یک ارتباط بین دو فرآیند A و B عمل می‌کند. در فصل دوم و در برنامه‌ی مشتری جاوا که در آن فصل آورده شد ملاحظه گردید پروتکل TCP جهت برقراری ارتباط بین دو فرآیند از فرمانی مشابه آنچه در ذیل آمده است بهره می‌گرفت.

```
Socket clientSocket = newSocket ("hostname" , portNumber);
```

در این دستور جاوا نام سرور hostname و شناسایی فرآیند بر روی سرور به‌وسیله‌ی اطلاعات مربوط به PortNumber انجام می‌گرفت. پس از پردازش این دستورات لایه‌ی انتقال مشتری نسبت به برقراری ارتباط با سرور اقدام می‌نماید. در انتهای این بخش جزئیات چگونگی ایجاد ارتباط تشریح شده است. اما به شکل خلاصه این فرآیند به این گونه انجام می‌شود که ابتدا برنامه‌ی کاربردی مشتری یک قطعه‌ی TCP ویژه را ارسال می‌کند و سرور در پاسخ به آن قطعه‌ی دوم TCP ویژه‌ای را گسیل می‌نماید و سرانجام مشتری با قطعه‌ی سوم TCP پاسخ آن را می‌دهد.

دو قطعه‌ی نخست به همراه خود حامل هیچگونه محموله‌ای نمی‌باشند اما قطعه‌ی سوم ممکن است همراه خود محموله‌ی داده‌ی متعلق به لایه‌ی کاربرد را حمل نماید. از آنجائیکه تعداد قطعه‌های ارسالی بین دو میزبان سه می‌باشد، این مرحله‌ی برقراری ارتباط را اصطلاحاً **دست‌دهی سه جانبه**[1] نام نهاده‌اند.

هنگامی‌که یک ارتباط TCP برقرار شد دو پردازش کاربردی به مبادله‌ی داده‌ها بین خود

1- Three eay handshake

می‌پردازند. اگر فرآیند ارسال داده از سوی برنامه مشتری به سرور در نظر گرفته شود آنگاه برنامه‌ی مشتری یک جریان داده را از طریق سوکت عبور می‌دهد با این عمل داده در اختیار TCP در حال اجرا بر روی برنامه‌ی مشتری قرار می‌گیرد. در شکل ۳-۲۸ نشان داده شده است که TCP این داده را به سمت **بافر گسیل**[1] ارتباط هدایت می‌کند. این بافر به هنگام انجام عملیات دست‌دهی سه جانبه خلق می‌شود. در مقاطع زمانی گوناگون TCP بخشی از داده‌ی درون بافر گسیل را اخذ و آنها را در قالب یک قطعه‌ی TCP ارسال می‌نماید. در مستند RFC793 قید گردیده است پروتکل TCP بنابر میل خود و در زمان مناسبی که گزینش می‌کند، عملیات اشاره شده را انجام می‌دهد. حداکثر طول داده‌ای که TCP از بافرگسیل اخذ و آن را در یک قطعه قرار می‌دهد از سوی آنچه اصطلاحاً **حداکثر اندازه‌ی قطعه**[2] (MSS) نامیده شده است محدود می‌گردد. معمولاً اندازه‌ی MSS از طریق شناسایی طول بزرگترین قاب لایه‌ی پیوند که میزبان محلی قادر به ارسال آن می‌باشد و به آن **حداکثر جزء انتقال**[3] **(MTU)** گفته می‌شود، محاسبه می‌گردد. پس از اندازه‌گیری MTU است که بزرگی MSS تعریف می‌شود تا این اطمینان حاصل گردد که یک قطعه‌ی TCP به هنگام کپسول‌بندی در یک داده گرام IP در قالب لایه ی پیوند داده قرار می‌گیرد. اندازه‌های متداول عبارتند از 1460، 536، 512 بایت. یادآوری این نکته ضروری است که MSS حداکثر اندازه‌ی داده‌ی متعلق به برنامه‌ی لایه‌ی کاربردی در قطعه می‌باشد و نبایستی آن را به‌عنوان حداکثر طول قطعه‌ی TCP به همراه سر پیام آن به شماره آورد. اگر چه استفاده از واژه‌ی قطعه در هر دو مورد گمراه کننده می‌باشد و لیکن تمایز بین آن‌ها بسیار با اهمیت است.

پروتکل TCP با افزودن سر پیام TCP به هر قطعه‌ی داده مشتری آن را تبدیل به قطعه‌های TCP می‌نماید. این قطعه‌ها سپس به لایه‌ی شبکه گسیل می‌شوند و در آنجا در قالب داده گرام‌های IP لایه شبکه کپسول‌بندی می‌شوند. داده گرام‌های IP آنگاه وارد شبکه می‌شوند. هنگامی‌که TCP قطعه‌ای را در طرف دیگر دریافت نمود آن را در بافر دریافت خود قرار می‌دهد. بر اساس آنچه در شکل ۳-۲۸ نشان داده شده است برنامه‌ی کاربردی طرف مقابل جریان داده‌ی دریافتی را از این بافر می‌خواند. ملاحظه می‌شود هر طرف ارتباط دارای بافر انحصاری خویش می‌باشد. آنچه به اختصار می‌توان نتیجه‌گیری نمود آن است که هر ارتباط TCP شامل متغیرها، بافرها و یک سوکت ارتباط به یک فرآیند در یک میزبان و مجموعه‌ای از بافرها، متغیرها و یک سوکت ارتباط به یک فرآیند در میزبان دیگر است. بدیهی است در تجهیزات شبکه مانند راهیاب‌ها، مسیریاب‌ها و تکرار کننده‌ها هیچگونه بافر و متغیری برای ارتباط آن‌ها در بین میزبان‌ها تعریف نمی‌شود.

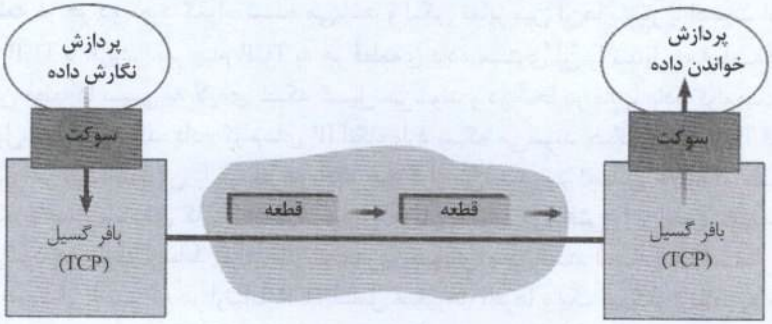
• یادآوری تاریخی

در سراسر فصولی که تاکنون بررسی شده است همواره از واژه‌های TCP و IP و به طور کلی پشته‌ی پروتکلی TCP/IP نام برده شده است بدون آنکه به سرچشمه‌ی پیدایش آن‌ها اشاره‌ای رفته باشد. بنابراین آنچه به دنبال این سطور می‌آید تاریخچه‌ای از این رویداد می‌باشد. در اویل دهه‌ی 1970 میلادی شبکه‌های راهیابی بسته (PSN) با گسترش چشمگیری رو به رو شدند. در کنار این شبکه‌ها، ARPANET نیز که آن را می‌توان پیش درآمد شبکه‌ی اینترنت امروزی به‌شمار آورد، قرار می‌گرفت. هر یک از این شبکه‌ها از پروتکل‌های کاربردی ویژه‌ی خود بهره می‌گرفتند. در این مقطع تاریخی دو

1- Send buffer
2- Maximum segment size
3- Maximum transfer unit

پژوهشگر به نام‌های کِرف (Cerf) و کَن (Kahn) به اهمیت ارتباط بین این شبکه‌ها پی بردند و به خلق یک پروتکل ارتباط بین شبکه‌ای به نام TCP/IP اقدام نمودند. مفهوم نامی این عنوان عبارت بود از: پروتکل کنترل انتقال و پروتکل ارتباط بین شبکه. اگرچه در آغاز آن‌ها این مهم را در قالب تنها یک پروتکل تعریف نمودند اما بعدها این پروتکل خود به دو بخش مجزای TCP و IP تقسیم شد که هر یک از آن‌ها به گونه‌ی مستقل عمل می‌نمود. این دو پژوهشگر نتیجه‌ی حاصل از این طراحی را در قالب مقاله‌ای پژوهشی در ماه مه 1974 در مجله‌ی فناوری ارتباطات انجمن مهندسین برق و الکترونیک (IEEE) منتشر نمودند. پروتکل TCP/IP که امروز آن را پوست و بافت اینترنت می‌نامند قبل از پیدایش و عرضه‌ی رایانه‌های شخصی (PC) و ترمینال‌های کاری امروزی و نیز قبل از گسترش شبکه‌های اترنت و دیگر فناوری‌های شبکه‌های محلی (LAN) و از همه مهم‌تر قبل از وب و نیز همه‌ی کاربردهای صوتی و تصویری و متنی پدید آمد. این دو پژوهشگر فرهیخته به نیازهای یک پروتکل شبکه که از یک سو قادر به پشتیبانی کاربردهای وسیعی که در آینده بایستی تعریف می‌گردید پی بردند و از سوی دیگر لزوم برقراری ارتباط بین پروتکل‌های لایه‌ی پیوند و میزبان‌های خود مختار را شناسایی نمودند.

در سال 2004 میلادی به این دو پژوهشگر به پاس تلاش آن‌ها برای چنین ثمره‌ای نشان آلن تورینگ که معادل جایزه‌ی نوبل می‌باشد، اهداء گردید.



شکل ۳-۲۸ ♦ بافرهای ارسال و دریافت TCP

• ساختار قطعه‌ی TCP

با توجه به موارد اشاره رفته برای ارتباط TCP اکنون ساختار قطعه در TCP بررسی می‌شود. هر قطعه ی پروتکل TCP شامل چندین میدان سرپیام و یک محدوده‌ی داده می‌باشد. دامنه‌ی داده شامل بخشی از تمامی داده‌ی کاربردی است. قبلاً اشاره گردید حداکثر طول میدان داده در قطعه از طریق MSS محدود می‌شود. هنگامی‌که یک فایل نسبتاً پر حجم مانند یک تصویر به‌عنوان بخشی از یک صفحه وب از سوی TCP در حال ارسال است. فایل موردنظر به قطعه‌های کوچکتری تقسیم می‌شود که هر کدام از آن‌ها به جز قطعه‌ی آخر از لحاظ اندازه طولی برابر اندازه‌ی MSS است. معمولاً همیشه قطعه‌ی آخر کوچکتر از این اندازه می‌باشد. البته کاربردهای محاوره‌ای در اغلب موارد از قطعه داده‌هایی استفاده می‌کنند که کوچکتر از MSS می‌باشند. به‌عنوان مثال در یک کاربرد مانند تِلنِت (Telnet) اندازه‌ی محدوده‌ی داده در قطعه‌ی TCP معادل یک بایت است. از آنجائیکه طول سرپیام TCP معادل 20 بایت می‌باشد و این طول به اندازه‌ی 12 بایت بیشتر از سرپیام UDP است آنگاه قطعه‌ی ارسالی از طریق تلنت طولی معادل 21 بایت را دارد. شکل ۳-۲۹ ساختار قطعه‌ی TCP را نشان می‌دهد. همانند

قطعه‌ی UDP سرپیام شامل دامنه‌های **شماره پورت مبدا و مقصد** می‌باشد که عملیات تسهیم داده را از سوی لایه‌ی بالاتر و یا به طرف آن کنترل می‌کند. علاوه‌بر آن مانند UDP سرپیام قطعه ی TCP شامل یک **میدان کُد مجموع مقابله** می‌باشد. سرپیام یک قطعه‌ی TCP شامل موارد ذیل می‌گردد:

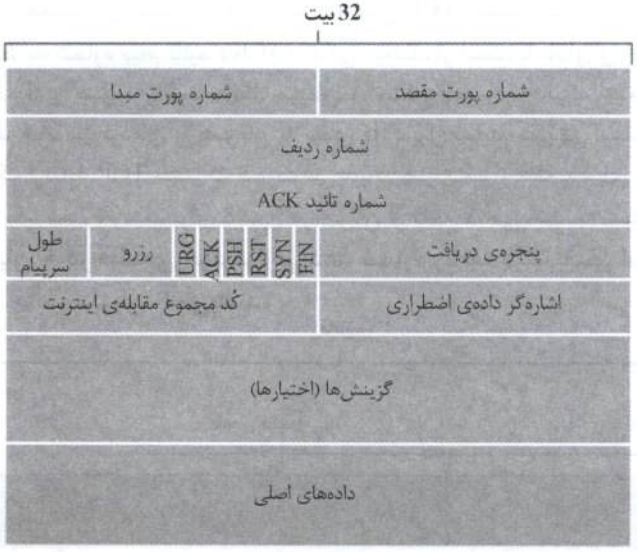
۱- **دامنه‌ی شماره ردیف** با طول 32 بیت و نیز **دامنه‌ی شماره تائید** (ACK) که فرستنده و گیرنده‌ی TCP از آن جهت پیاده‌سازی مطمئن انتقال داده به گونه‌ای که در ذیل آمده است بهره می‌گیرند.

۲- **دامنه‌ی پنجره دریافت** با طول 16 بیت که از آن برای کنترل جریان استفاده می‌شود. این دامنه تعداد بایت‌های قابل قبول جهت دریافت را از سوی یک گیرنده مشخص می‌کند.

۳- **دامنه‌ی طول سرپیام** با بزرگی 4 بیت. این محدوده طول سرپیام TCP را در یک کلمه‌ی 32 بیت مشخص می‌کند. سرپیام TCP می‌تواند طول متغیری را به دلیل وجود محدوده‌ی قابل گزینش در TCP اختیار نماید. به‌عنوان مثال هنگامی‌که **محدوده‌ی قابل گزینش**[1] در سرپیام تهی باشد، سرپیام TCP معادل 20 بایت می‌شود.

۴- **دامنه‌ی قابل گزینش** (Options) با طول متغیر و اختیاری هنگامی مورد استفاده قرار می‌گیرد که فرستنده و گیرنده به تعیین حداکثر اندازه‌ی قطعه‌ی (MSS) و یا مقیاس پنجره برای کاربرد شبکه‌های پرسرعت می‌پردازند.

۵- **دامنه‌ی نشان یا پرچم** شامل 6 بیت می‌باشد. از بیت ACK جهت اعتبارسنجی ارزش حمل شده در محدوده‌ی تائید استفاده می‌شود. به عبارت دیگر قطعه شامل یک پیام تائید برای قطعه‌ی دیگری است که دریافت آن موفق بوده است. بیت‌های RST ، SYN و FIN جهت برقراری و نیز قطع ارتباط مورد استفاده قرار می‌گیرند.



شکل ۳-۲۹ ♦ ساختار قطعه‌ی TCP

هنگامی‌که بیت PSH برابر یک می‌شود گیرنده را مجبور به تحویل بی‌درنگ داده به لایه‌ی بالاتر

1- Option field

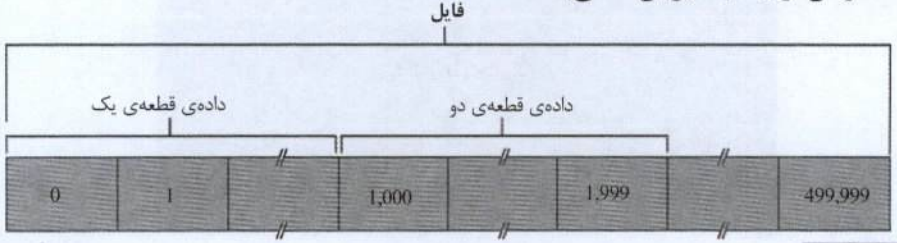
می‌کند. سرانجام بیت URG است که از آن برای مشخص نمودن داده‌ای استفاده می‌شود که موجودیتی در لایه‌ی بالاتر در سمت ارسال، به آن برچسب **فوری** اختصاص داده است. آدرس و یا مکان آخرین بایت این داده‌ی اضطراری از سوی **دامنه‌ی اشاره‌گر داده‌ی اضطراری**[1] (UDPF) که 16 بیت می‌باشد، مشخص می‌گردد پروتکل TCP بایستی موجودیت لایه‌ی بالاتر گیرنده را به هنگام وجود داده‌ی اضطراری آگاه نموده و به آن یک اشاره‌گر که انتهای این داده را مشخص می‌کند، اختصاص دهد. از بیت‌های PDH و URG و اشاره‌گر اضطراری داده عملاً در شبکه استفاده نمی‌شود.

شماره‌های ردیف (SN) و تائید (ACK)

دو دامنه‌ی بسیار مهم در سر پیام قطعه ی TCP عبارتند از شماره ردیف و شماره تائید. این میدان‌ها از نقطه‌نظر انجام انتقال مطمئن داده در TCP بسیار حیاتی هستند. داده‌های دریافتی از نقطه‌نظر TCP ساخت یافته نیستند اما جریانی از بایت‌ها به گونه‌ی نظم یافته قلمداد می‌شوند. استفاده‌ی TCP از شماره‌های ردیف جهت انعکاس این نقطه‌نظر است که آن‌ها بر روی جریان بایت‌ها حرکت می‌کنند و نه بر روی گروهی از قطعه‌های انتقال یافته. بنابراین شماره‌ی ردیف برای یک قطعه یک شماره‌ی جریان بایت از نخستین بایت در قطعه می‌باشد. جهت شفاف شدن موضوع به مثال ذیل توجه کنید.

اگر فرض شود فرآیندی در میزبان A مایل به ارسال جریانی از داده به فرآیند دیگری در میزبان B از طریق یک ارتباط TCP باشد. در میزبان A پروتکل TCP هر یک از بایت‌ها را در جریان داده‌ها شماره‌گذاری می‌کند. اگر فرض شود جریان داده شامل یک فایل 500,000 بایتی باشد و اندازه MSS در آن نیز 1000 بایت و شماره‌گذاری نخستین بایت از جریان داده نیز با صفر آغاز گردد. بر اساس شکل ۳-۳۰ پروتکل TCP تعداد 500 قطعه را از درون جریان داده می‌سازد. به نخستین قطعه شماره ردیف صفر اختصاص می‌یابد و قطعه‌ی دوم شماره ردیف 1000 و قطعه سوم 2000 و غیره را اختیار می‌کنند. هر یک از آن‌ها در محدوده‌ی شماره ردیف سرپیام قطعه‌ی TCP مناسب خود وارد می‌شوند.

موضوع مربوط به شماره پیام تائید (ACK) اندکی پیچیده‌تر است. با یادآوری این نکته که TCP یک پروتکل دو طرفه‌ی همزمان است ، آنگاه میزبان A ممکن است در حالیکه مشغول دریافت داده‌ای از میزبان B باشد خود نیز داده‌ای را همزمان به سوی B گسیل نماید. بدیهی است این مبادله‌ی دو طرفه در قالب یک ارتباط TCP مشترک صورت می‌پذیرد. هر یک از قطعه‌های وارد شده از جانب میزبان B حامل یک شماره ردیف برای داده‌ی در جریان از B به A می‌باشد. به عبارت دیگر **شماره تائیدی که میزبان A در قطعه‌ی خود قرارمی‌دهد شماره ردیف بایت بعدی است که میزبان A در انتظار آن از جانب میزبان B می‌باشد.**



شکل ۳-۳۰ ♦ تقسیم‌بندی فایل داده در قطعه‌های TCP

1- Urgent data pointer field

به منظور شفاف‌سازی موضوع به مثال‌هایی که به دنبال می‌آید دقت نمائید.

فرض کنید میزبان A همه‌ی بایت‌های از شماره‌ی صفر تا 535 را از میزبان B دریافت نموده است. اکنون فرض کنید A قصد ارسال قطعه‌ای را به B دارد. میزبان A منتظر دریافت بایت 536 و تمامی بایت‌های به دنبال آن در جریان داده‌ی میزبان B می‌باشد. بنابراین میزبان A شماره تائید 536 را در میدان قطعه وارد می‌نماید و سپس آن را به سوی B گسیل می‌نماید.

در مثال دیگر اگر فرض شود میزبان A قطعه‌ای را که شامل بایت‌های از صفر تا 535 است در کنار قطعه‌ی دیگری که شامل بایت‌های 900 تا 1000 است دریافت نماید. اگر بنا بر دلایلی میزبان A هنوز موفق به دریافت بایت‌های از 536 تا 899 نشده باشد بنابراین میزبان A هنوز در انتظار بایت 536 و دنباله‌ی آن جهت بازیابی مجدد جریان داده‌ی B می‌باشد. بنابراین قطعه‌ی بعدی A به B در میدان شماره تائید خود حامل ارزش 536 می‌باشد زیرا TCP فقط تا نخستین بایت مفقود شده در جریان را تائید می‌کند، به عبارت دیگر گفته می‌شود TCP **تائیدهای انباشتی** را فراهم می‌آورد.

مثال دیگری که در این زمینه از اهمیت ویژه‌ای برخوردار است به‌صورت ذیل خلاصه می‌شود:

اگر فرض شود میزبان A قطعه‌ی سوم که شامل بایت‌های از 900 تا 1000 است را قبل از دریافت قطعه‌ی دوم که شامل بایت‌های از 536 تا 899 است دریافت نموده باشد آنگاه قطعه‌ی سوم خارج از نوبت دریافت شده است. موضوع مهم آن است که یک میزبان به هنگام دریافت قطعه‌های خارج از نوبت در یک ارتباط TCP چه عملی را انجام می‌دهد؟ پاسخ را نمی‌توان در اسناد RFC جستجو نمود زیرا این اسناد انجام این مهم را بر عهده‌ی طراحان برنامه‌های پیاده‌سازی TCP گذارده‌اند. اصولاً از دو رویکرد می‌توان بهره گرفت. نخست آنکه گیرنده بی‌درنگ از قطعه‌های خارج از نوبت صرف‌نظر نماید. این موضوع سبب ساده شدن طراحی گیرنده می‌شود و دوم آنکه گیرنده بایت‌های خارج از نوبت را نگهداری می‌کند و در انتظار بایت‌های مفقود شده جهت تکمیل مکان‌های خالی آن‌ها می‌شود. بدیهی است رویکرد دوم از نقطه‌نظر پهنای باند شبکه به مراتب از کارآمدی بیشتری برخوردار است و بنابراین رویکردی است که عملاً از آن استفاده می‌شود. در شکل ۳-۳۰ فرض بر آن بود که شماره ردیف در ابتدا صفر است در صورتی‌که در عمل هر دو طرف ارتباط TCP به گونه‌ی تصادفی یک شماره ردیف اولیه را گزینش می‌نمایند. این عمل از آن جهت صورت می‌گیرد که یک قطعه‌ی حاضر در شبکه از نتیجه‌ی یک ارتباط پایان گرفته‌ی بین دو میزبان حاصل شده است که با قطعه‌ی معتبر دیگری بین همان دو میزبان در ارتباط بعدی اشتباه گرفته نشود. بدیهی است هر دو ارتباط ممکن است از یک شماره پورت مشابه بهره گرفته باشند.

۳-۱۲ تِلنت[1] موضوع مطالعه شماره‌های ردیف و تائید

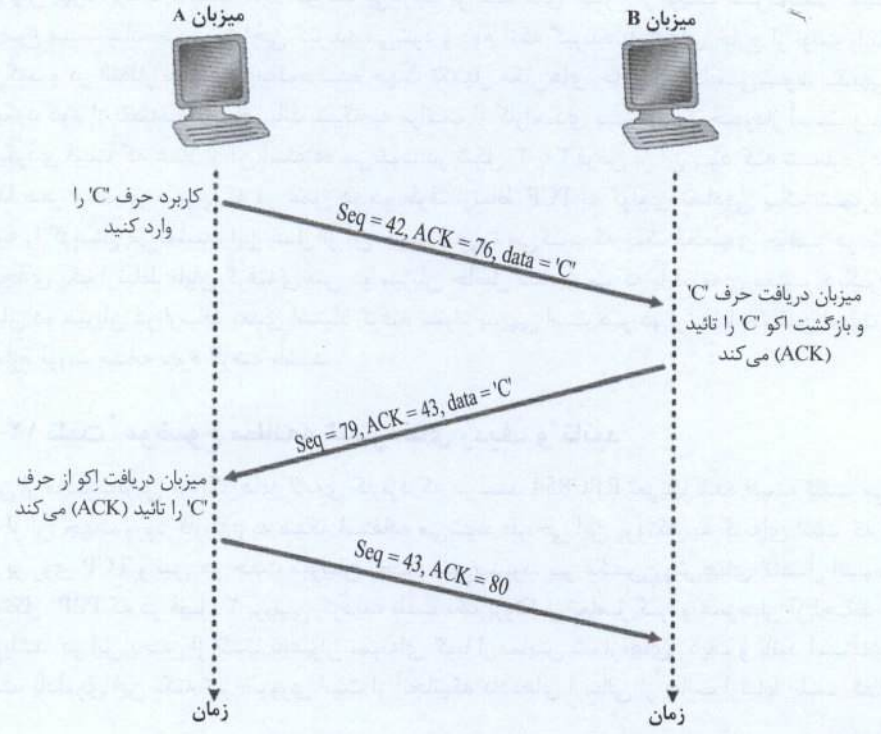
یکی از محبوب‌ترین پروتکل‌های لایه‌ی کاربرد که در سند RFC854 تعریف شده است، تلنت می‌باشد. که از آن جهت ورود کاربران به شبکه استفاده می‌شود. طراحی این پروتکل به گونه‌ای است که اجرای آن بر روی TCP و بین هر جفت میزبان انجام می‌پذیرد. بر عکس روش‌های انتقال انبوه مانند پروتکل P2P که در فصل ۲ بررسی گردید، تلنت یک پروتکل تعامل‌گرا و همزمان از لحاظ کاربرد می‌باشد. در این بخش از تلنت به‌عنوان نمونه‌ای گویا از نمایش شماره‌های ردیف و تائید استفاده شده است. یادآوری این نکته نیز ضروری است از آنجائیکه داده‌های ارسالی از جانب ارتباط تلنت که کلمه‌ی

1- Telnet

عبور را نیز در درون حمل می‌کند رمز نمی‌شود و بنابراین یک سوراخ امنیتی را در این پروتکل سبب می‌گردد بسیاری از کاربران ترجیح می‌دهند از پروتکل دیگری به نام **SSH** استفاده نمایند. این موضوع در فصل ۸ بیشتر بررسی شده است.

اگر فرض شود میزبان A آغازگر یک تعامل تلنت با میزبان B باشد. بنابراین میزبان A مشتری و میزبان B سرور قلمداد می‌شود. در این تعامل هر کاراکتری را که میزبان A (مشتری) گسیل نماید، میزبان B نسخه‌ای (کپی) از آن کارکتر را به سوی کاربر تلنت (مشتری) بازگشت می‌دهد. از این روش بازگشت اِکُو به این سبب استفاده می‌شود تا اطمینان حاصل گردد که کاراکتر از سوی سرور B دریافت و پردازش شده است. به عبارت دیگر هر داده ی ارسالی دوباره مسیر بین کاربرانی نظیر A و B را در شبکه پیمایش می‌کند.

اکنون اگر فرض شود کاربر کاراکتر C را وارد نماید و آنگاه مشغول وظیفه‌ی دیگری گردد آنگاه تبادل قطعه‌های TCP بین مشتری و سرور را می‌توان بر اساس شکل ۳-۳۱ تجسّم نمود. اگر شماره ردیف‌های اولیه به ترتیب 42 و 79 برای مشتری و سرور باشند. قبلاً اشاره گردید شماره ردیف یک قطعه ی TCP در حقیقت شماره ردیف نخستین بایت در محدوده‌ی داده در قطعه می‌باشد.بنابراین نخستین قطعه‌ی ارسالی از سوی مشتری دارای شماره ردیف 42 می‌باشد و نخستین قطعه بازگشتی از سوی سرور نیز با شماره ردیف 79 مشخص می‌گردد. هم‌چنین شماره تائید (ACK num) عبارت است از شماره ردیف بایت بعدی که میزبانی در انتظار آن می‌باشد. پس از برقراری ارتباط TCP اما قبل از ارسال هر داده‌ای، میزبان مشتری منتظر بایت 79 می‌باشد و سرور نیز در انتظار بایت 42 به سر می‌برد.



شکل ۳-۳۱ ♦ شماره ردیف تائید برای یک کاربرد تلنت ساده بر روی TCP

شکل ۳-۳۱ نشان می‌دهد تعداد سه قطعه گسیل شده است. نخستین قطعه از مشتری به سرور شامل یک بایت ASCII به نمایندگی حرف C در محدوده‌ی داده و دارای شماره ردیف 42 در میدان مربوط به آن می‌باشد. از آنجاییکه مشتری هنوز داده‌ای را از سوی سرور دریافت ننموده است این نخستین قطعه در دامنه‌ی شماره تائید خود حامل ارزش 79 است. هنگامی‌که قطعه ی دوم از سوی سرور به مشتری گسیل می‌شود، این قطعه دو هدف را دنبال می‌کند. نخست تائیدی را از داده‌ی دریافتی به‌وسیله سرور فراهم می‌آورد و این هدف را از طریق قرار دادن ارزش 43 در دامنه‌ی تائید انجام می‌دهد. پیام این عمل عبارت است از آگاه شدن مشتری از سوی سرور از این واقعیت که همه‌ی داده‌های ارسالی با موفقیت از ابتدا تا بایت شماره‌ی 42 از سوی سرور دریافت شده است و بنابراین سرور در انتظار بایت شماره 43 به بعد می‌باشد. هدف دوم از گسیل این قطعه پاسخ به‌صورت پیام اِکو برای داده‌ی C می‌باشد. بنابراین دومین قطعه حامل معادل ASCII کارکترC نیز در محدوده‌ی داده‌ی خود می‌باشد. بدیهی است این قطعه در برگیرنده‌ی شماره ردیف 79 می‌باشد. این شماره ردیف اولیه‌ی جریان داده بین مشتری و سرور است زیرا در این ارتباط TCP نخستین بایتی است که سرور ارسال نموده است. بدیهی است داده‌ی تائید مشتری به سرور از سوی قطعه‌ای حمل می‌شود که حامل داده‌ی سرور به مشتری است و بنابراین گفته می‌شود پیام تائید بر **پشت**[1] قطعه‌ی سرور و مشتری نشسته است.

سرانجام ارسال قطعه‌ی سوم از سوی مشتری به سرور صورت می‌پذیرد. هدف اصلی این قطعه تائید داده‌ی دریافتی از سوی سرور به‌وسیله‌ی مشتری می‌باشد. در این قطعه میدان داده تهی است به عبارت دیگر پیام تائید بر پشت داده‌ی مشتری به سرور حمل نشده است. قطعه دارای ارزش 80 در میدان شماره تائید می‌باشد زیرا برنامه‌ی مشتری جریانی از بایت‌ها را از ابتدا تا بایت با شماره ردیف 79 دریافت نموده و در حال حاضر در انتظار بایت‌های 80 به بعد می‌باشد. بدیهی است با وجود آنکه قطعه حامل داده‌ای نمی‌باشد اما دارای یک شماره ردیف است علت آن وجود دامنه‌ی شماره ردیف در یک قطعه‌ی TCP است و بنابراین بایستی حامل داده‌ای باشد.

۳-۱۳ تخمین زمان‌های خروج و دوره‌ی رفت و برگشت[2]

پروتکل TCP نیز همانند پروتکل rdt از مکانیزم زمان خروج و انتقال مجدد جهت بازیابی قطعه‌های مفقودی بهره می‌جوید. پیاده‌سازی جزئیات این عملیات به ظاهر ساده در پروتکلی مانند TCP نیازمند مهارت و دقت قابل توجهی می‌باشد. یکی از مهم‌ترین موارد در این پیاده‌سازی تناوب و طول زمان خروج می‌باشد. معمولاً زمان خروج همواره بایستی بزرگتر از زمان دوره‌ی رفت و برگشت ارتباط RTT باشد. قبلاً اشاره گردید این زمان هنگامی است که قطعه‌ای ارسال و سپس تائید آن بازگشت داده می‌شود. پرسش‌های مهم مقدار بزرگی این زمان، تخمین اولیه‌ی RTT ، ارتباط یک زمان‌سنج با هر یک از قطعه‌های تائید نشده و موارد دیگر می‌باشد. این بخش به بررسی این موضوعات در ارتباط با سند جاری RFC2988 می‌پردازد.

الف) تخمین زمان دوره‌ی رفت و برگشت RTT

ابتدا به بررسی چگونگی انجام تخمین زمان دوره‌ی رفت و برگشت در TCP بین فرستنده و گیرنده می‌پردازیم. با تعریف یک نام متغیر RTT به‌صورت Sample RTT برای طول زمان آغاز می‌کنیم. عبارت

1- Piggy backed
2- Round trip and time out

Sample RTT . برای یک قطعه مقدار زمانی است که طول می‌کشد تا قطعه تحویل IP گردد و تائید دریافت آن وصول شود. به جای اندازه‌گیری این متغیر برای هر قطعه‌ی ارسال شده، بسیاری از پیاده‌سازی‌های TCP تنها اقدام به اندازه‌گیری Sample RTT فقط برای یکی از قطعه‌های گسیل شده‌ای که هنوز تائید آن دریافت نگردیده است، می‌کنند. بنابراین در فاصله هر RTT تقریباً ارزشی برای Sample RTT بدست می‌آید. پروتکل TCP این ارزش را برای قطعه‌ای که مجدداً انتقال یافته است اندازه‌گیری نمی‌کند.

بدیهی است با این روش اندازه‌گیری، ارزش‌های Sample RTT از قطعه‌ای به قطعه‌ی دیگر در نوسان است. علت این موضوع به سبب وجود تراکم در مسیریاب‌ها و نیز بارهای متغیر در سیستم‌های انتهایی است. برای تخمین یک RTT طبیعی است که میانگین ارزش‌های Sample RTT بایستی محاسبه گردد. بنابراین پروتکل TCP همواره از یک ارزش میانگین بهره‌برداری می‌کند که اصطلاحاً Estimated RTT (تخمین RTT برای ارزش‌های نمونه‌های RTT) نام گرفته است. هنگامی‌که یک ارزش جدید برای Sample RTT بدست می‌آید، پروتکل TCP مقدار تخمین RTT را بر اساس رابطه ذیل به روز رسانی می‌کند:

$$\text{Estimated RTT} = (1-\alpha).\ \text{EstimatedRTT} + \alpha .\ \text{SampleRTT} \qquad (۴-۳)$$

رابطه‌ی اشاره شده در قالب دستورالعمل‌های یک زبان برنامه‌نویسی مانند C++ تعریف می‌شود. این مقدار جدید تخمین RTT بر مبنای ارزش‌های قبلی آن و نیز ویرایش جدید نمونه‌ی RTT توزین می‌شود. ارزش قرار دادی برای α برابر 0.125 یا $\frac{1}{8}$ می‌باشد. با توجه به این ارزش رابطه‌ی اشاره شده در بالا به گونه‌ی ذیل تعریف می‌شود:

$$\text{EstimatedRTT} = 0.875\text{EstimatedRTT} + 0.125\ \text{SampleRTT} \qquad (۵-۳)$$

بدیهی است ارزش EstimatedRTT **میانگین توزینی**[1] ارزش‌های SampleRTT است. ارزش میانگین توزینی سبب می‌شود وزن بیشتری بر روی نمونه‌های جدیدتر در مقایسه با نوع قدیمی‌تر آن‌ها گذارده شود. این نکته یک امر طبیعی است زیرا نمونه‌های جدیدتر انعکاس بهینه‌ای از تراکم جاری در شبکه می‌باشند. در مطالعات آماری چنین میانگین‌هایی اصطلاحاً **میانگین متحرک توزین تصاعدی**[2] (EWMA) نامگذاری می‌شوند. واژه‌ی تصاعدی در EWMA به دلیل آن است که هر نمونه‌ی RTT به هنگام انجام مراحل به روزرسانی با شتاب میرا می‌شود.

شکل ۳-۳۲ ارزش‌های SampleRTT و EstimatedRTT را برای مقدار $\alpha = \frac{1}{8}$ در یک ارتباط TCP بین دانشگاه ماساچوست آمریکا (gaia.cs.umass.edu) و نهادی آموزشی در جنوب فرانسه (fantasia.eyrecom.fr) نشان می‌دهد. بدیهی است تغییرات SampleRTT در محاسبه برای تخمین RTT یعنی EstimatedRTT هموار شده است.

علاوه بر انجام محاسبه برای تخمین RTT اگر تغییرات RTT نیز بر اساس پیشنهاد سند RFC2988 محاسبه گردد، سودمندی بیشتری را در پی خواهد داشت. در این سند این تغییرات اصطلاحاً DevRTT نامیده شده است و به مفهوم مقدار تخمین انحراف SampleRTT از EstimatedRTT می‌باشد. این موضوع در رابطه‌ی ۳-۶ نشان داده شده است.

1- Weighted Average
2- Exponential Weighted Moving Average

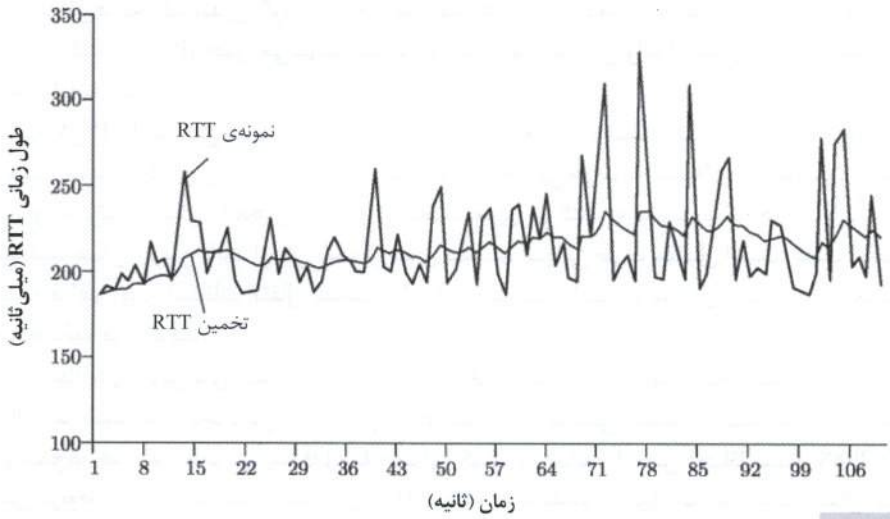
$$DevRTT = (1-\beta).\ DevRTT + \beta\ [SampleRTT - EstimattedRTT] \qquad (۶-۳)$$

لازم به توضیح است که DevRTT در یک EWMA عبارت است از تفاوت بین نمونه‌ی RTT و اندازه‌ی تخمین RTT . اگر ارزش‌های SampleRTT دچار تعداد تغییرات اندکی باشند آنگاه DevRTT بسیار ناچیز است. از طرف دیگر اگر میزان تغییرات انبوه باشد اندازه‌ی DevRTT نیز قابل توجه است. مقدار پیشنهادی β برابر 0.25 می‌باشد.

ب) تعریف و مدیریت تناوب زمان خروج در انتقال مجدد

با توجه به ارزش‌های EstimatedRTT و DevRTT از چه ارزشی پروتکل TCP بایستی برای تناوب زمان خروج استفاده کند؟ بدیهی است این تناوب بایستی بزرگتر و یا برابر EstimatedRTT باشد در غیر این صورت گسیل انتقال مجدد که نتیجه‌ی آن بی‌حاصل است، انجام می‌شود. تناوب زمان خروج نبایستی بسیار بزرگ‌تر از EstimatedRTT باشد زیرا هنگامی‌که یک قطعه مفقود می‌شود پروتکل TCP نمی‌تواند به سرعت نسبت به انتقال مجدد قطعه اقدام نماید. این خود سبب می‌شود تاخیر طولانی در انتقال داده‌ها روی دهد. بنابراین مطلوب آن است که زمان خروج معادل EstimatedRTT و اندکی بزرگتر از آن باشد. این مقدار افزایش اندک به دلیل احتمال وجود نوسانات زیاد در ارزش‌های SampleRTT است. ارزش DevRTT نیز در تاثیر این محاسبات مؤثر است. با در نظر گرفتن تمامی این موارد زمان خروج انتقال مجدد در TCP از طریق رابطه‌ی ذیل تعریف می‌شود:

$$TimeoutInterval = EstimatedRTT + 4\ DevRTT \qquad (۴-۳)$$



شکل ۳-۳۲ ♦ تخمین و نمونه‌های RTT

پروتکل TCP انتقال مطمئن داده را از طریق پیام‌های تائید مثبت و زمان شمارهای مشابه آنچه در بخش ۳-۶ توضیح داده شده است، انجام می‌دهد. در این حالت پروتکل TCP پیام تائید داده‌ای که سالم دریافت شده است را گسیل و نسبت به انتقال مجدد قطعه‌هایی که تصور می‌رود مفقود و یا آلوده شده‌اند اقدام می‌نماید. بعضی از پیاده‌سازی‌های TCP از مکانیزم‌های انحصاری NAK نیز بهره می‌برند که در آن‌ها عملیات انتقال مجدد با سرعت انجام می‌شود. دریافت سه نسخه‌ی متوالی از پیام تائید (ACK) برای یک قطعه به مفهوم یک NAK برای قطعه‌ی بعدی بوده و در نتیجه سبب تحریک

عملیات انتقال مجدد قبل از وقوع زمان خروج است. پروتکل TCP از زنجیره‌ای از اعداد برای سهولت در عملیات شناسایی قطعه‌های مفقود شده و یا تکراری استفاده می‌کند. مانند پروتکل انتقال rdt 3.0 پروتکل TCP قادر به تشخیص آنکه یک قطعه و یا ACK وابسته به آن مفقود، آلوده و یا دچار تاخیر بیش از حد معمول شده‌اند، نمی‌باشد. در فرستنده واکنش TCP انتقال مجدد قطعه‌ای است که دچار چنین وضعیتی شده است.

پروتکل TCP علاوه بر موارد بالا از خط لوله نیز بهره می‌جوید. این کاربرد به فرستنده اجازه می‌دهد به انجام چندین انتقال همزمان قبل از دریافت پیام‌های تائید قطعه‌های ارسال شده بپردازد. قبلاً ملاحظه گردید خط لوله می‌تواند سبب بهبود و افزایش بازدهی در مواقعی گردد که نسبت اندازه‌ی قطعه به RTT کوچک است. تعداد قطعه‌های تائید نشده‌ای که یک فرستنده می‌تواند در اختیار داشته باشد تعیین کننده‌ی نوع مکانیزم کنترل جریان و تراکم در TCP است.

۳-۱۴ انتقال مطمئن داده

قبلاً اشاره شد انتقال در لایه شبکه‌ی اینترنت (IP) غیر مطمئن است واین لایه هیچگونه ضمانتی را در قبال داده گرام‌ها از لحاظ تحویل سالم، رعایت ترتیب در تحویل و یا حفظ یکپارچگی آن‌ها فراهم نمی‌آورد. در پروتکل IP داده گرام‌ها می توانند سبب سرریزی بافرها در مسیریاب‌ها شده و در نتیجه‌ی آن داده گرام‌ها ممکن است هرگز به مقصد خود نرسند و یا آنکه خارج از نوبت ارسال، به مقصد وارد گردند. علاوه بر این موارد بیت‌ها در داده گرام‌ها ممکن است دچار خطا شده و ارزش یک آن‌ها به صفر و یا صفر آن‌ها به یک تبدیل گردد. از آنجائیکه قطعه‌های داده متعلق به لایه انتقال TCP در فراسوی شبکه به کمک لایه IP حمل می‌شوند بنابراین لایه‌ی انتقال نیز می‌تواند از سوی این نارسایی‌ها به چالش گرفته شود.

پروتکل TCP به خلق سرویس مطمئن انتقال داده بر فراز لایه انتقال غیر مطمئن IP می‌پردازد. این سرویس جریان داده‌ای را که یک فرآیند از بافر TCP خود می‌خواند به گونه‌ای سالم، بدون تناوب و تکرار و به ترتیب ارسال داده‌های آن تحویل مقصد بعدی می‌کند. به عبارت دیگر جریان بایت تحویل شده عیناً همان جریان بایت ارسال شده از سوی سیستم انتهایی طرف مقابل است. علت آنکه TCP قادر به فراهم آوری امکانات انتقال مطمئن داده است به سبب اصول و مواردی است که در بخش ۳-۶ مورد مطالعه قرار گرفت.

در مطالعه‌ی روش‌های انتقال مطمئن داده قبلاً فرض شد که یک زمان‌سنج وابسته به هر قطعه‌ی ارسالی اما تائید نشده وجود دارد. اگرچه از لحاظ نظری این موضوع بسیار جالب است اما مدیریت زمان‌سنج‌ها خود به ایجاد سرآیند قابل ملاحظه‌ای کمک می‌نماید . از این لحاظ سند RFC 2988 در بخش مربوط به مدیریت زمان شمارهای TCP توصیه به استفاده از تنها یک زمان‌سنج انتقال می‌کند حتی اگر چندین قطعه‌ی انتقال یافته‌ی همزمان اما تائید نشده وجود دارد. پروتکل TCP که در ادامه تشریح می‌شود از چنین ویژگی برخوردار است. آنچه در ادامه آمده است پیاده‌سازی انتقال مطمئن داده از سوی TCP که فقط از زمان خروج جهت بازیابی قطعه‌های مفقود شده استفاده می‌کند و پس از آن توصیف کاملی از روشی که از تکرار پیام تائید علاوه بر زمان خروج بهره می‌جوید، ارایه می‌گردد. در این راستا فرض بر آن است که داده‌ها تنها در یک جهت ارسال می‌شوند یعنی از میزبان A به B و فایل ارسالی نیز دارای حجم قابل توجهی است.

شکل ۳-۳۳ توصیف بسیار ساده‌ای از یک فرستنده‌ی TCP است. در این شکل ملاحظه می‌شود که

سه رخداد مهم در ارتباط با انتقال و ارسال مجدد داده در فرستنده‌ی TCP وجود دارد. این سه عبارتند از: داده‌ی دریافتی از کاربرد بالاتر، زمان‌شمار مربوط به محاسبه‌ی زمان خروج و دریافت پیام تائید ACK . با وقوع نخستین رویداد مهم پروتکل TCP داده‌ای را از کاربردی تحویل، کپسول‌بندی در قالب قطعه و تحویل آن به IP می‌کند. هر قطعه شامل یک شماره ردیف که حامل شماره بایت جریان برای نخستین بایت داده در قطعه است، می‌باشد. چنانچه زمان شمار برای قطعه‌های دیگر فعال نمی‌باشد، پروتکل TCP هنگامی‌که قطعه را به سمت IP عبور می‌دهد، آن‌ها را فعال می‌کند. زمان‌سنج را می‌توان وابسته به قدیمی‌ترین قطعه‌ی تائید نشده دانست. زمان پایان اعتبار این زمان شمار عبارت Timeout Interval می‌باشد که از طریق روابط (۳-۲) و (۳-۳) که قبلاً توصیح داده شده است محاسبه می‌شود.

رخداد مهم دوم زمان خروج است. پروتکل TCP پاسخ این رخداد را از طریق انتقال مجدد قطعه‌ای که سبب این رویداد شده است، فراهم می‌آورد و سپس زمان شمار را فعال می‌سازد. رویداد سوم که بایستی از سوی فرستنده‌ی TCP مدیریت شود عبارت است از ورود پیام تائید قطعه ACK . بدیهی است این رخداد از سوی قطعه‌ای می‌باشد که ارزش دامنه‌ی ACK در آن معتبر قلمداد شده است. به هنگام وقوع این رویداد، TCP ارزش y از ACK را با ارزش متغیر Send Base مقایسه می‌کند. متغیر حالت TCP یعنی Send Base شماره ردیف آخرین بایت شناخته شده‌ای است که سالم و با رعایت نظم از سوی گیرنده دریافت شده است. قبلاً اشاره شد TCP از تائید انباشتی استفاده می‌کند به گونه‌ای که ارزش y تائید کننده‌ی همه‌ی بایت‌های دریافتی قبل از بایت شماره y می‌باشد. چنانچه ارزش y>SendBase باشد آنگاه ACK به تائید یک یا چندین قطعه‌ی تائید نشده‌ی قبلی می‌پردازد. بنابراین فرستنده متغیر Send Base را به روزرسانی نموده و زمان شمار را مجدداً در صورتی‌که قطعه‌های هنوز تائید نشده‌ای وجود دارند، فعال می‌کند.

```
/* Assume sender is not constrained by TCP flow or congesting
control, that data from above is less than Mss in size , and that
data transfer is in one direction only.*/

NextSeqnum=InitalSeqnumber
SendBase=InitalSeqnumber

loop (forever) {
    Switch(event)

        event: data received from application above
            create TCP segment with sequence number NextSeqnum
            if (timer currently not running)
                start timer
            pass segment to IP
            NextSeqNum=NextSeqnum+length (data)
            break;

        event: timer timeout
            retransmit not-yet-acknowledged segment with
                smallest sequence number
            start timer
            break;

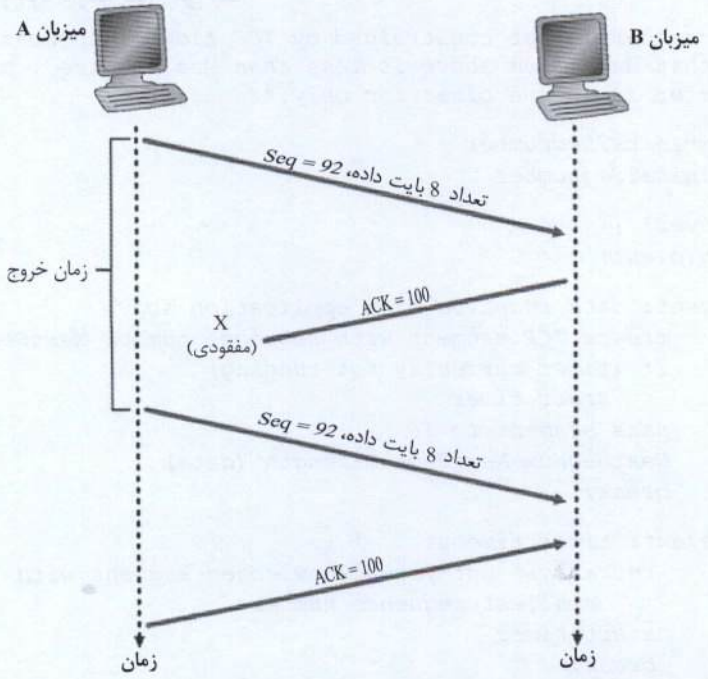
        event: Ack received, with ACK field value of y
```

```
            if (y > SendBase) {
                SendBase=y
                if (there are currently any not-yet-acknowleged segments)
                    start timer
                }
            break;
    } /* end of loop forever */
```

شکل ۳-۳۳ ♦ فرستنده‌ی ساده شده‌ی TCP

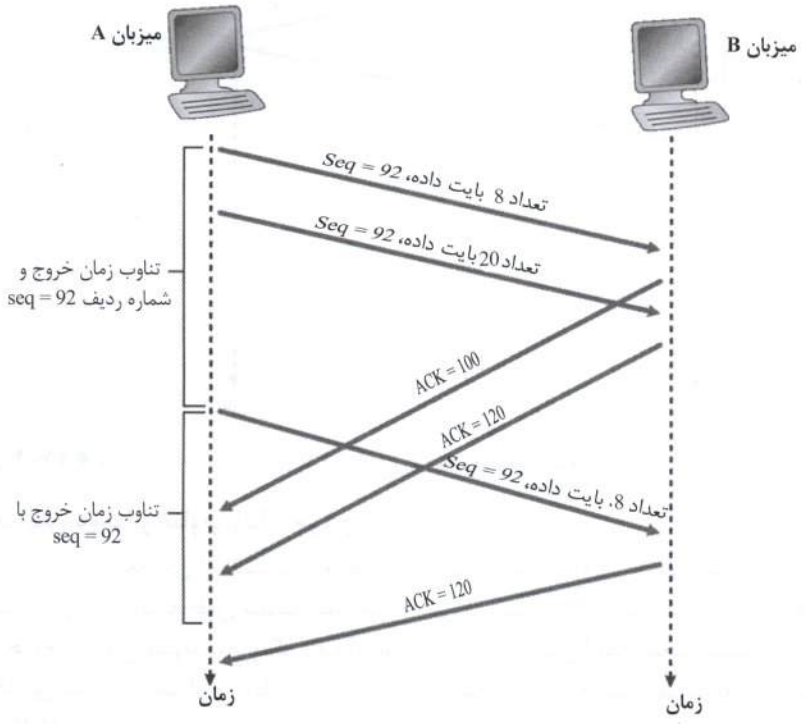
۳-۱۵ بررسی چند مورد جالب

آنچه در بالا گذشت توصیف ساده‌ترین روش انتقال مطمئن داده در TCP بود. اما همین روش ساده برخوردار از جزئیات زیرکانه‌ای می‌باشد که تامل بر روی آن‌ها جالب است. برای روشن شدن بیشتر موضوع به تشریح چند مثال می‌پردازیم. نخستین نمونه مربوط به شکل ۳-۳۴ می‌باشد که در آن میزبان A قطعه‌ای را به میزبان B گسیل می‌دارد. اگر این قطعه دارای شماره ردیف 92 و نیز حامل تعداد 8 بایت داده باشد. پس از ارسال این قطعه میزبان A منتظر قطعه‌ای از سوی B با شماره ردیف تائید 100 می‌شود. اگرچه قطعه‌ی ارسالی از A به‌وسیله‌ی B دریافت می‌گردد اما پیام تائید ACK از B به A مفقود می‌شود. در این حالت رخداد زمان خروج اتفاق می‌افتد و میزبان A مجدداً همان قطعه را ارسال می‌کند. بدیهی است هنگامی‌که B انتقال مجدد را دریافت می‌کند از طریق شماره ردیف ملاحظه می‌کند که قطعه شامل داده‌ای است که آن را قبلاً دریافت نموده است. بنابراین TCP در میزبان B از بایت‌های مجدداً گسیل شده در قطعه صرف‌نظر می‌کند.



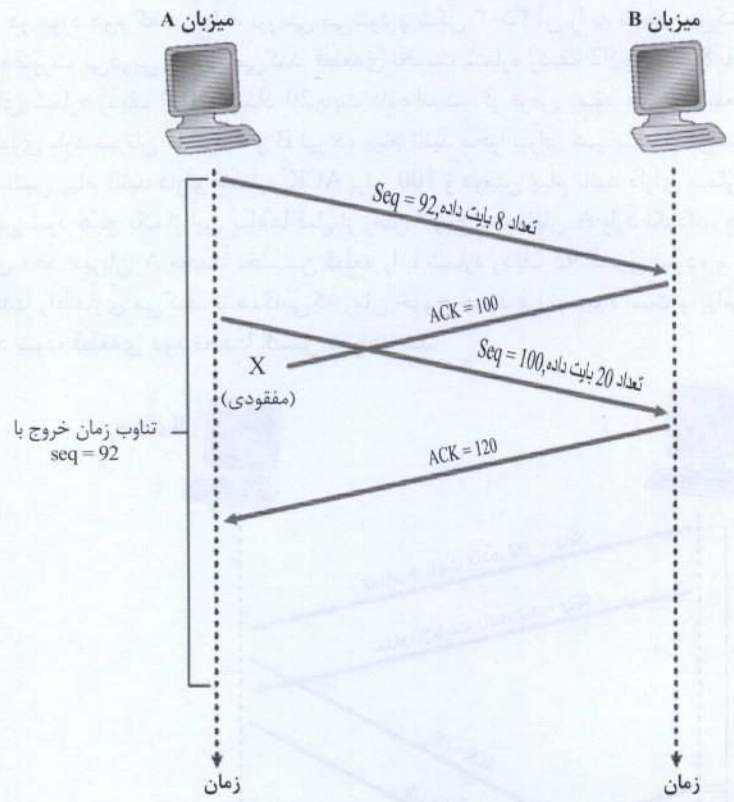
شکل ۳-۳۴ ♦ عملیات انتقال مجدد به واسطه مفقودی پیام تائید

در مورد دوم که در ادامه بررسی می‌شود و شکل ۳-۳۵ آن را به تصویر می‌کشد. میزبان A دو قطعه را به‌صورت پی‌درپی گسیل می‌کند. قطعه‌ی نخست شماره ردیف 92 و تعداد 8 بایت داده و قطعه‌ی دوم دارای شماره ردیف 100 و تعداد 20 بایت داده است. اگر فرض شود هر دو قطعه بدون آسیب و یا تغییری وارد میزبان B گردند و B نیز دو پیام تائید مجزا برای هر یک از آن‌ها گسیل نماید آنگاه نخستین پیام تائید دارای شماره ACK برابر 100 و دومین پیام تائید دارای شماره‌ی 120 می‌باشد. اگر فرض شود هیچ یک از این پیام‌ها قبل از زمان خروج به میزبان A وارد نگردند، هنگامی‌که زمان خروج روی دهد میزبان A مجدداً نخستین قطعه را با شماره ردیف 92 گسیل نموده و زمان‌سنج (تایمر) را مجدداً راه‌اندازی می‌کند. تا هنگامی‌که زمان خروج جدید فرا نرسیده است و پیام ACK برای قطعه دوم وارد شود، قطعه‌ی دوم مجدداً گسیل نخواهد شد.



شکل ۳-۳۵ ♦ قطعه‌ی 100 ارسال مجدد نشده است

در حالت سوم فرض کنید میزبان A دو قطعه را به همان صورت حالت دوم گسیل نماید. پیام تائید ACK متعلق به نخستین قطعه دقیقاً قبل از رخداد زمان خروج در شبکه مفقود شده است. بنابراین میزبان A پیام تائید را با شماره 120 دریافت نموده است نتیجه آنکه میزبان A از این موضوع آگاه می‌شود که طرف مقابل یعنی میزبان B همه‌ی قطعه‌های داده را تا بایت شماره 119 دریافت نموده است. بنابراین A به ارسال مجدد هیچ یک از دو پیام اقدام نمی‌کند. شکل ۳-۳۶ این فرآیند را نشان داده است.



شکل ۳-۳۶ ♦ بکارگیری تائید انباشتی مانع از انتقال مجدد نخستین قطعه می‌شود

- **افزایش دو برابر تناوب زمان خروج**

در این زیر بخش تعدادی از تغییرات که اغلب پیاده‌سازی‌های TCP از آن بهره می‌برند بررسی می‌شود. نخستین مورد مربوط به طول تناوب زمان خروج پس از اتمام آن می‌باشد. در این تغییر هر مرتبه که زمان خروج به وقوع بپیوندد، پروتکل TCP قطعه‌هایی که تا کنون ارسال مجدد نشده‌اند را با توجه به کوچکترین شماره ردیف آن‌ها مطابق آنچه در بالا توضیح داده شد، گسیل می‌نماید. اما با هر بار انتقال مجدد TCP طول زمان تناوب خروج را به دو برابر ارزش قبلی آن افزایش می‌دهد. بنابراین در این رویکرد از روابط (۳-۵) و (۳-۶) جهت محاسبه‌ی اندازه‌ی جدید زمان خروج استفاده نمی‌شود. به‌عنوان مثال اگر Timeout Interval وابسته به قدیمی‌ترین قطعه‌ی تائید نشده 0.75 ثانیه باشد هنگامی‌که تایمر به انتهای زمان شمارش خود می‌رسد پروتکل TCP آن قطعه را مجدداً ارسال می‌دارد و زمان خروج جدید را برابر 1.5 ثانیه می‌کند. اگر این زمان‌شمار نیز به انتهای خود برسد و TCP مجدداً این قطعه را ارسال کند آنگاه زمان به 3 ثانیه افزایش می‌یابد. بنابراین افزایش زمان با هر انتقال مجدد به‌صورت تصاعدی انجام می‌شود. چنانچه زمان‌شمار پس از وقوع هر دو رخداد یعنی دریافت داده از یک کاربرد بالاتر و نیز دریافت پیام ACK راه‌اندازی شود مقدار Timeout Interval با استفاده از دو رابطه‌ی (۳-۵) و (۳-۶) محاسبه می‌گردد. این رویکرد سبب می‌شود نوعی کنترل تراکم محدود شده‌ای به شبکه اعمال گردد. در این حالت وقوع زمان خروج به واسطه‌ی تراکم موجود در شبکه روی می‌دهد که

مفهوم آن انباشته شدن تعداد قابل توجهی بسته در صف یک یا چندین مسیریاب در طول مسیر مبدا و مقصد می‌باشد. بدیهی است در این حالت یا از بسته‌ها صرف‌نظر می‌شود و یا آنکه آن‌ها مجبور به تحمل تاخیر طولانی در صف‌ها می‌باشند. هنگامی‌که تراکم روی می‌دهد اگر منابع ارسال بسته اصرار بر ادامه‌ی فعالیت خود داشته باشند وضعیت تراکم دچار وخامت شدید می‌شود. در چنین حالتی TCP مدیریت کنترل تراکم را اعمال می‌کند که در فصل ۵ بررسی خواهد شد.

- **انتقال مجدد سریع**

یکی از دشواری‌های مربوط به فعال‌سازی زمان خروج انتقال مجدد، مربوط است به طولانی شدن نسبی تناوب این زمان، هنگامی‌که قطعه‌ای مفقود می‌شود طولانی شدن این زمان سبب می‌شود که فرستنده گسیل مجدد بسته‌ی تلف شده را با تاخیر انجام دهد و بنابراین تاخیر حاصل در بین دو نقطه‌ی انتهایی را طولانی‌تر نماید. در اغلب مواقع فرستنده به شناسایی بسته‌ی مفقود شده قبل از رخداد زمان خروج از طریق انجام نسخه‌برداری از ACK و به عبارت دیگر انجام دوبار کپی‌برداری از ACK به این موضوع پی می‌برد. یک پیام ACK **نسخه‌برداری شده** به مفهوم تائید مجدد قطعه‌ای است که فرستنده تائید آن را قبلاً دریافت نموده است. برای درک بهتر واکنش فرستنده به یک ACK نسخه‌برداری شده بایستی علت مبادرت به انجام چنین عملی را از سوی گیرنده شناسایی نمود. جدول ۳-۲ خلاصه‌ای از علل تولید ACK به‌وسیله‌ی گیرنده‌ی TCP را بر اساس دو سند RFC 1122 و RFC 2581 نشان می‌دهد. هنگامی‌که یک گیرنده‌ی TCP قطعه‌ای را با شماره ردیف بزرگتر از قطعه‌ی بعدی مورد انتظار دریافت می‌کند در اثر این رویداد به فاصله‌ی ایجاد شده در جریان داده پی می‌برد. به عبارت دیگر از مفقود شدن قطعه آگاه می‌شود.

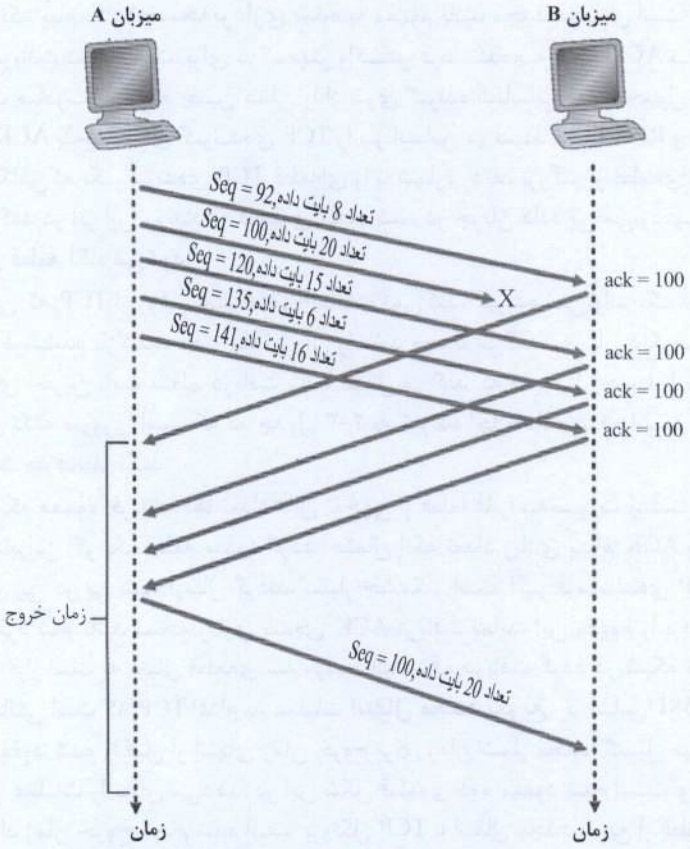
از آنجایی که TCP از روش تائید منفی استفاده نمی‌کند، گیرنده نمی‌تواند یک تائید منفی (NAK) قطعی را به فرستنده بازگشت دهد بنابراین آن را تائید مجدد می‌کند. به بیان دیگر نسخه‌ی دوباره‌ای از ACK را برای آخرین بایت منظم دریافت شده گسیل می‌کند که مفهوم آن دوبار ارسال ACK است. توجه به این نکته ضروری است که در جدول ۳-۲ به گیرنده اجازه داده شده است که از قطعه‌های خارج از نوبت صرف‌نظر نکند.

از آنجائیکه معمولاً فرستنده‌ها تعداد قابل توجهی از قطعه‌ها را به‌صورت پُشت به پُشت گسیل می‌نمایند بنابراین اگر یک قطعه مفقود گردد احتمال آنکه تعداد زیادی پیام ACK به گونه‌ی دوباره نسخه‌برداری پی در پی شده ارسال گردند، بسیار چشمگیر است. اگر فرستنده‌ی TCP از یک قطعه تعداد سه فقره پیام تائید نسخه‌برداری شده‌ی ACK دریافت نماید، این مفهوم را برداشت می‌کند که قطعه‌ای که قرار است به دنبال قطعه‌ی سه مرتبه تائید شده دریافت گردد در شبکه مفقود شده است. در چنین حالتی است که TCP اقدام به عملیات **انتقال مجدد سریع**[1] بر اساس RFC 2581 می‌نماید و قطعه‌ی مفقود شده را قبل از انتهای زمان خروج برای زمان شمار مجدداً گسیل می‌کند. شکل ۳-۳۷ مکانیزم این عملیات را نشان می‌دهد. در این شکل قطعه‌ی دوم مفقود شده است و ارسال مجدد آن قبل از رخداد زمان خروج انجام شده است. پروتکل TCP با انتقال مجدد سریع از قطعه برنامه‌ی ذیل به‌عنوان جایگزین آنچه در شکل ۳-۳۳ آمده است، استفاده می‌کند:

```
event: ACK received, with ACK field vqlue of y
            if (y > sendBase) {
                  sendBase=y
```

1- Fast Retransmit

```
            if (there are currently any not yet
                    acknowledged segments)
                start timer
            }
        else { /* a duplicate ACK for already ACKed
            segment */
            increment number of duplicate ACKs
            received for y
            if (number of duplicate ACKs received
            for y==3)
            /*TCP fast retransmit */
            resend segment with sequence number y
        }
    break;
```



شکل ۳-۳۷ ♦ انتقال مجدد سریع: قطعه‌ی مفقود شده قبل از به انتها رسیدن زمان تایمر برای قطعه گسیل می‌شود.

جدول ۳-۲ ♦ چگونگی تولید پیام ACK در پروتکل TCP [RFC 1122 , RFC 2581]

رخداد	عملیات گیرنده
ورود منظم قطعه با شماره ردیف مورد انتظار. همگی داده‌ها تا شماره ردیف مربوطه تاکنون تائید گردیده‌اند.	پیام تائید تاخیری، زمانی معادل500 میلی‌ثانیه برای ورود قطعه‌ی بعدی اختصاص می‌یابد. اگر قطعه در این فاصله دریافت نگردد پیام ACK گسیل می‌شود.
ورود منظم قطعه با شماره ردیف مورد انتظار و یک قطعه‌ی دیگر با رعایت نوبت در انتظار ارسال تائید ACK می‌باشد.	بی‌درنگ یک پیام ACK انباشتی گسیل گردد که هر دو قطعه‌ی دارای نظم را تائید نماید.
ورود خارج از نوبت قطعه با شماره ردیف بزرگتر از آنچه مورد انتظار است. فاصله در رعایت نوبت تشخیص داده می‌شود.	بی‌درنگ نسخه‌ی دوباره‌ای از ACK گسیل گردد. این به مفهوم شماره بایت مورد انتظار بعدی است که دنباله‌ی کوچکتر فاصله است.
ورود قطعه‌ای که به گونه‌ی نسبی و یا کامل فاصله‌ی ایجاد شده را در داده‌های دریافتی پُر می کند.	بی‌درنگ ACK گسیل گردد مشروط بر آنکه قطعه از دنباله‌ی کوچکتر فاصله آغاز گردد.

رفت و برگشت Go-Back-N و یا تکرار گزینشی SR

پایان بخش مکانیزم‌های بازیابی خطا در پروتکل TCP بررسی و مقایسه‌ی دو پروتکل GBN و یا SR می‌باشد. با توجه به این نکته که پیام‌های تائید TCP انباشتی می‌باشند و قطعه‌های دریافتی سالم اما خارج از نوبت از سوی گیرنده به گونه‌ی منفرد تائید می‌شوند. در نتیجه بنابر آنچه در شکل‌های ۳-۳۳ و نیز ۳-۳۹ نشان داده شده است فرستنده‌ی TCP تنها نیازمند نگهداری کوچکترین شماره ردیف یک بایت ارسال شده اما تائید نشده می‌باشد. به عبارت دیگر در انتظار دریافت بایت Send Base و شماره ردیف بایت بعدی Next Seqnum می‌باشد. با تامل بر روی این حالت TCP بسیار شبیه به پروتکل‌های GBN عمل می‌نماید. اما تفاوت‌های قابل توجهی نیز بین TCP و GBN وجود دارد. بسیاری از پیاده‌سازی‌های TCP قطعه‌های سالم اما خارج از نوبت دریافتی را بافر می‌کنند. اکنون حالتی را در نظر بگیرید که فرستنده‌ای تعدادی قطعه مانند 1, 2, ..., N را ارسال می‌کند و همگی قطعه‌ها به نوبت و بدون خطا وارد گیرنده می‌شوند اگر همچنین پیام تائید ارسالی برای بسته‌ی $n < N$ مفقود گردد اما پیام‌های باقیمانده به تعداد $N - 1$ هر کدام قبل از زمان خروج اختصاص یافته به آن‌ها وارد فرستنده شوند آنگاه در این وضعیت GBN نه تنها مبادرت به انتقال مجدد بسته‌ی n می‌کند بلکه همه‌ی بسته‌های $n + 1$ ، $n + 2$ تا N را نیز مجدداً گسیل می‌دارد. از طرف دیگر TCP حداکثر یک قطعه را گسیل می‌نماید و آن قطعه‌ی n می‌باشد. علاوه بر آن TCP حتی قطعه‌ی n را چنانچه تائید قطعه‌ی n+1 وارد شده باشد، قبل از وقوع زمان خروج n ارسال مجدد نمی‌کند.

تحول پیشنهادی برای اجرا در TCP مربوط به **تائید گزینشی**[1] RFC 2018 است. این روش به یک گیرنده‌ی TCP اجازه می‌دهد به‌صورت گزینشی قطعه‌های خارج از نوبت TCP را تائید نماید. این عمل جایگزین تائید انباشتی برای آخرین قطعه‌ی سالم و به نوبت دریافتی می‌باشد. هنگامی‌که ترکیب انتقال مجدد گزینشی با روش ارسال مجدد قطعه‌هایی که به گونه‌ی گزینشی از سوی گیرنده تائید شده‌اند، انجام می‌گیرد آنگاه قطعه‌های حاصل از این عمل ارسال مجدد نمی‌شوند و TCP شبیه به پروتکل

1- Selective Acknowledgment

ژنریک SR عمل می‌کند. بنابراین مکانیزم بازیابی خطا در TCP را می‌توان ترکیبی از دو پروتکل GBN و SR در نظر گرفت.

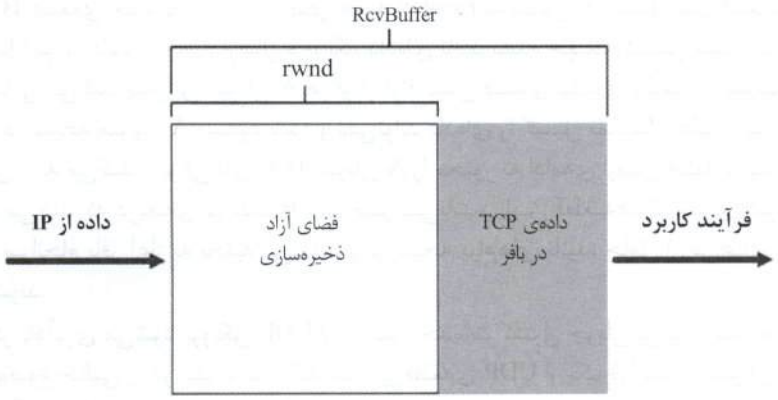
۳-۱۶ کنترل جریان

در مطالب پیشین عنوان شد که هر میزبان در هر طرف ارتباط TCP یک بافر دریافت را به آن پیوند اختصاص می‌دهد. هنگامی‌که ارتباط TCP بایت‌های سالم و به نوبت را دریافت می‌کند آن‌ها را در بافر دریافت تعریف شده‌ای قرار می‌دهد. پردازش کاربردی وابسته به این بافر نسبت به خواندن داده‌ها اقدام می‌کند اما لزوماً پس از وارد شدن آن‌ها این عمل را انجام نمی‌دهد. فرآیند کاربردی دریافت ممکن است در آن لحظه مشغول وظایف دیگری باشد بنابراین هیچگونه تلاشی را در جهت خواندن داده‌های وارد شده تا مدتی پس از قرار گرفتن آن‌ها در بافر انجام نمی‌دهد. چنانچه کاربردی از لحاظ سرعت خواندن داده‌ها کُند باشد فرستنده ممکن است سبب سرریزی بافر دریافت ارتباط گردد. علت آن ارسال داده‌های انبوه با سرعت بالا می‌باشد.

پروتکل TCP جهت جلوگیری از سرریزی بافر دریافت ارتباط برای کاربردهای خود سرویس **کنترل جریان** فراهم می‌آورد. کنترل جریان را می‌توان خدمات نگاشت سرعت تصور نمود. به بیان دیگر ارسال بایت‌ها از سوی فرستنده در مقابل سرعت دریافت و خواندن آن‌ها از سوی کاربرد گیرنده نگاشت می‌شود. قبلاً به این نکته اشاره گردید که در اثر تراکم در بین شبکه‌ی IP فرستنده‌ی TCP را می‌توان از لحاظ سرعت عبور کنترل نمود. این روش کنترل فرستنده اصطلاحاً **کنترل تراکم** نامیده شده است. اگر چه هر دو روش کنترل **جریان** و **تراکم** مشابه یکدیگر می‌باشند و جهت کنترل فرستنده به کار گرفته می‌شوند اما بدیهی است که علل متفاوتی در بهره‌جویی از آن‌ها وجود دارد. شوربختانه اغلب نویسندگان این دو را تبادل‌پذیر یکدیگر می‌دانند اما عاقلانه است که یک خواننده دانا بین این دو تمایز قایل شود. اکنون چگونگی انجام عملیات کنترل جریان از سوی TCP بررسی می‌شود. در تمامی موضوعات TCP که تا کنون بررسی شدند همواره فرض در نظر گرفته شده این گونه بود که گیرنده‌ی TCP از قطعه‌های خارج از نوبت صرف‌نظر می‌کند.

پروتکل TCP از طریق وادار نمودن فرستنده به نگهداری یک متغیر به نام **دریچه‌ی دریافت** کنترل جریان را اعمال می‌کند. از این پنجره جهت تعیین فضای آزاد بافر در گیرنده و اعلان آن به فرستنده استفاده می‌شود. از آنجائیکه TCP یک پروتکل دو طرفه‌ی همزمان است فرستنده در هر دو سوی ارتباط موظف به تعریف یک دریچه‌ی دریافت مجزا می‌باشد. در مثال ذیل پنجره‌ی دریافت در چارچوب یک عملیات انتقال فایل بررسی می‌شود. اگر فرض شود میزبان A در حال گسیل یک فایل پر حجم به میزبانی مثل B از طریق یک ارتباط TCP باشد. میزبان B یک بافر دریافت را به این ارتباط اختصاص می‌دهد و آن را RcvBuffer می‌نامد. در فواصل زمانی معین برنامه‌ی کاربردی در B اطلاعاتی را از بافر می‌خواند و متغیرهای ذیل را تعریف می‌کند:

- متغیر Last ByteRead به مفهوم شماره‌ی آخرین بایت خوانده شده از بافر به‌وسیله‌ی فرآیند کاربردی B در جریان داده‌ها می‌باشد.
- متغیر Last ByteRcv به مفهوم شماره‌ی آخرین بایت وارد شده از شبکه و قرار گرفتن آن در بافر دریافت B در جریان داده‌ها است.



شکل ۳-۳۸ ♦ دریچه‌ی دریافت rwnd و بافر دریافت RcvBuffer

از آنجائیکه TCP مجاز به سرریزی بافر اختصاص یافته نمی‌باشد بایستی همواره رابطه‌ی ذیل برقرار گردد.

$$LastByteRcvd - LastByteRead \leq RcvBuffer$$

پنجره‌ی دریافت rwnd با اندازه‌ی حجم فضای آزاد در بافر مقدار دهی می‌شود:

$$rwnd = RcvBuffer - [LastByteRcv - LastByteRead]$$

از آنجائیکه اندازه‌ی این فضای ذخیره‌سازی نسبت به زمان متغیر است بنابراین rwnd نیز پویا می‌باشد. شکل ۳-۳۸ این متغیر را نشان می‌دهد.

اکنون این پرسش مطرح می‌شود که چگونه ارتباط TCP از متغیر rwnd جهت انجام فرآیند کنترل جریان استفاده می‌کند؟ پاسخ این پرسش را می‌توان اینگونه بیان کرد که میزبان B مقدار فضای آزاد در بافر ارتباط را از طریق قرار دادن ارزش جاری rwnd در محدوده‌ی پنجره‌ی دریافت هر قطعه‌ی ارسالی به A تنظیم می‌کند. در آغاز عملیات میزبان B مقدار دهی اولیه را به صورت: rwnd=RcvBuffer تعریف می‌کند. بدیهی است برای احراز موفقیت در انجام این مهم میزبان B بایستی چندین متغیر ویژه‌ی ارتباط را رهگیری نماید.

میزبان A نیز دو متغیر را رهگیری می‌کند این دو عبارتند از LastByteSend و LastByteAcked که مفهوم هر یک از آن‌ها مشخص است. لازم به توضیح است که تفاضل دو متغیر: LastByteSend – LastByteAcked عبارت است از تعداد داده‌های تائیدنشده‌ای که میزبان A به سوی ارتباط گسیل نموده است. نگهداری تعداد داده‌های تائید نشده **کمتر از** مقدار rwnd است که میزبان A از عدم وقوع سرریزی در بافر دریافت میزبان B مطمئن می‌شود. بنابراین میزبان A در سراسر دوره‌ی حیات ارتباط همواره این اطمینان را برای خود حاصل می‌کند که:

$$LastByteSend - LastByteAcked \leq rwnd$$

این روش دچار یک ضعف تکنیکی نسبتاً کوچکی می‌باشد. برای ملاحظه‌ی این ضعف به سناریوی ذیل توجه فرمایید. فرض کنید بافر دریافت میزبان A کاملاً انباشته شده است و بنابراین rwnd = 0 است. پس از آگاه نمودن A از مقدار جاری rwnd ، فرض کنید میزبان B داده‌ای را جهت گسیل به A در اختیار ندارد. در این حالت هنگامی‌که یک پردازش کاربردی در B مشغول تخلیه‌ی بافر می‌باشد،

پروتکل TCP قطعه‌ی جدیدی را با ارزش‌های جدید از rwnd به میزبان A گسیل نمی‌کند بلکه TCP در صورت دارا بودن داده‌ای جهت ارسال و یا یک داده‌ای تائید نشده جهت گسیل مبادرت به انجام چنین عملیاتی می‌کند. بنابراین میزبان A هرگز از آزاد شدن فضای بافر دریافت در میزبان B آگاه نمی‌شود و در نتیجه میزبان A مسدود شده و نمی‌تواند داده‌ای را گسیل نماید! راهکار این چالش را TCP چنین ارایه می‌کند، ویژگی‌های TCP میزبان A را مجبور به ادامه‌ی ارسال قطعه با یک بایتِ داده در مواردی می‌کند که دریچه‌ی دریافت B برابر صفر می‌باشد. این قطعه‌ها از سوی گیرنده تائید می‌شوند و سرانجام بافر آغاز به تخلیه می‌نماید و در نتیجه پیام‌های تائید، حامل ارزش‌های غیر صفر rwnd می‌شوند.

بار دیگر یادآوری می‌شود پروتکل UDP از مکانیزم خدمات کنترل جریان بی‌بهره است و برای درک بهتر این موضوع حالتی را در نظر بگیرید که تعدادی قطعه‌ی UDP از یک فرآیند در میزبان A در حال گسیل به فرآیند دیگری بر روی میزبان B می‌باشند. در یک پیاده‌سازی UDP این قطعه‌ها در قالب یک بافر با اندازه‌ی معین تعریف می‌شوند که مقدم بر سوکت مربوطه می‌باشد. پردازش هر بار یک قطعه‌ی کامل را از بافر فراخوانی می‌کند. اگر پردازش با شتاب کافی قادر به خواندن قطعه‌ها از بافر نباشد آنگاه سرریزی بوجود آمده در بافر سبب حذف قطعه‌ها می‌شود.

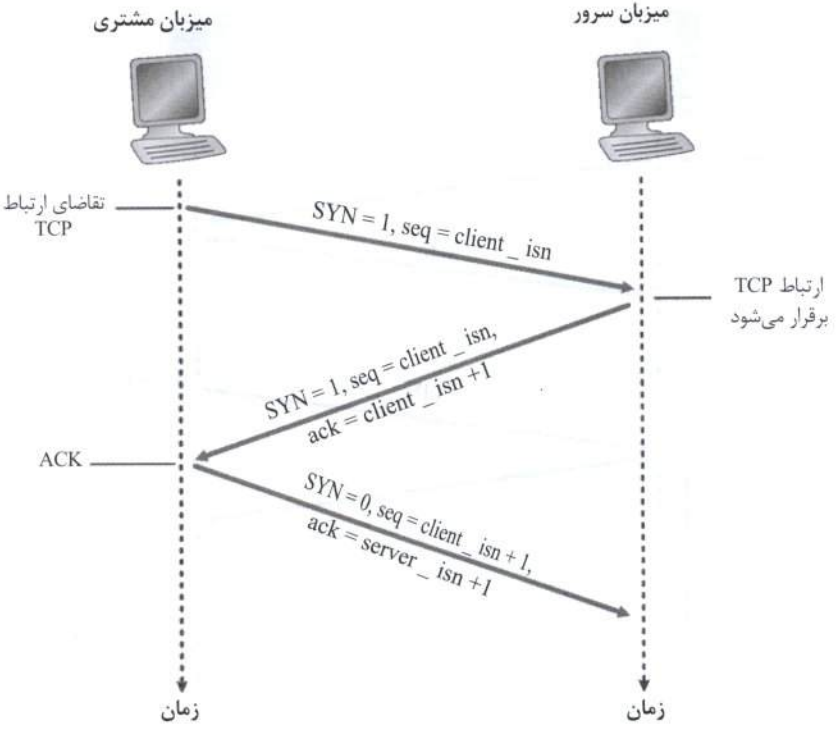
۳-۱۷ مدیریت ارتباط TCP

در این زیر بخش، مدیریت ارتباط TCP از نقطه‌نظر برقراری و قطع آن بررسی می‌شود. چگونگی برقراری ارتباط و قطع آن به‌صورت چشمگیر می‌تواند در افزایش تاخیرها در یک شبکه مؤثر باشد. علاوه بر آن بسیاری از تهاجم‌های انجام شده به شبکه و از جمله معروف‌ترین آن به نام سیلاب SYN از نقاط ضعف موجود در مدیریت ارتباط TCP سوء استفاده نموده‌اند. در آغاز چگونگی ایجاد یک ارتباط TCP بررسی می‌شود. اگر فرض شود فرآیندی بر روی میزبانی مانند یک مشتری متقاضی ارتباط با فرآیند دیگری بر روی میزبان یا سرور دیگری است. برنامه‌ی کاربرد مشتری ابتدا مشتری TCP را از منظور خود جهت برقراری ارتباط با یک برنامه‌ی سرور آگاه می‌کند. پس از آن برنامه‌ی TCP در مشتری نسبت به برقراری یک ارتباط TCP با برنامه‌ی سرور با رعایت موارد ذیل اقدام می‌کند:

- **گام ۱** برنامه‌ی مشتری TCP قطعه‌ی TCP ویژه‌ای را به سمت برنامه‌ی سرور TCP گسیل می‌کند. این قطعه حامل هیچ داده‌ای از نوع لایه‌ی کاربرد نمی‌باشد. اما یکی از نشان‌های سرپیام قطعه بر اساس شکل ۳-۲۹ بیت SYN می‌باشد. علاوه بر آن مشتری به گونه‌ی تصادفی شمار ردیف اولیه‌ای را گزینش می‌کند با نام (client-isn) و آن را در میدان شماره ردیف قطعه‌ی نخستین SYN در TCP قرار می‌دهد. این قطعه در قالب یک داده گرام IP کپسول‌بندی و به سمت سرور گسیل می‌شود. دقت قابل ملاحظه‌ای از لحاظ گزینش تصادفی client-isn جهت جلوگیری از تهدیدهای امنیتی ویژه از سوی طراحان پروتکل‌های TCP اعمال می‌شود.
- **گام ۲** هنگامی‌که داده گرام IP حامل قطعه‌ی TCP SYN وارد میزبان سرور می‌شود. البته در صورتی‌که وارد شود! برنامه‌ی سرور آنگاه این قطعه را از داده گرام جدا نموده، بافرهای TCP و متغیرهایی را به ارتباط اختصاص می‌دهد و سپس یک قطعه داده که مجوز ارتباط قلمداد می‌شود را به مشتری TCP گسیل می‌کند. در فصل ۸ اختصاص این بافرها و متغیرها قبل از تکمیل گام 3 یک دست‌دهی سه طرفه است که پروتکل TCP را در برابر یک حمله‌ی DoS

شناخته شده‌ای مانند سیلاب SYN دچار ضعف می‌سازد. این قطعه‌ی مجوز ارتباط نیز حامل هیچ داده از نوع لایه‌ی کاربرد نمی‌باشد. اما شامل سه قطعه‌ی مهم از اطلاعات در سر پیام قطعه می‌باشد. ابتدا بیت SYN برابر 1 می‌شود، آنگاه در سر پیام قطعه‌ی TCP، میدان تائید ACK با ارزش متغیر client-isn+1 مقداردهی می‌شود و در انتها سرور شماره ردیف اولیه‌ی خود server-isn را گزینش و مقدار آن را در میدان شماره ردیف سر پیام قطعه‌ی TCP قرار می‌دهد. آنچه در واقع قطعه‌ی مجوز ارتباط بیان می‌کند مشابه آن است که بگوید: " من بسته‌ی SYN را جهت آغاز یک ارتباط با شماره ردیف اولیه‌ی client-isn دریافت نمودم. موافقت خود را با برقراری این ارتباط اعلام می‌کنم. شماره ردیف اولیه من نیز server-isn می‌باشد". قطعه‌ی مجوز برقراری ارتباط اصطلاحاً قطعه‌ی SYNACK نامیده می‌شود.

- **گام ۳** هنگام دریافت قطعه‌ی SYNACK برنامه‌ی مشتری بافرها و متغیرها را به ارتباط اختصاص می‌دهد. میزبان مشتری آنگاه قطعه‌ی دیگری را به سوی سرور گسیل می‌نماید. این آخرین قطعه مجوز ارتباط سرور را تائید می‌کند. برنامه‌ی مشتری این تائید را از طریق قرار دادن ارزش server-isn+1 در دامنه‌ی سرپیام قطعه‌ی تائید TCP انجام می‌دهد. آنگاه بیت SYN را صفر می‌کند، زیرا ارتباط برقرار شده است. این سومین گام از دست‌دهی سه طرفه است و ممکن است داده‌ی برنامه‌ی مشتری به سرور را در قطعه محموله‌ی داده با خود انتقال دهد.

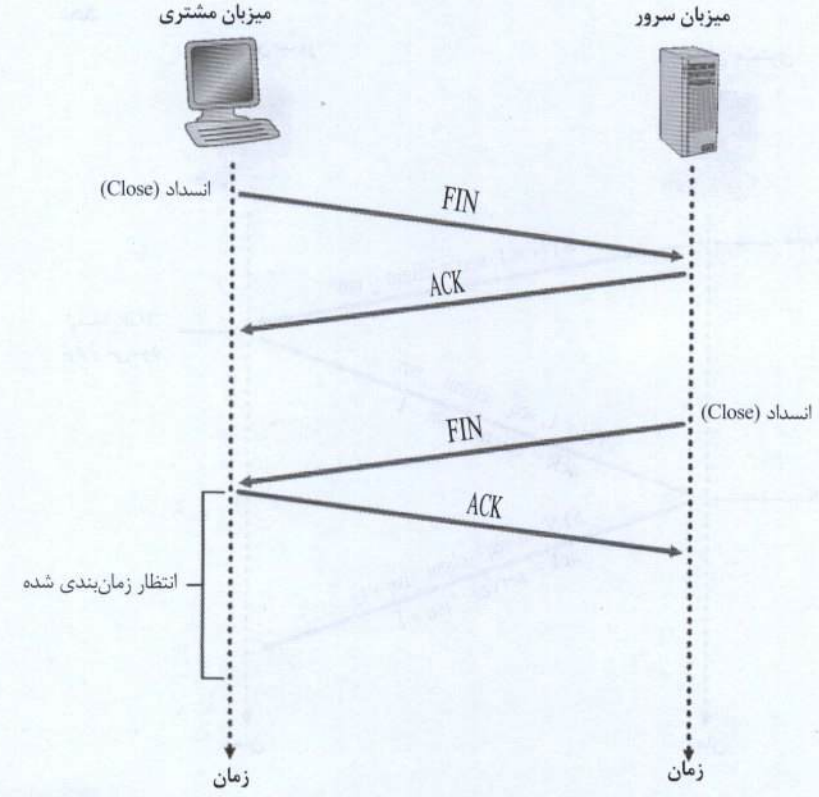


شکل ۳-۳۹ ♦دست‌دهی سه‌جانبه‌ی TCP : مبادله‌ی قطعه

هنگامی‌که این سه مرحله پایان می‌پذیرد، میزبان‌های مشتری و سرور می‌توانند به تبادل قطعه‌های

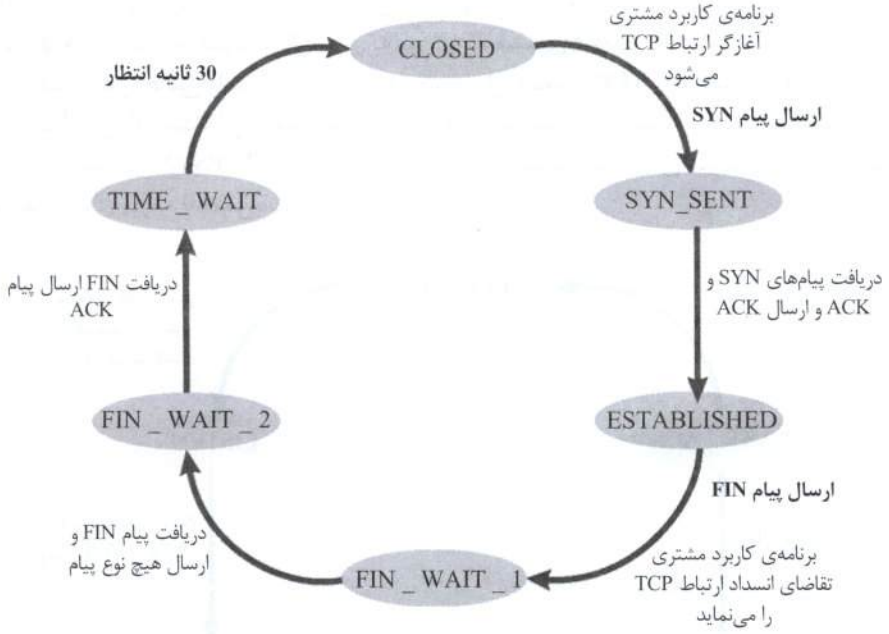
داده بین یکدیگر مبادرت نمایند. در هر یک از قطعه‌های بعدی بیت SYN برابر صفر می‌شود. هنگام برقراری ارتباط تعداد سه بسته بین دو میزبان بر اساس شکل ۳-۳۹ مبادله می‌شود. به همین دلیل این گام در برقراری ارتباط اصطلاحاً **دست‌دهی سه‌جانبه** نامیده شده است. بدیهی است پس از استفاده از ارتباط TCP یکی از دو فرآیند مشتری و یا سرور می‌توانند به این ارتباط پایان دهند. هنگامی‌که ارتباط قطع می‌گردد بافرها و متغیرهای تعریف شده در میزبان‌ها جابه‌جا شده و به ارتباط دیگری اختصاص می‌یابند. به‌عنوان مثال فرض کنید یک مشتری تصمیم به انسداد ارتباطی می‌گیرد در شکل ۳-۴۰ نشان داده شده است که پردازش کاربرد مشتری فرمان مسدود شدن ارتباط را صادر می‌کند. این فرمان سبب می‌شود برنامه‌ی مشتری TCP قطعه‌ی TCP ویژه‌ای را به فرآیند سرور گسیل نماید. این قطعه‌ی ویژه حامل یک بیت نشان در سرپیام قطعه می‌باشد و آن بیت FIN می‌باشد که ارزش آن برابر یک می‌شود. هنگامی‌که برنامه‌ی سرور این قطعه را دریافت نماید در پاسخ یک قطعه‌ی تائید را به مشتری بازگشت می‌دهد.

برنامه‌ی سرور آنگاه قطعه‌ی مسدود نمودن وابسته به خود را گسیل می‌نماید که بیت FIN آن برابر یک می‌باشد. سرانجام مشتری دریافت قطعه‌ی انسداد سرور را تائید می‌کند. در این حالت است که همه‌ی منابع در دو میزان جابه‌جا شده و به ارتباط دیگری تعلق می‌گیرند.



شکل ۳-۴۰ ♦ انسداد یک ارتباط TCP

در طول دوره‌ی حیات یک ارتباط TCP، پروتکل TCP در حال اجرا بر روی هر میزبان گذرهایی را از درون **حالت‌های TCP** [1] گوناگون انجام می‌دهد. شکل ۳-۴۱ زنجیره‌ای از این حالت‌ها را که از سوی یک مشتری TCP ملاقات شده‌اند به تصویر کشیده است. برنامه‌ی مشتری TCP ملاقات‌های خود را از حالت CLOSED آغاز می‌کند. کاربرد سمت مشتری ارتباط TCP جدیدی را از طریق تعریف یک سوکت می‌گشاید. این موضوع سبب می‌شود TCP در مشتری یک قطعه‌ی SYN را به برنامه‌ی TCP سرور گسیل نماید. پس از ارسال قطعه‌ی SYN مشتری TCP وارد حالت SYN-SENT می‌شود. هنگامی‌که مشتری در این حالت قرار می‌گیرد منتظر قطعه‌ای از برنامه‌ی TCP می‌شود که شامل یک پیام تائید برای قطعه‌ی قبلی برنامه‌ی مشتری و حامل بیتِ SYN با ارزش یک می‌باشد. پس از دریافت چنین قطعه‌ای مشتری TCP وارد حالت ESTABLISHED می‌شود. تا زمانیکه در این حالت قرار دارد برنامه‌ی مشتری TCP می‌تواند قطعه‌های TCP را گسیل و یا دریافت نماید که همگی آن‌ها حامل داده‌های تولید شده از سوی لایه کاربرد می‌باشند.



شکل ۳-۴۱ ♦ زنجیره حالت‌های TCP که از سوی یک مشتری TCP ملاقات شده است.

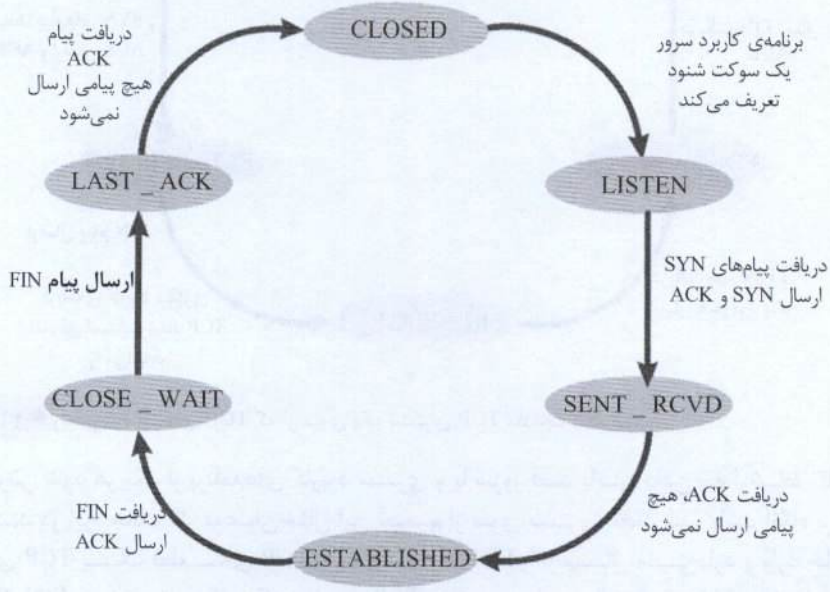
اگر فرض شود هر یک از برنامه‌های کاربرد مشتری و یا سرور قصد پایان دادن به ارتباط TCP را داشته باشند در این حالت اگر به‌عنوان مثال این تصمیم از سوی مشتری اتخاذ شده باشد آنگاه برنامه‌ی مشتری TCP یک قطعه‌ی TCP با بیت FIN=1 را ارسال می‌دارد و وارد حالت FIN_WAIT_1 می‌شود. تا هنگامی‌که برنامه‌ی TCP مشتری در این حالت قرار دارد به انتظار یک قطعه‌ی TCP از جانب سرور که همراه آن یک پیام تائید است باقی می‌ماند. هنگام دریافت این قطعه مشتری TCP وارد حالت FIN_WAIT_2 می‌گردد و تا زمانیکه در این حالت قرار دارد منتظر قطعه‌ی

1- TCP States

بعدی از سرور با بیت FIN=1 می‌شود. پس از دریافت این قطعه برنامه‌ی مشتری قطعه‌ی ارسالی سرور را تائید و وارد مرحله‌ی TIME-WAIT می‌شود. این مرحله به مشتری TCP اجازه می‌دهد پیام تائید نهایی را در صورتی‌که ACK مفقود شده است مجدداً گسیل نماید. زمان تلف شده در حالت TIME-WAIT بستگی به نوع پیاده‌سازی انجام شده دارد و معمولاً ارزش‌هایی ما بین 30، 60 و یا 120 ثانیه را شامل می‌شود. پس از گذشت زمان انتظار ارتباط رسماً مسدود می‌شود و همه‌ی منابع طرف مشتری شامل شماره پورت‌ها آزاد می‌شوند.

شکل ۳-۴۲ مجموعه‌ای از حالت‌های ملاقات شده از سوی سرور TCP را با فرض درخواست قطع ارتباط از سوی مشتری نشان می‌دهد. گذرهای انجام شده از نظر توضیحات روشن و نیازی به شرح ندارند. در این دو تصویر انتقال حالت‌ها فقط چگونگی برقراری و سپس قطع یک ارتباط TCP نشان داده شده است. به عبارت دیگر سناریوهای ویژه‌ای از قبیل درخواست همزمان دو طرف جهت برقراری و یا قطع ارتباط نیز می‌تواند روی دهد که موارد آن تصویر نشده است.

آنچه در بالا در نظر گرفته شد رعایت این فرض بود که هر دو طرف سرور و مشتری جهت ارتباط آمادگی لازم را دارند. به عبارت دیگر سرور به پورتی که مشتری قطعه‌ی SYN را گسیل می‌کند، گوش فرا می‌دهد. اکنون حالتی را در نظر بگیرید که یک میزبان قطعه‌ی TCP را دریافت می‌دارد که شماره پورت و یا آدرس IP مبداء آن با هیچیک از سوکت‌های موجود در آن میزبان نگاشت نمی‌شود. به‌عنوان مثال میزان بسته‌ی TCPSYN با آدرس مقصد 80 را دریافت می‌کند اما آن میزبان قادر به پذیرش ارتباط‌ها بر روی پورت 80 نمی‌باشد.



شکل ۳-۴۲ ♦ زنجیره حالت‌های TCP که از سوی یک سرور TCP ملاقات شده است

به بیان دیگر این میزبان برنامه‌ی وب سرور، بر روی پورت 80 را اجرا نمی‌کند. در این حالت میزبان یک قطعه‌ی ویژه را به مبدا گسیل می‌نماید. این قطعه‌ی TCP دارای بیت نشان RST می‌باشد که ارزش یک را اختیار نموده است. بنابراین هنگامی‌که میزبان این قطعه را گسیل می‌کند آنچه در

حقیقت به مبدا اعلام می‌کند مانند این عبارت است که:" من سوکتی را برای قطعه‌ی ارسالی در اختیار ندارم. لطفاً از ارسال مجدد این قطعه خودداری نمائید". هنگامی‌که میزبانی یک بسته‌ی UDP با شماره پورت غیرقابل نگاشت با هیچیک از سوکت‌های جاری خود را دریافت می‌کند آن میزبان یک داده گرام ویژه‌ی ICMP را که در فصل ۴ بررسی شده است گسیل می‌نماید. اکنون می‌توان به این نتیجه رسید که درک نسبتاً قابل قبولی از مدیریت ارتباط TCP را بدست آورده‌ایم. بنابراین بار دیگر ابزار پویش پورت nmap را مجدداً بررسی و چگونگی عملکرد آن را از نزدیک تحلیل می‌کنیم. جهت کاوش یک پورت TCP ویژه مانند 6789 بر روی یک میزبان هدف، فرآیند nmap قطعه‌ی TCPSYN را با شماره پورت مقصد 6789 به آن میزبان گسیل می‌کند. نتیجه‌ی حاصل از این ارسال را می‌توان در سه حالت ذیل خلاصه نمود:

۱- **میزبان مبداء قطعه‌ی TCPSYNACK را از میزبان هدف دریافت می‌کند.** از آنجائیکه مفهوم این عبارت یعنی کاربردی در حال اجرا با پورت 6789 متعلق به TCP در میزبان مقصد است پاسخ nmap ارسال پیام Open خواهد بود.

۲- **میزبان مبدا یک قطعه ی TCPRST را از میزبان مقصد دریافت می‌کند.** این به مفهوم آن است که قطعه‌ی SYN به میزبان مقصد رسیده است اما آن میزبان در حال اجرای کاربرد با پورت TCP شماره 6789 نمی‌باشد. در این حالت مهاجم می‌داند که قطعه‌ی هدف‌گیری شده به میزبان پورت 6789 از سوی هیچ مکانیزم امنیتی مانند دیواره‌ی آتش و نظایر آن در مسیر بین میزبان‌ها مبدا و مقصد مسدود نشده است.

۳- **مبدا هیچ‌گونه قطعه‌ای را دریافت نمی‌کند.** به احتمال قوی مفهوم این واکنش آن است که قطعه‌ی SYN از سوی یک مانع امنیتی شبکه مسدود شده و هرگز به میزبان هدف نرسیده است.

ملاحظه گردید ابزار nmap مکانیزم قدرتمندی است که قادر به ردیابی واکنش‌های پورت‌های آزاد TCP و نیز UDP می‌باشد. از این ابزار جهت پیکربندی نرم‌افزارهای امنیتی شبکه و حتی نسخه‌هایی از برنامه‌های کاربردی و سیستم عامل‌ها می‌توان استفاده نمود. بسیاری از این امکانات را می‌توان با به کارگیری قطعه‌های مدیریت ارتباط TCP فعال نمود. اگر با یک ماشین Linux مشغول کار می‌باشید می‌توانید تجربه‌ای را از nmap بدست آورید. کافی است در خط فرمان عبارت nmap را ماشین کنید. برای سایر سیستم‌عامل‌ها می‌توانید nmap را از طریق آدرس: http://insecure.org/nmap بارگیری نمائید.

در این مقطع از فصل ۳ موارد مربوط به کنترل خطا و جریان در TCP به انتها می‌رسد در بخش ۳-۲۲ مجدداً به TCP بازگشت نموده و کنترل تراکم را در این پروتکل با جزئیات بیشتر بررسی می‌نمائیم. اما قبل از آن موضوعات مربوط به کنترل تراکم را در یک چارچوب گسترده‌تر بررسی می‌کنیم.

۳-۱۸ تهاجم سیلابی SYN

در بحث مربوط به دست‌دهی سه طرفه‌ی TCP ملاحظه شد که یک سرور متغیرها و بافرهایی را در واکنش به دریافت بیت SYN به یک ارتباط TCP اختصاص و مقداردهی اولیه می‌کند. برنامه‌ی سرور آنگاه یک پیام SYNACK را گسیل و منتظر می‌شود تا قطعه‌ی ACK را از جانب برنامه‌ی مشتری دریافت نماید. سومین و آخرین دست‌دهی، قبل از عملیات کامل برقراری یک ارتباط صورت می‌پذیرد. اگر مشتری پیام

ACK را برای تکمیل گام سوم دست‌دهی سه جانبه گسیل نکند، پس از گذشت یک یا چند دقیقه برنامه‌ی سرور این ارتباط نیمه باز را خاتمه داده و همه‌ی منابع اختصاص یافته را باز پس می‌گیرد.

این روش پروتکل مدیریت ارتباط TCP راه را برای تهاجم کلاسیک DoS که اصطلاحاً **تهاجم سیلابی SYN**[1] نامیده شده است، هموار می‌سازد. در این تهاجم تعداد انبوهی قطعه‌ی TCPSYN از سوی برنامه‌ی مهاجم قبل از تکمیل گام سوم دست‌دهی به سوی هدف گسیل می‌شود. توان حمله را می‌توان از طریق گسیل قطعه‌های SYN از سوی چندین منبع مهاجم تقویت و گسترش داد. با انجام این عمل تهدید تبدیل به DDoS که همان تهاجم سیلابی SYN به‌شمار می‌آید، خواهد شد. با این سیلاب قطعه‌های SYN منابع ارتباطی برنامه‌ی سرور به سرعت رو به کاهش می‌گذارد زیرا همگی آن‌ها به ارتباط‌های نیمه‌باز TCP اختصاص می‌یابد که هرگز مورد استفاده قرار نمی‌گیرد. با کاهش شدید منابع سرور، کاربران مجاز محروم از این خدمات خواهند شد. نظیر چنین تهاجماتی در سال‌های 1996 و 2009 میلادی به وقوع پیوست.

تهاجم سیلابی SYN بسیار مخرب و اثرگذار می‌باشد. با این همه یک راهکار مؤثر دفاعی نیز وجود دارد و آن استفاده از **کوکی‌های SYN**[2] می‌باشد که این راه‌کار در اغلب سیستم‌عامل‌های مهم گنجانیده شده است. طریقه‌ی عملکرد این کوکی‌ها به گونه‌ی ذیل است:

- هنگامی‌که برنامه‌ی سرور قطعه‌ای را دریافت می‌کند، قادر به تشخیص اینکه این قطعه از سوی یک کاربر مجاز است و یا آن که بخشی از یک تهاجم سیلابی SYN است، نمی‌باشد. بنابراین سرور برای آن که یک ارتباط TCP نیمه‌باز خلق نمی‌کند. به جای آن برنامه‌ی سرور یک شماره ردیف TCP اولیه‌ای را که یک تابع پیچیده‌ی از نوع Hash می‌باشد از آدرس‌های IP مبدا و مقصد و نیز شماره پورت قطعه‌ی SYN به همراه یک ارزش رمز شناخته شده که فقط سرور آن را می‌شناسد، تعریف می‌کند. برنامه‌ی سرور از این ارزش رمز برای تعداد زیاد ارتباط‌های TCP استفاده می‌کند. این شماره ردیف دقیق و در عین حال پیچیده‌ای که خلق می‌شود اصطلاحاً **کوکی** نامیده می‌شود. برنامه‌ی سرور سپس یک بسته‌ی SYNACK را به همراه این شماره ردیف ویژه گسیل می‌کند. بیان این نکته ضروری است که برنامه‌ی سرور، کوکی و یا حالت‌های دیگر اطلاعات را در رابطه با SYN به هیچ عنوان در حافظه‌ی خود نگهداری نمی‌کند.
- اگر برنامه‌ی مشتری مجاز باشد آنگاه قطعه‌ی ACK را بازگشت می‌دهد. سرور بی‌درنگ پس از دریافت این پیام ACK بایستی اطمینان حاصل نماید که آن ACK مرتبط با پیام SYN گسیل شده‌ی قبلی می‌باشد. اما چگونه سرور از این موضوع آگاه می‌شود در صورتی‌که قبلاً اشاره گردید ذهنیتی از اطلاعات ندارد؟ همانگونه که ممکن است حدس زده باشید کسب این آگاهی را از طریق کوکی انجام می‌دهد. برای یک ACK قانونی، ارزش درون دامنه‌ی پیام تائید برابر است با شماره ردیف (SYNACK+1) همانگونه که در شکل ۳-۳۹ نشان داده شده است. برنامه‌ی سرور آنگاه همان تابع را با میدان‌های مشابه در قطعه‌ی ACK به همراه شماره‌ی رمز ویژه اجرا می‌کند و اگر نتیجه‌ی حاصل از تابع با جمع ارزش 1 مشابه شماره تائید باشد آنگاه سرور نتیجه‌گیری می‌کند که پیام ACK وابسته به قطعه‌ی SYN قبلی است و بنابراین تهدیدآمیز نمی‌باشد. در این حالت است که سرور یک ارتباط کامل TCP را به همراه سوکت آن می‌گشاید.

1- SYN flood attack
2- SYN cookies

- از طرف دیگر اگر برنامه‌ی مشتری یک قطعه‌ی ACK را بازگشت ندهد آنگاه پیام SYN اصلی خسارتی را به سرور وارد نمی‌کند زیرا برنامه‌ی سرور هیچ‌گونه امکاناتی را به آن اختصاص نداده است.

کوکی‌های SYN به گونه‌ای مؤثر، تهدید برآمده از یک تهاجم سیلابی SYN را حذف می‌کنند. تغییری که می‌تواند از سوی یک تهاجم سیلابی SYN بوجود آید آن است که برنامه‌ی مشتری مهاجم، یک قطعه‌ی کاملاً مجاز و مشروع ACK را برای هر یک از قطعه‌های SYNACK که سرور آن‌ها را خلق می‌نماید، بازگشت دهد. این موضوع سبب می‌شود سرور یک ارتباط TCP کاملاً باز را تعریف کند اگرچه سیستم‌عامل آن از کوکی‌های SYN استفاده می‌کند. اگر ده‌ها هزار برنامه‌ی مشتری به گونه‌ی DDoS با آدرس‌های منبع IP متفاوت جهت تهاجم مورد استفاده قرار گیرند، آنگاه تشخیص مشتری‌های مجاز از غیر مشروع برای سرور بسیار دشوار و غیر ممکن خواهد بود. بنابراین دفاع در برابر این **تهاجم دست‌دهی کامل شده** بسیار دشوارتر از تهاجم سیلابی SYN می‌باشد.

۳-۱۹ اصول کنترل تراکم

در بخش‌های قبل روش‌های کلی و ویژه‌ای که TCP از آن جهت انتقال مطمئن داده برای مقابله با اتلاف بسته‌ها استفاده می‌کند، تشریح گردید. در آن مطالب خاطر نشان گردید که در عمل سرریزی بافرها در مسیریاب‌ها سبب مفقود شدن بسته می‌شود و انتقال مجدد بسته‌ها یک رویکرد جهت حل چالش تراکم در شبکه و جلوگیری از اتلاف قطعه‌ها در لایه انتقال می‌باشد. بدیهی است این رویکردها نمی‌توانست از ایجاد تراکم در شبکه جلوگیری نماید. جهت جلوگیری از به وجود آمدن تراکم در شبکه و به عبارت دیگر پیشگیری قبل از درمان روش‌هایی جهت کنترل فرستنده‌ها در برابر ایجاد تراکم در شبکه نیاز می‌باشد.

در این بخش ابتدا چالش کنترل تراکم در یک چارچوب کلی بررسی می‌شود و دلایل کافی جهت توجیه زیان‌باری تراکم و تاثیر آن بر کارایی کاربردهای لایه‌های بالاتر تشریح می‌گردد. مطالعه‌ی کنترل تراکم ترافیک در شبکه از آن جهت اهمیت دارد که آن را می‌توان یکی از ده مورد اصلی دشواری‌های یک شبکه به‌شمار آورد. در انتهای این بخش توضیحات سودمندی پیرامون کنترل تراکم در رویکرد آنچه اصطلاحاً **نرخ بیت در دسترس**[1] (ABR) در شبکه‌هایی که از روش **انتقال غیر همزمان**[2] (ATM) استفاده می‌کنند، ارائه می‌گردد. اما درآغاز مطالب این بخش جزئیات مربوط به الگوریتم‌های کنترل تراکم در TCP تشریح می‌شود.

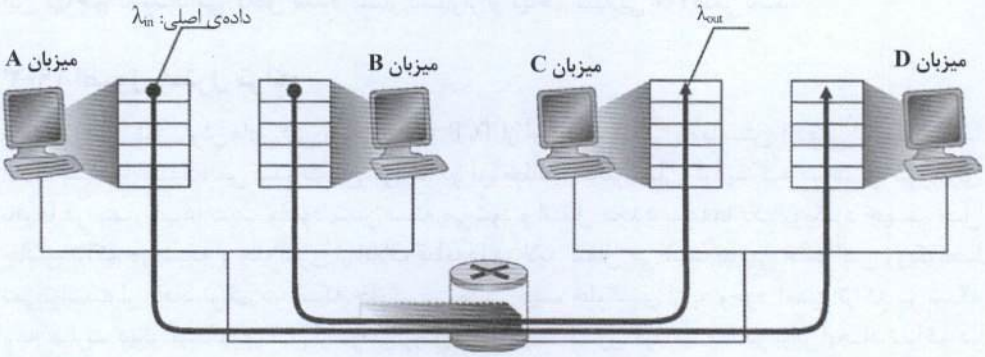
• علل و هزینه‌های تراکم در شبکه‌های کامپیوتری

جهت آگاهی از علل و هزینه‌ای که تراکم بر یک شبکه تحمیل می‌نماید سه وضعیت نسبتاً پیچیده‌ای که سبب ایجاد تراکم و به دنبال آن هزینه‌ای که این وضعیت به منابع شبکه تحمیل می‌نماید بررسی می‌شود. بدیهی است در این مقطع از بررسی راه‌کارهای جلوگیری از وقوع این رخداد و نیز بررسی این که در هنگام افزایش سرعت انتقال یک میزبان که ممکن است نتیجه‌ی آن سبب تراکم شبکه گردد چه رویدادهایی به وقوع می‌پیوندد، خودداری شده است.

1- Available bit rate
2- Asynchronous transmission mode

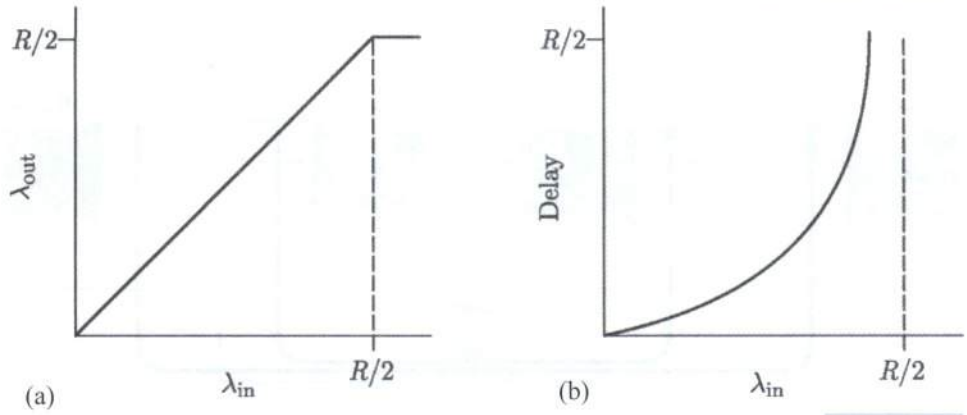
- **حالت نخست: دو فرستنده و یک مسیریاب با تعداد بی‌شماری بافر**

ساده‌ترین حالت ممکن در ایجاد تراکم در شبکه هنگامی است که دو میزبان A و B هر کدام از یک ارتباط مشترک بر اساس شکل ۳-۴۳ استفاده می‌کنند. اگر فرض گردد کاربردی در میزبان A در حال تزریق داده‌ای به ارتباطی مانند ارسال داده به پروتکل لایه انتقال از طریق یک سوکت با سرعت و یا نرخ میانگین λ_{in} بایت بر ثانیه می‌باشد. این داده‌ها تنها یک مرتبه به سوکت گسیل می‌شوند و پروتکل لایه‌ی انتقال پس از دریافت آن‌ها را بدون اعمال عملیات بازیابی خطا مانند انتقال مجدد، کنترل جریان و یا کنترل ترکم بسته‌بندی می‌نماید. بدون در نظر گرفتن سرآیند اضافی مربوط به اطلاعات سرپیام لایه‌های پایین و نیز انتقال سرعتی که میزبان A از نقطه‌نظر ترافیک به مسیریاب در این حالت اعمال می‌کند و برابر λ_{in} بایت بر ثانیه است، میزبان B، نیز سرعت λ_{in} بایت بر ثانیه را اختیار می‌کند. بدیهی است یکسان فرض شدن دو سرعت میزبان جهت ساده‌سازی موضوع می‌باشد.



شکل ۳-۴۳ ♦ تراکم حالت 1: دو ارتباط به گونه‌ی مشترک از یک مسیریاب با تعداد بی‌شماری بافر استفاده کننده.

بسته‌ها از میزبان‌ها A و B با عبور از یک مسیریاب و از طریق یک پیوند خروجی مشترک با سرعت R بیت بر ثانیه به سوی مقصد گسیل می‌شوند. بافرهای موجود در مسیریاب اجازه‌ی ذخیره‌ی بسته‌های ورودی را برای شرایطی که سرعت ورود بسته‌ها بیش از توان سرعت و یا به عبارت دیگر ظرفیت پیوند خروجی است، فراهم می‌آورند. در این حالت ویژه فرض شده است تعداد بافرهای مسیریاب بی‌شمار است. شکل ۳-۴۴ نمودار **نهفتگی ارتباط** یعنی تعداد بایت‌های دریافت شده در هر ثانیه به‌وسیله گیرنده را در برابر سرعت ارسال نشان می‌دهد. برای یک نرخ ارسال بین صفر و $\frac{R}{2}$، نهفتگی در گیرنده برابر است با سرعت فرستنده به عبارت دیگر هر داده‌ی ارسالی از سوی فرستنده با یک تاخیر معین به گیرنده وارد می‌شود. هنگامی‌که یک سرعت گسیل فراتر از $\frac{R}{2}$ باشد، نهفتگی فقط $\frac{R}{2}$ می‌باشد. این محدودیت نهفتگی به علت اشتراک ظرفیت پیوند بین دو ارتباط می‌باشد. به عبارت ساده‌تر پیوند نمی‌تواند بسته‌ها را به گیرنده با سرعت پایداری که فراتر از $\frac{R}{2}$ باشد، بدون توجه به سرعت ارسال میزبان‌های A و B تحویل نماید. علت این واقعیت آن است که هر دو میزبان نهفتگی بالاتر از $\frac{R}{2}$ را به رسمیت نمی‌شناسند.



شکل ۳-۴۴ ♦ تراکم حالت نخست: نهفتگی و تأخیر به‌عنوان تابعی از سرعت ارسال میزبان

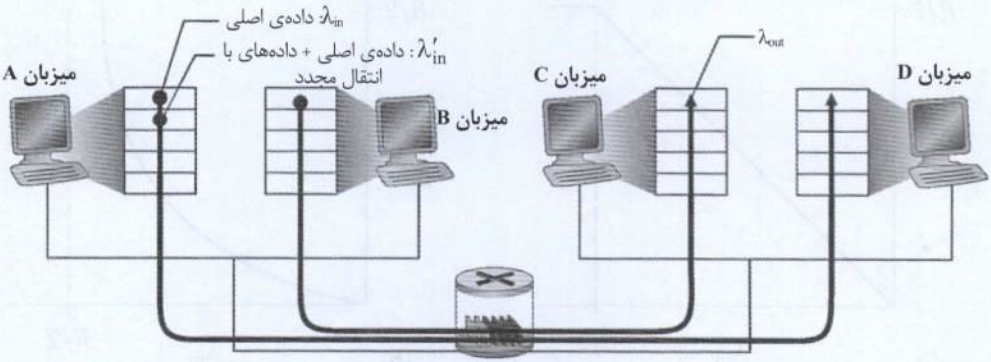
دستیابی به نهفتگی $\frac{R}{2}$ در هر ارتباط ممکن است ظاهراً مطلوب باشد، زیرا بهره‌وری از پیوند در تحویل بسته‌ها به مقصدشان در توان حداکثر میزبان ممکن می‌باشد. اما شکل ۳-۴۴ (b) تاثیر عملکرد در ظرفیت و یا سرعت معادل ظرفیت پیوند را نشان می‌دهد. هر چقدر که سرعت ارسال به نرخ $\frac{R}{2}$ نزدیک شود بر اساس شکل ۳-۴۴ (a) میانگین زمان تأخیر افزایش می‌یابد. هنگامی‌که نرخ گسیل از $\frac{R}{2}$ بزرگ‌تر می‌شود، میانگین تعداد بسته‌های به صف درآمده در مسیریاب به سمت بی‌نهایت میل می‌کند و میانگین زمان تاخیر نیز بین مبداء و مقصد دچار رشد بسیار بزرگی می‌شود. بدیهی است فرض آن است که ارتباط عمل کننده در این نرخ ارسال برای زمان نسبتاً طولانی و بهره‌گیری از امکانات ذخیره‌سازی پر حجمی برخوردار می‌باشد. بنابراین اگرچه ممکن است یک نهفتگی انباشتی در نزدیکی نرخ R مطلوب به نظر برسد اما این عملکرد از لحاظ زمان تاخیر غیر قابل قبول است. حتی در این وضعیت فوق‌العاده مطلوب یک هزینه از تراکم شبکه که عبارت است از طولانی شدن صف و در نتیجه زمان تاخیر ناشی از آن مشخص می‌شود و این هنگامی است که بسته‌ها با سرعت برابر ظرفیت پیوند وارد مسیریاب می‌شوند.

- **حالت دوم: دو فرستنده‌ی داده و یک مسیریاب با تعداد محدودی بافر**

در این وضعیت اندکی تغییرات در حالت نخست ایجاد شده است. این دگرگونی‌ها بر اساس شکل ۳-۴۵ شامل دو مورد است: (۱) حجم بافرهای مسیریاب محدود فرض شده است و (۲) هر ارتباط از لحاظ نوع آن مطمئن در نظر گرفته می‌شود.

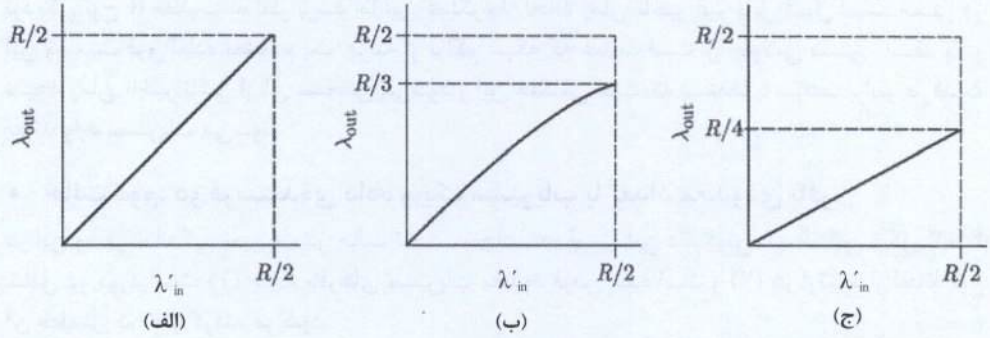
تأثیر فرض نخست آن است که در شرایط واقعی بسته‌ها به هنگام ورود در یک بافر تکمیل ظرفیت شده حذف می‌شوند و تاثیر فرض دوم آن است که اگر یک بسته حامل قطعه‌ی لایه‌ی انتقال از طریق مسیریاب حذف شده باشد، فرستنده آن بسته را مجدداً گسیل می‌نماید. از آنجائیکه بسته‌ها می‌توانند مجدداً ارسال گردند کاربرد از لحاظ سرعت ارسال نیازمند دقت بیشتر می‌باشد. اگر بار دیگر نرخ گسیل داده به درون یک سوکت λ_{in} بایت در ثانیه فرض شود آنگاه سرعت ارسال قطعه ها از سوی لایه‌ی انتقال به درون شبکه که عبارت است از داده‌های اصلی و داده‌های با وضعیت انتقال مجدد برابر است با λ'_{in} بایت بر ثانیه. عبارت λ'_{in} در واژگان شبکه **بار پیشنهادی**[1] به شبکه نامیده می‌شود.

1- Offered Load



شکل ۳-۴۵ ♦ حالت 2: دو میزبان با انتقال مجدد و یک مسیریاب با تعداد بافر محدود

کارایی حاصل در حالت دوم بستگی کامل به چگونگی انجام عملیات انتقال مجدد دارد. ابتدا یک مورد غیرواقعی بررسی می‌شود که میزبان A در آن به نوعی می‌تواند آزاد بودن یک بافر را در مسیریاب تشخیص دهد و در صورت آزاد بودن بافر بسته‌ای را گسیل نماید. در این حالت مفقودی بسته روی نمی‌دهد و λ_{in} معادل λ'_{in} خواهد شد و بنابراین نهفتگی ارتباط نیز برابر λ_{in} می‌باشد. این حالت در شکل ۳-۴۶ (الف) نشان داده شده است. از نقطه‌نظر نهفتگی کارایی مطلوب حاصل می‌شود به این مفهوم که هر آنچه ارسال می‌گردد دریافت خواهد شد. بدیهی است میانگین سرعت گسیل میزبان بیش از $\frac{R}{2}$ در این حالت نخواهد شد زیرا فرض بر آن است که مفقودی بسته هرگز روی نمی‌دهد.

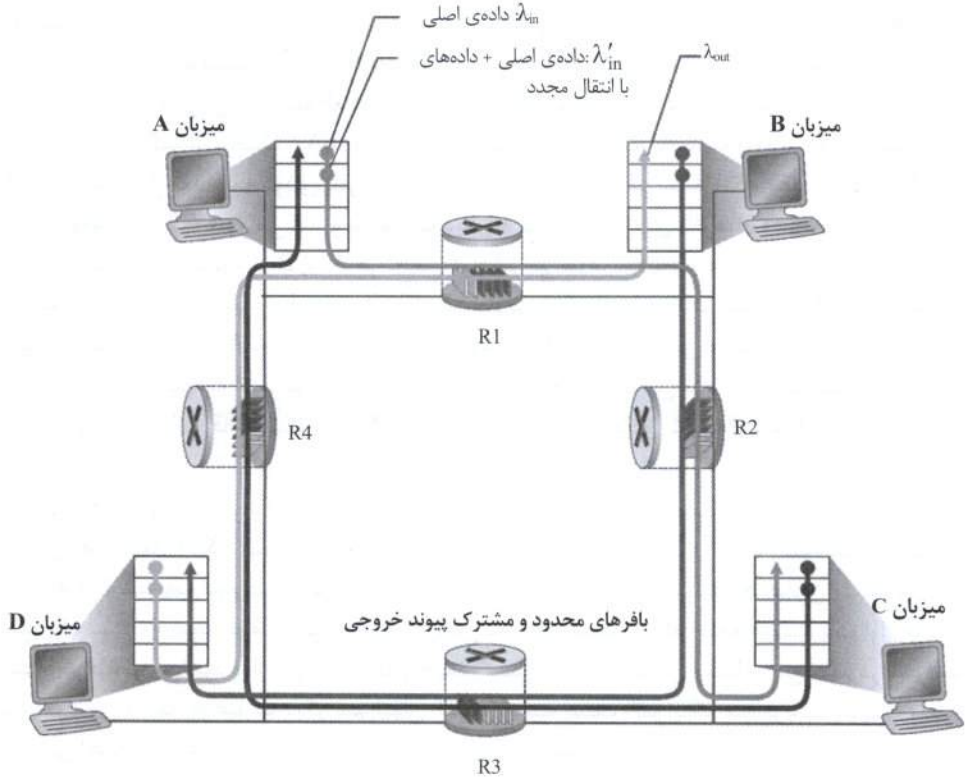


شکل ۳-۴۶ ♦ حالت دوم کارایی با تعداد بافر محدود

اکنون حالت واقعی مورد بالا را در نظر بگیرید، با فرض آنکه فرستنده هنگامی‌که کاملاً از مفقودی بسته مطمئن می‌شود نسبت به ارسال مجدد آن اقدام می‌کند. بدیهی است این فرض نیز در عمل کمی از واقعیت دور می‌باشد. اما در هر حال این واقعیت وجود دارد که میزبان فرستنده اندازه‌ی زمان خروج را به حدّ کافی طولانی در نظر گیرد تا تقریباً این اطمینان را حاصل نماید که عدم دریافت تائید بسته به مفهوم مفقود شدن آن می‌باشد. در این شرایط کارایی شبیه شکل ۳-۴۶ (ب) می‌باشد. برای درک بهتر موضوع حالتی را در نظر بگیرید که بار پیشنهادی شبکه λ'_{in} و مقدار آن نیز برابر $\frac{R}{2}$ باشد. بنابر شکل ۳-۴۶ (ب) در این ارزش بار پیشنهادی، سرعت تحویل داده‌ها به کاربردی در گیرنده برابر

$\frac{R}{3}$ خواهد شد. از بین مقدار 0.5 R داده‌ی ارسال شده، مقدار میانگین 0.333 R بایت بر ثانیه داده‌های اصلی و 0.166 R بایت بر ثانیه داده‌های مجدداً ارسال شده می‌باشند. در این حالت نیز یکی دیگر از هزینه‌های تحمیل شده در ارتباط با تراکم شبکه ملاحظه می‌شود. فرستنده بایستی نسبت به ارسال مجدد بسته‌ها جهت جبران بسته‌های مفقود شده در اثر سرریزی بافرها اقدام نماید.

سرانجام در بررسی شرایط حالت 2، وضعیتی را در نظر بگیرید که فرستنده به ارسال مجدد بسته‌ای اقدام نماید که دچار تاخیر در صف شده اما مفقود نشده است. این وضعیت اصطلاحاً **زمان خروج نارس** (غیر بالغ) خوانده می‌شود. در این شرایط هر دو بسته‌ی اصلی و انتقال مجدد شده ممکن است به گیرنده وارد شوند. بدیهی است گیرنده تنها یک نسخه از داده را لازم دارد و در نتیجه از بسته‌ی با حالت انتقال مجدد صرف‌نظر می‌نماید. کار انجام شده از سوی مسیریاب در ارسال مجدد نسخه‌ای از بسته بی‌فایده می‌باشد. به عبارت دیگر مطلوب آن است که مسیریاب از ظرفیت پیوند خود جهت ارسال بسته‌ی دیگری استفاده می نمود. در اینجا بازهم هزینه یک تراکم شبکه که عبارت است از ارسال مجدد بیهوده‌ی بسته‌ها از سوی فرستنده با توجه به تاخیر طولانی که سبب می‌شود تا مسیریاب از ظرفیت پیوند خود استفاده‌ی بی‌حاصل را نماید، مشخص می‌شود. شکل ۳-۴۶ (ج) نهفتگی در برابر بار پیشنهادی شبکه هنگامی‌که یک مسیریاب دو مرتبه یک بسته را گسیل می‌نماید، نشان داده شده است. از آنجائیکه هر بسته دو مرتبه گسیل شده است نهفتگی ارزشی همگام با مقدار $\frac{R}{4}$ را در شرایطی اختیار می‌کند که بار به سمت $\frac{R}{2}$ میل می‌کند.



شکل ۳-۴۷ ♦ چهار فرستنده و مسیریاب‌های با تعداد محدود بافر و چندین پیوند خروجی

- **حالت سوم: چهار فرستنده و یک مسیریاب با تعداد محدود بافر و چندین پیوند خروجی**

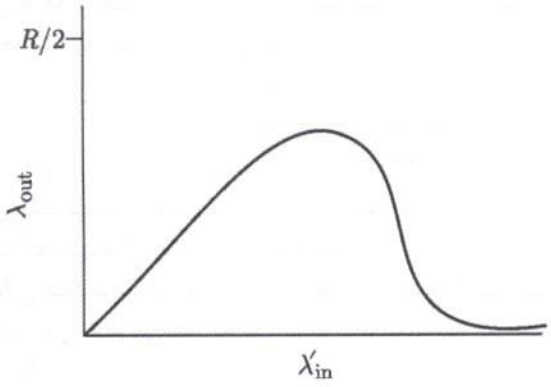
در حالت سوم تراکم، تعداد چهار میزبان به گسیل بسته‌ها می‌پردازند و هر یک از طریق اشتراک با دو مسیر پیوند خروجی بر اساس آنچه در شکل ۳-۴۷ نشان داده شده است. در این حالت نیز فرض می‌شود هر میزبان از یک روش زمان خروج و انتقال مجدد جهت پیاده‌سازی انتقال مطمئن داده‌ها استفاده می‌کند و همه‌ی میزبان‌ها دارای ارزش λ_{in} برابر با ظرفیت R بیت بر ثانیه برای هر یک از پیوندها در مسیریاب‌ها می‌باشند.

اگر ارتباط از میزبان A به C در نظر گرفته شود و این ارتباط از درون R1 و R2 عبور نماید آنگاه ارتباط A به C از مسیریاب R1 با ارتباط D به B و مسیریاب R2 با ارتباط B به D مشترکاً استفاده می‌کند. برای ارزش‌های فوق‌العاده کوچک λ_{in} سرریزی بافرهای مسیریاب‌ها به ندرت اتفاق می‌افتد و بنابراین نهفتگی تقریباً برابر است با بار پیشنهادی شبکه. برای ارزش‌های اندکی بزرگتر از λ_{in} نهفتگی بزرگتر است زیرا مقدار داده‌ی اصلی انتقال یافته به شبکه و سرانجام مقصد بیشتر است و سرریزی نیز به ندرت اتفاق می‌افتد. بنابراین برای ارزش‌های کوچک λ_{in} یک افزایش در λ_{in} سبب یک افزایش در λ_{out} می‌شود.

با در نظر گرفتن حالت ترافیک فوق‌العاده پایین اکنون شرایطی بررسی می‌شود که در آن λ_{in} و در نتیجه λ'_{in} بسیار بزرگ می‌باشند. اگر مسیریاب R2 در نظر گرفته شود ترافیک ورودی از ارتباط A به C به مسیریاب R2 که پس از عبور از R1 نتیجه می‌شود دارای حداکثر سرعت R در R2 باشد. به عبارت دیگر برابر ظرفیت پیوند از R1 به R2 بدون توجه به ارزش λ_{in}، اگر برای همه‌ی ارتباط‌ها حتی ارتباط Bبه D ارزش λ_{in} فوق‌العاده بزرگ باشد آنگاه سرعت ورود ترافیک ارتباط B به D در R2 می‌تواند بسیار بزرگتر از ترافیک Aبه C باشد. از آنجائیکه ترافیک‌های A به C و B به D در مسیریاب R2 برای استفاده از تعداد محدودی بافر، با یکدیگر مجبور به رقابت می‌شوند آنگاه مقدار ترافیک A به C که با موفقیت از R2 عبور می‌کند و به عبارت دیگر به دلیل سرریزی بافرها مفقود نمی‌شود، با افزایش بار پیشنهادی شبکه از B به A هر لحظه کوچکتر می‌شود. سر انجام هنگامی که بار پیشنهادی شبکه به سمت بی‌نهایت میل نماید هر بافر آزاد در R2 بی‌درنگ از سوی یک بسته‌ی B به D اشغال می‌شود و نهفتگی ارتباط A به C در R2 به سمت صفر متمایل می‌شود. این وضعیت به مفهوم آن است که نهفتگی ارتباط انتها به انتها در ارتباط A به C در حدّ ترافیک سنگین برابر صفر می‌شود. نتیجه‌ی حاصل از این بررسی در قالب شکل ۳-۴۸ به‌عنوان نمودار بار پیشنهادی شبکه در مقابل نهفتگی ظاهر می‌گردد.

علت کاهش نهفتگی در برابر افزایش بار پیشنهادی شبکه از نقطه‌نظر مقدار کار بیهوده‌ای که شبکه انجام می‌دهد، کاملاً آشکار است. در شرایط ترافیک سنگین که در بالا به آن اشاره رفته است هرگاه از یک بسته در مسیریاب دوم صرف‌نظر می‌شود کار انجام گرفته از سوی پیوند نخستین مسیریاب در گسیل یک بسته به پیوند دوم مسیریاب بیهوده تلقی شده در نتیجه شبکه چنانچه مسیریاب اول صرفاً از بسته صرف‌نظر می‌نمود و از ظرفیت انتقال جهت گسیل بسته‌ی دیگری استفاده می‌شد. در شرایط بهتری قرار می‌گرفت. به عبارت دیگر می‌توان گفت بهینه است اگر یک مسیریاب اولویت را به بسته‌هایی اختصاص دهد که از درون چندین مسیریاب بالا دست در شبکه عبور نموده‌اند و به آن مسیریاب وارد شده‌اند. بنابراین بار دیگر هزینه‌ی تحمیل شده به شبکه در اثر صرف‌نظر نمودن از یک بسته به دلیل تراکم آشکار می‌شود . این هزینه هنگامی است که در اثر صرف‌نظر نمودن از یک بسته

(حذف بسته) ظرفیت انتقال استفاده شده در پیوندهای بعدی تا نقطه‌ای که بسته حذف شده است بیهوده تلف می‌گردد.



شکل ۳-۴۸ ♦ حالت 3: کارایی با تعداد محدود بافرها و مسیریاب‌های چندگانه

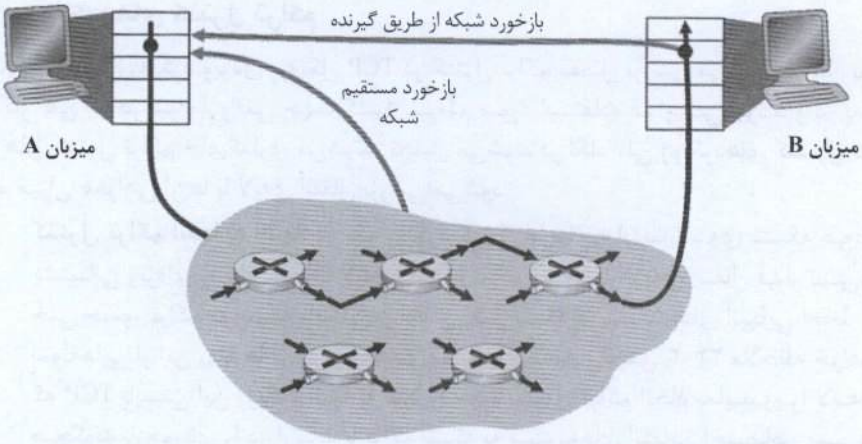
۳-۲۰ رویکردهای کنترل تراکم

در انتهای این فصل رویکرد ویژه‌ی پروتکل TCP در کنترل تراکم مفصّل بررسی می‌شود. در این بخش دو راه کار کلی که در شرایط واقعی جهت کنترل تراکم مورد استفاده قرار می‌گیرند و در بدنه‌ی پروتکل‌های کنترل تراکم جای‌گذاری می‌شوند، تحلیل می‌شود. در نگاه کلی رویکردهای کنترل تراکم از لحاظ میزان همراهی آن‌ها با لایه‌ی انتقال بررسی می‌شود:

- **کنترل تراکم انتها به انتها:** در یک رویکرد کنترل انتها به انتها لایه‌ی شبکه هیچگونه پشتیبانی ویژه‌ای را برای اهداف لایه‌ی کنترل تراکم در اختیار لایه‌ی انتقال قرار نمی‌دهد. حتی حضور تراکم در شبکه بایستی بر اساس رفتار شبکه در سیستم‌های انتهایی لحاظ گردد. نمونه‌هایی از این رفتارها تاخیر و مفقودی بسته می‌باشد. در بخش ۳-۲۴ ملاحظه خواهد شد که TCP بایستی این رویکرد انتها به انتها را جهت کنترل تراکم اتخاذ نماید زیرا لایه‌ی IP هیچگونه بازخوردی را در ارتباط با تراکم شبکه به سیستم‌های انتهایی اختصاص نمی‌دهد. مفقودی قطعه‌های TCP که از طریق زمان خروج و یا یک بسته با تائید سه مرتبه تکرار شده روی می‌دهد به‌عنوان نمادی از تراکم شبکه به‌شمار می‌آید و پروتکل TCP اندازه‌ی پنجره را بر همین اساس کاهش می‌دهد. در پژوهش‌های جدید انجام شده پیرامون موضوع کنترل تراکم، پروتکل TCP از ارزش‌های تاخیر دوره‌ی رفت و برگشت RTT فزاینده به‌عنوان نشان افزایش تراکم شبکه استفاده می‌کند.
- **کنترل تراکم به کمک شبکه:** در این رویکرد عناصر لایه‌ی شبکه مانند مسیریاب‌ها بازخوردی انحصاری و بیرونی را به فرستنده با توجه به وضعیت تراکم شبکه منتقل می‌کنند. این بازخورد می‌تواند به سادگی از طریق لحاظ فقط یک بیت که بیان کننده‌ی تراکم در یک پیوند است، انجام گیرد. این رویکرد در معماری‌های مختلف شبکه از جمله IBMSNA ، DECNET، اخیراً شبکه‌های مبتنی بر TCP/IP و در ATMABR استفاده شده است ،

به‌عنوان مثال یک راه کنترل تراکم **ATMABR**[1] ایجاد امکان برای یک مسیریاب جهت آگاهی فرستنده از نرخ انتقالی است که مسیریاب می‌تواند برای یک پیوند خروجی فراهم سازد. پروتکل XCP بازخورد محاسبه شده‌ی مسریاب را برای هر منبع داده فراهم می‌آورد که اطلاعات آن در سر پیام بسته حمل می‌شود و این امکان را برای منبع فراهم می‌آورد که چگونه نسبت به افزایش و یا کاهش سرعت انتقال اقدام نماید.

در رویکرد کنترل تراکم به کمک شبکه، اطلاعات مربوط به تراکم به کمک شبکه در اختیار فرستنده از طریق یکی از دو روش نشان داده شده در شکل ۳-۴۹ گذارده می‌شود. بازخورد مستقیم ممکن است از یک مسیریاب شبکه به فرستنده انجام گیرد. این گونه اعلان به مفهوم یک نوع ایجاد مانع برای بسته می‌باشد که مفهوم آن مشابه این جمله است که "من در حال حاضر دچار تراکم شدید هستم!" گونه‌ی دوم اعلان هنگامی است که یک مسیریاب یک دامنه را نشان‌گذاری و یا به روزرسانی می‌کند و این عمل را هنگامی انجام می‌دهد که یک بسته از فرستنده به گیرنده در جریان است، تا بیان کننده‌ی تراکم باشد. هنگامی‌که یک بسته‌ی نشان‌گذاری شده دریافت می‌شود، گیرنده اقدام به آگاه نمودن فرستنده از وضعیت تراکم می‌کند. بدیهی است زمان لازم برای انجام این دو مهم در حالت اخیر برابر یک زمان دوره رفت و برگشت RTT می‌باشد.



شکل ۳-۴۹ ♦ دو مسیر بازخورد برای اطلاعات تراکم شبکه

۳-۲۱ کنترل تراکم ATMABR

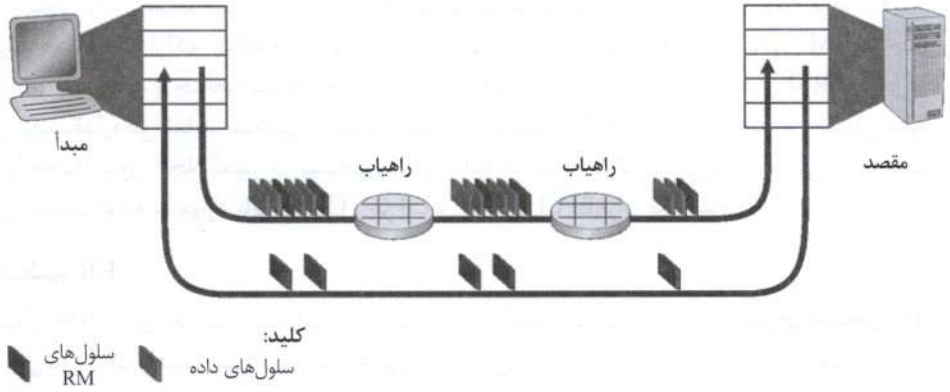
از جمله پروتکل‌های شبکه که دارای رویکرد کنترل تراکم به کمک شبکه است، پروتکل ATMABR است. در این بخش هدف بررسی شبکه‌ی ATM و معماری آن نمی‌باشد بلکه معرفی پروتکلی است که رویکردی کاملاً متفاوت را از نقطه‌نظر کنترل تراکم در مقایسه با پروتکل TCP در شبکه‌ی اینترنت اختیار می‌کند. در ذیل به بررسی آن بخشی از ATM پرداخته شده است که جهت درک عملکرد کنترل تراکم ABR مورد نیاز می‌باشد.

اصولاً ATM از رویکرد مدار مجازی VC از لحاظ راهیابی بسته استفاده می‌کند. در فصل اول توضیح داده شد که مفهوم VC به این معنی است که هر راه‌یاب در مسیر مبداء به مقصد نگهداری

1- ATM available bit rate

حالتی از مدار VC مبدا به مقصد را انجام می‌دهد که این وضعیت به راه‌یاب اجازه‌ی رهگیری رفتار هر یک از فرستنده‌ها را می‌دهد. به‌عنوان مثال از جمله‌ی این رفتارها رهگیری میانگین نرخ انتقال است. حالت دیگر مربوط است به واکنش کنترل تراکم در راه‌یاب مبداء مانند ارسال علائم لازم به فرستنده جهت کاهش سرعت هنگامی‌که راه‌یاب متراکم شده است. گزینش این حالت VC در راه‌یاب‌ها، شبکه‌ی ATM را از نقطه‌نظر کنترل تراکم به کمک شبکه بسیار مناسب می‌سازد.

پروتکل ABR از سوی دیگر به‌عنوان یک سرویس انتقال داده‌ی انعطاف‌پذیر شناخته می‌شود که رفتاری مشابه با TCP دارد. هنگامی‌که شبکه در معرض تراکم نمی‌باشد. سرویس ABR بایستی از پهنای باند آزاد در دسترس بهره‌برداری نماید و هنگامی‌که شبکه متراکم است ABR لازم است به کنترل سرعت انتقال از طریق تعریف یک حداقل نرخ انتقال از پیش تعریف شده‌ای جهت مدیریت تراکم اقدام نماید. شکل ۳-۵۰ چارچوب عملیاتی کنترل تراکم ATMABR را نشان می‌دهد. لازم به یادآوری است که بر اساس واژگان ATM از واژه‌های **راهیاب** و **سلول** به ترتیب به‌عنوان حایگزین‌های مسیریاب و بسته استفاده شده است. بنابراین در پروتکل ATMABR سلول‌های داده به کمک تعدادی از راهیاب‌های میانی از مبداء به مقصد منتقل می‌شوند و در ارتباط با سلول‌های داده از آنچه اصطلاحاً **سلول‌های مدیریت منابع**[1] (RM) نامیده می‌شوند جهت انتقال اطلاعات مربوط به تراکم شبکه در بین میزبان‌ها و راهیاب‌ها استفاده می‌شود.



شکل ۳-۵۰ ♦ چارچوب کنترل تراکم پروتکل **ATMABR**

هنگامی‌که یک سلول RM به مقصد وارد می‌شود ابتدا اطلاعات درون آن اندکی از سوی مقصد تغییر می‌کند و سپس مجدداً به سوی مبدا بازگشت داده می‌شود. این امکان نیز وجود دارد که یک راهیاب سلول‌های RM خود را تولید و این سلول‌های RM را مستقیماً به منبع گسیل نماید. بنابراین از سلول‌های RM می‌توان جهت بازخورد مستقیم شبکه و نیز از سوی گیرنده مطابق آنچه در شکل ۳-۵۰ نشان داده شده است، بهره گرفت. کنترل تراکم ATMABR یک رویکرد بر مبنای سرعت است. به عبارت دیگر فرستنده حداکثر نرخی را که قادر به ارسال سلول‌های داده می‌باشد، محاسبه می‌نماید و سپس سرعت خود را بر مبنای آن تنظیم می‌کند. پروتکل ABR سه روش را جهت سیگنالینگ اطلاعات کنترل تراکم از راهیاب‌ها به گیرنده در اختیار دارد. این سه رویکرد عبارتند از:

1- Resource management cells

- **بیت EFCI**

در این روش هر سلول داده شامل یک بیت **انحصاریِ بیانِ تراکم به جلو**[1] **(EFCI)** می‌باشد. یک راهیاب متراکم شبکه می‌تواند با مقداردهی بیت EFCI با ارزش **یک** در سلول داده میزبان مقصد را از تراکم حاصل در شبکه آگاه نماید. میزبان مقصد موظف است بیت EFCI را در همه‌ی سلول‌های داده‌ی دریافتی آزمایش نموده و چنانچه آخرین سلول داده‌ی دریافتی در مقصد دارای ارزش یک در بیت EFCI می‌باشد آنگاه مقصد، ارزش بیت نشان تراکم CI را در سلول RM برابر یک می‌نماید و سپس آن را به سوی فرستنده بازگشت می‌دهد. با بهره‌گیری از بیت EFCI در سلول‌های داده و نیز بیت CI در سلول‌های RM فرستنده می‌تواند از وضعیت تراکم در یک راهیاب شبکه آگاه گردد.

- **بیت CI و NI**

همانگونه که در مطالب قبل ملاحظه شد سلول‌های RM فرستنده به گیرنده در بین سلول‌های داده جایگذاری شده‌اند. به عبارت دیگر می‌توان این سلول‌ها را متفرق از یکدیگر به‌شمار آورد. در این حالت نرخ پراکندگی سلول‌ها می‌تواند به‌عنوان یک معیار که دارای ارزش پیش‌فرض به‌صورت **یک سلول RM** برای هر یک از تعداد 32 سلول RM در نظر گرفته می‌شود، مورد استفاده قرار گیرد. این سلول‌های RM اصطلاحاً دارای **یک بیت تراکم**[2] CI و یک بیت **عدم افزایش**[3] **NI** می‌باشند. هر دو بیت از سوی یک راهیاب شبکه در حالت تراکم می‌توانند مقداردهی شوند. در حالت ویژه یک راهیاب می‌تواند هنگامی‌که تراکم شبکه حالت عادی و در جریان را دارد، بیت NI را در سلول RM در حال عبور برابر یک نماید و چنانچه شبکه دچار تراکم شدید است آنگاه بیت CI را در سلول RM عبوری با ارزش یک مقداردهی نماید. هنگامی‌که میزبان مقصد یک سلول RM را دریافت نماید آن میزبان سلول RM را مجدداً بدون ایجاد تغییر در بیت‌های NI و CI به فرستنده بازگشت می‌دهد. بدیهی است مقصد ممکن است با توجه به موارد بالا بیت CI را بر اساس رویکرد EFCI برابر یک نماید.

- **تنظیم ER**

هر سلول RM دارای یک میدان به طول 2 بایت است که اصطلاحاً به آن میدان **نرخ آشکار**[4] **ER** گفته می‌شود. یک راهیاب متراکم در شبکه می‌تواند ارزش درون میدان ER را به هنگام عبور سلول RM کاهش دهد. در این روش میدان ER برابر حداقل نرخ قابل پشتیبانی از سوی همه‌ی راهیاب‌های در مسیر مبداء به مقصد می‌گردد.

در پروتکل ATMABR مبداء نسبت به تغییر سرعت ارسال سلول‌ها به‌عنوان تابعی از ارزش‌های CI ، NI و ER در یک سلول RM عودت داده شده مبادرت می‌کند. قواعد مربوط به تغییرات در نرخ، اغلب از پیچیدگی قابل توجهی برخوردار است.

۲۲-۳ کنترل تراکم TCP

در بخش‌های قبل توضیح داده شد چگونه پروتکل TCP انتقال مطمئن داده‌ها را بین دو فرآیند در حال اجرا در میزبان‌های گوناگون فراهم می‌آورد. یکی از عناصر کلیدی دیگری که TCP از آن بهره‌مند است

1- Explicit forward congestion indication
2- Congestion indication
3- No increase
4- Explicit rate

پیاده‌سازی کنترل تراکم ترافیک است. قبلاً اشاره گردید TCP از کنترل تراکم انتها به انتها در مقابل روش کنترل تراکم به کمک شبکه استفاده می‌کند، زیرا لایه‌ی IP بازخورد آشکاری را در اختیار سیستم‌های انتهایی کنترل تراکم قرارنمی‌دهد.

رویکرد گزینش شده از سوی TCP، هر فرستنده را موظف به محدود نمودن نرخ ارسال ترافیک به درون ارتباط به‌عنوان تابعی از تراکم حاصل در شبکه می‌کند. اگر برداشت یک فرستنده‌ی TCP از شبکه تراکم اندک در مسیر بین خویش و مقصد باشد آنگاه فرستنده‌ی TCP نرخ ارسال را افزایش می‌دهد. چنانچه برداشت فرستنده وجود تراکم در مسیر باشد آنگاه سرعت ارسال را کاهش می‌دهد. اما واقعیت آن است که این رویکرد سه پرسش اساسی را مطرح می‌نماید. این سه عبارتند از: (۱) چگونه فرستنده‌ی TCP سرعت ارسال ترافیک به درون ارتباط را محدود می‌کند؟ (۲) چگونه یک فرستنده‌ی TCP از تراکم در مسیر بین خود و مقصد آگاه می‌شود؟ (۳) از چه الگوریتمی فرستنده بایستی جهت تغییر سرعت ارسال به‌عنوان تابعی از تراکم دریافت انتها به انتها بهره گیرد؟

ابتدا پرسش نخست بررسی می‌شود که چگونه فرستنده نرخ انتقال ترافیک به درون ارتباط را محدود می‌کند. در بخش ۳-۱۱ ملاحظه شد هر سوی یک ارتباط TCP شامل یک بافر دریافت، یک بافر ارسال و چندین متغیر می‌شد. روش کنترل تراکم TCP در فرستنده رهگیری یک متغیر اضافه‌ی دیگر به نام **پنجره‌ی تراکم**[1] را نیز شامل می‌شود. پنجره‌ی تراکم با نام متغیر **cwnd** مشخص می‌شود و محدودیتی را بر سرعت فرستنده‌ی TCP از نقطه‌نظر مقدار ترافیک ارسالی به شبکه اعمال می‌نماید. به طور کلی این محدودیت در قالب این شرط است که مقدار داده‌های تائید نشده در یک فرستنده ممکن است بیش از حداقل مقدار دو متغیر cwnd و rwnd نگردد. به عبارت دیگر:

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{Min}\{\text{cwnd}, \text{rwnd}\}$$

جهت تمرکز بر روی کنترل تراکم در مقابل کنترل جریان فرض بر آن است که بافر دریافت TCP به اندازه‌ای بزرگ است که از محدودیت پنجره‌ی دریافت می‌توان صرف‌نظر نمود. بنابراین مقدار داده‌های تائید نشده در فرستنده فقط با ارزش cwnd محدود می‌شود. بدیهی است فرض دیگر آن است که فرستنده همواره، داده جهت گسیل در اختیار دارد به مفهوم دیگر همه‌ی قطعه‌های TCP موجود در پنجره‌ی تراکم گسیل می‌شوند.

اعمال رویکرد بالا سبب ایجاد محدودیت در مقدار داده‌ی تائید نشده در فرستنده می‌گردد و بنابراین به گونه‌ی غیر مستقیم سرعت ارسال فرستنده را محدود می‌نماید. جهت مشاهده‌ی این موضوع ارتباطی را در نظر بگیرید که در آن از هر دو مورد اتلاف و زمان تاخیر انتقال بسته‌ها می‌توان صرف‌نظر نمود، آنگاه تقریباً در آغاز هر RTT ، محدودیت اعمال شده این امکان را برای فرستنده ایجاد می‌کند که قادر به ارسال تعدادی cwnd بایت به درون ارتباط باشد. در انتهای زمان RTT فرستنده بسته‌های تائید داده‌ها را دریافت می‌نماید. آنچه می‌توان از مطالب عنوان شده نتیجه‌گیری نمود عبارت است از نرخ ارسال فرستنده که تقریباً برابر $\frac{\text{cwnd}}{\text{RTT}}$ بایت بر ثانیه است. از طریق تغییر ارزش cwnd فرستنده می‌تواند سرعت انتقال داده را به درون ارتباط خود کنترل نماید.

در ادامه پرسش دوم بررسی می‌گردد، چگونه یک فرستنده‌ی TCP از وجود تراکم در مسیر خود و مقصد آگاه می‌شود؟ ابتدا به تعریف **رخداد مفقود شدن**[2] می‌پردازیم که در آن یک فرستنده‌ی TCP

1- Congestion window
2- Loss event

در شرایط زمان خروج و یا دریافت سه مرتبه‌ی پیام تائید ACK از گیرنده از وقوع آن آگاه می‌شد و سرانجام با اعمال تغییراتی مانند تسریع در ارسال مجدد پس از دریافت تعداد سه ACK اقدام می‌نمود. هنگامی‌که شدت تراکم بیش از حدّ معین است آنگاه بافرهای یک یا چند مسیریاب در طول مسیر با سرریزی رو به رو می‌شوند. این وضعیت سبب می‌شود یک داده گرام حامل قطعه‌ی TCP حذف گردد. در نتیجه‌ی این اقدام رخداد مفقودی روی می‌دهد که تفسیر فرستنده از آن تراکم در مسیر بین فرستنده و گیرنده می‌باشد.

با توجه به چگونگی تشخیص تراکم، اکنون حالت بهینه‌ای را در نظر بگیرید که در آن شبکه از لحاظ تراکم آزاد می‌باشد. به عبارت دیگر رخداد مفقودی وجود ندارد. در این حالت پیام‌های ACK متعلق به قطعه‌های TCP که قبلاً به‌صورت NAK بوده‌اند از سوی فرستنده TCP دریافت می‌شود. پروتکل TCP ورود این پیام‌های ACK را به منزله‌ی عادی بودن جریان انتقال قلمداد می‌کند و بنابراین از این پیام‌های تائید جهت افزایش اندازه‌ی پنجره‌ی تراکم و در نتیجه افزایش سرعت انتقال استفاده می‌کند. بدیهی است ورود پیام‌های ACK با نرخ نسبتاً کند سبب افزایش کند پنجره‌ی تراکم می‌گردد و این حالت مربوط به زمانی است که مسیر انتها به انتها دچار تاخیر طولانی و یا کاهش پهنای باند در پیوند گردد. از طرف دیگر اگر پیام‌های ACK با شتاب بیشتر وارد شوند آنگاه اندازه‌ی پنجره‌ی تراکم سریعتر افزایش می‌یابد. از آنجائیکه TCP از ACK به‌عنوان عامل تحریک در افزایش اندازه‌ی پنجره‌ی تراکم بهره می‌گیرد بنابراین گفته می‌شود پروتکل TCP یک پروتکل خود **تحریک‌گر**[1] است.

با توجه به روش اعمال تغییر در ارزش cwnd جهت کنترل سرعت ارسال پرسش مهم دیگر عبارت است از این که چگونه یک فرستنده‌ی TCP سرعت ارسال خود را تعیین می‌کند؟ اگر فرستنده‌های TCP مجموعاً با سرعت به گسیل اطلاعات بپردازند، آن‌ها می‌توانند سبب تراکم شبکه شده و رویدادی را ایجاد نمایند که در شکل ۳-۴۸ نشان داده شده است. در ادامه مطالب همین بخش پروتکل TCP از لحاظ فروپاشی اینترنت در اثر تراکم بررسی می‌گردد. از طرف دیگر اگر پروتکل TCP گسیل قطعه را با سرعت کند انجام دهد آنگاه بهره‌وری لازم و بهینه را از پهنای باند شبکه نکرده است به عبارت دیگر فرستنده‌های TCP می‌توانسته‌اند با سرعت بالاتر و بدون ایجاد تراکم در شبکه داده‌ها را گسیل نمایند. بنابراین چگونه فرستنده‌های TCP نرخ گسیل خویش را به گونه‌ای گزینش می‌نمایند که سبب عدم تراکم شبکه گردد و در عین حال از پهنای باند در دسترس شبکه حداقل بهره را دریافت نمایند؟ آیا فرستنده‌های TCP با یکدیگر هماهنگ عمل می‌کنند و یا آنکه از یک رویکرد توزیعی بهره می‌گیرند که در آن فرستنده‌های TCP هر کدام خود در مورد سرعت ارسال بر مبنای اطلاعات محل خود اقدام می‌کنند؟ پاسخ همه‌ی این پرسش‌ها را پروتکل TCP با توجه به اصول ذیل فراهم آورده است:

۱- مفقودی قطعه به مفهوم تراکم است

هنگامی‌که قطعه‌ی TCP مفقود شده است فرستنده‌ی TCP بایستی سرعت ارسال را کاهش دهد. در بخش‌های قبل توضیح داده شد به هنگام وقوع زمان خروج و یا دریافت چهار پیام تائید برای هر قطعه که شامل یک پیام ACK اصلی و سه پیام ACK تکراری است رویداد مفقودی قلمداد شده و عملیات ارسال مجدد قطعه‌ی تلف شده آغاز می‌گردد. از نقطه‌نظر کنترل تراکم دشواری در چگونگی کاهش پنجره‌ی تراکم از سوی فرستنده‌ی TCP و در نتیجه سرعت ارسال از سوی آن با توجه به شرایط به وجود آمده می‌باشد؟

1- Self triggering

۲- دریافت پیام ACK برای یک قطعه‌ی TCP

هنگامی‌که فرستنده قطعه پیام ACK را دریافت می‌کند مفهوم آن می‌تواند تحویل قطعه‌های فرستنده از سوی شبکه به گیرنده‌ای باشد و بنابراین سرعت فرستنده می‌تواند در صورتی افزایش یابد که پیام ACK دریافتی متعلق به قطعه‌ی تائید نشده‌ی قبلی باشد. ورود چنین پیامی به مفهوم عادی بودن جریان انتقال و ترافیک در شبکه می‌باشد. به عبارت دیگر قطعه‌ها سالم از فرستنده به گیرنده تحویل می‌شوند و شبکه نیز با تراکم رو به رو نمی‌باشد بنابراین اندازه‌ی پنجره تراکم می‌تواند افزایش یابد.

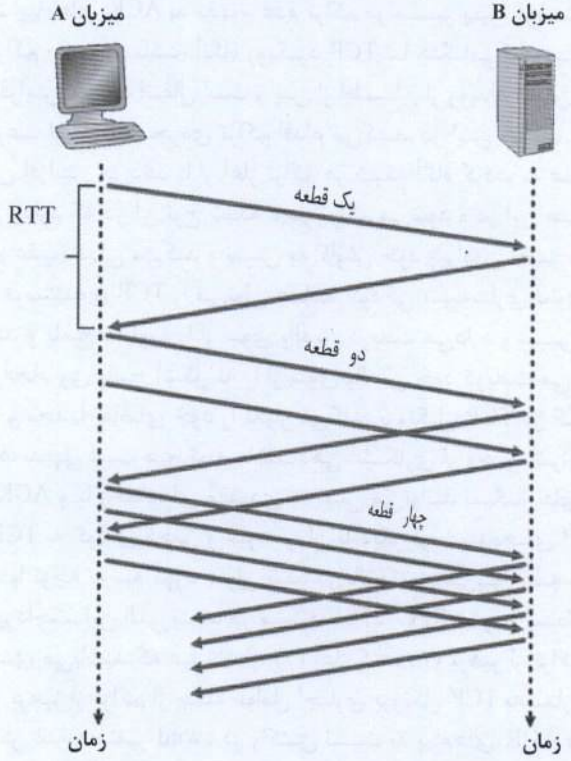
۳- کاوش پهنای باند

با فرض آنکه دریافت پیام‌های ACK به مفهوم عدم تراکم در مسیر بین فرستنده وگیرنده و رخداد مفقودی به معنی تراکم در مسیر باشد، آنگاه رویکرد TCP تا هنگامی‌که پیام‌های ACK دریافت می‌شوند مبتنی بر افزایش سرعت انتقال است و پس از اطمینان از رویداد فقدان قطعه‌ها پروتکل TCP نسبت به کاهش سرعت از طریق پنجره‌ی تراکم اقدام می‌کند. در این حالت فرستنده‌ی TCP نرخ انتقال را به آن میزان افزایش می‌دهد تا از آغاز تراکم در شبکه آگاه گردد. به عبارت دیگر به کاوش پیرامون سرعتی می‌پردازد که در آن نرخ شبکه دچار تراکم می‌شود و در این هنگام است که از سرعت پدید آورنده‌ی تراکم عقب‌نشینی می‌کند و سپس به کاوش خود پیرامون تغییر سرعت از لحاظ تراکم ادامه می‌دهد. رفتار فرستنده‌ی TCP را می‌توان همانند کودکی شبیه‌سازی نمود که تقاضایی را برای موردی مطرح می‌کند و پاسخ مطلوب را از سوی والدین دریافت می‌دارد و سپس بر میزان تقاضای خود می‌افزاید تا آنکه سرانجام وی پاسخ آشکار نَه را از سوی والدین خود دریافت می‌نماید. کودک لحظه‌ای عقب‌نشینی می‌کند و مجدداً تقاضای خود را تکرار می‌کند. پروتکل TCP در کاوش پهنای باند نیز چنین رویکردی دارد. بدیهی است هیچگونه علامت‌دهی آشکاری از وجود تراکم در شبکه ملاحظه نمی‌شود. پیام‌های ACK و یا رخدادهای مفقودی به نوعی می‌توانند سیگنال‌های روشنی قلمداد گردند که هر فرستنده‌ی TCP به گونه‌ی محلی و غیرهمزمان با دیگر فرستنده‌های TCP بر روی اطلاعات آن‌ها عمل می‌نماید. با توجه به سه مورد عنوان شده در بالا اکنون می‌توان به بررسی الگوریتم‌های کنترل تراکم TCP پرداخت. این الگوریتم‌ها در مستند RFC 2581 تشریح شده‌اند اما به طور کلی دارای سه عنصر کلیدی می‌باشند که عبارتند از: (۱) آغاز کُند، (۲) پرهیز از تراکم و (۳) بازیابی سریع. دو عنصر آغاز کُند و پرهیز از تراکم از جمله عوامل اجباری پروتکل TCP به‌شمار می‌آیند. تفاوت این دو در چگونگی افزایش اندازه متغیر cwnd در واکنش نسبت به پیام‌های ACK دریافتی می‌باشد. در ادامه ملاحظه خواهد شد که آغاز کُند علیرغم مفهوم نام آن افزایش سریع متغیر cwnd را در مقایسه با عامل خودداری از تراکم به همراه دارد. برای اغلب فرستنده‌های TCP بازیابی سریع مطلوب است اما نیاز جدی به‌شمار نمی‌آید. در ذیل موارد سه گانه‌ی بالا تشریح می‌شوند:

- **آغاز کُند**

هنگامی‌که یک ارتباط TCP آغاز می‌شود، ارزش cwnd بر اساس مستند RFC 3390 با مقدار اندک یک MSS مقداردهی می‌شود. با انجام این مقداردهی اولیه نرخ ارسال تقریباً برابر $\frac{MSS}{RTT}$ می‌شود. به‌عنوان مثال اگر MSS= 500 بایت و RTT = 200 میلی‌ثانیه باشد، سرعت اولیه‌ی ارسال فقط 20 Kbps خواهد بود. از آنجائیکه پهنای باند در دسترس فرستنده‌ی TCP ممکن است از مقدار $\frac{MSS}{RTT}$ بیشتر باشد، فرستنده‌ی TCP علاقه‌مند به یافتن سریع مقدار پهنای باند امکان‌پذیر می‌باشد. بنابراین در حالت آغاز کُند ارزش متغیر cwnd از مقدار یک MSS آغاز می‌گردد و ارزش آن هر مرتبه که یک قطعه‌ی ارسالی

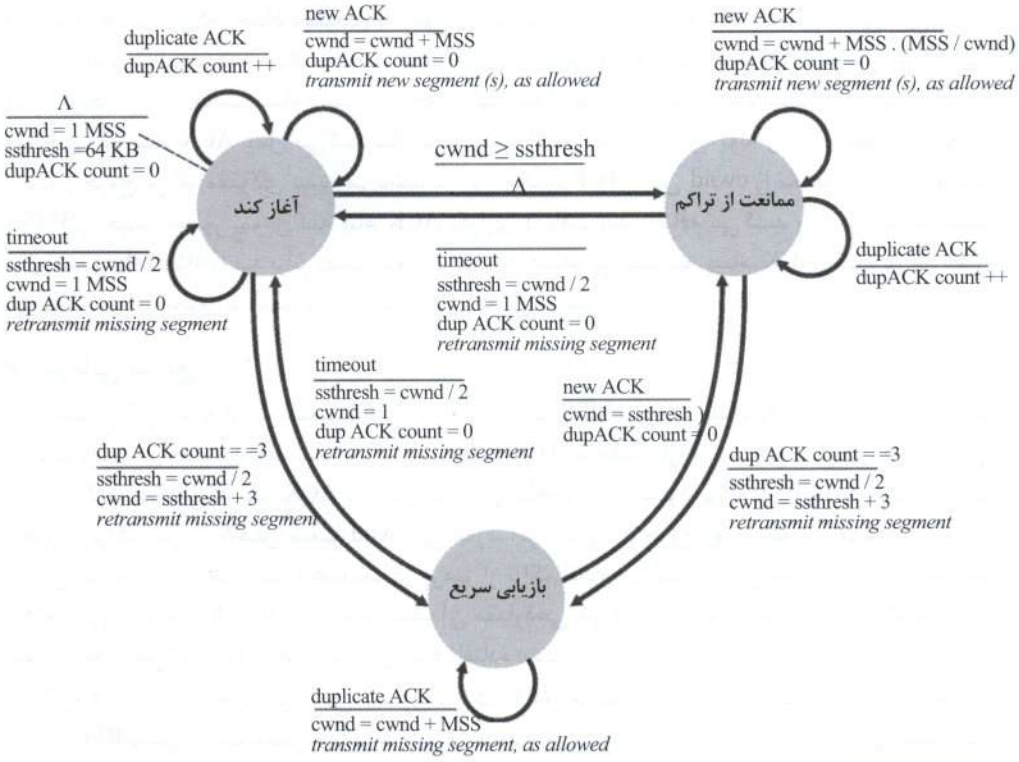
بار نخست تائید می‌گردد به اندازه‌ی یک MSS افزایش می‌یابد. در مثال شکل ۳-۵۱ پروتکل TCP نخستین قطعه را به درون شبکه گسیل می‌کند و در انتظار یک تائید می‌ماند. هنگامی‌که این تائید وارد می‌شود، فرستنده‌ی TCP مقدار پنجره‌ی تراکم را به اندازه‌ی یک MSS افزایش داده و دو قطعه‌ی TCP با حداکثر اندازه‌ی طولی را گسیل می‌نماید. این قطعه‌ها سپس تائید شده و در نتیجه فرستنده پنجره‌ی تراکم را به اندازه‌ی یک MSS برای هر یک از قطعه‌ها تائید شده افزایش می‌دهد، در نتیجه پنجره‌ی تراکم برابر 4 MSS و ارزش‌های بعدی آن به همین طریق افزایش می‌یابد. نتیجه‌ی حاصل از این فرآیند افزایش دو برابری نرخ ارسال برای هر RTT می‌باشد. بنابراین آغاز نرخ ارسال TCP در ابتدا ظاهراً کُند می‌باشد اما در طول عملیات فرآیند آغاز کُند با شتاب تصاعدی افزایش می‌یابد.



شکل ۳-۵۱ ♦ فرآیند آغاز کُند پروتکل TCP

اکنون این پرسش مطرح می‌شود که این رشد تصاعدی چه زمانی پایان می‌گیرد؟ فرآیند آغاز کُند چندین پاسخ را برای این پرسش فراهم می‌آورد. نخست، اگر رخداد مفقودی یا به عبارت دیگر تراکم از طریق یک زمان خروج مشخص گردد آنگاه فرستنده‌ی TCP ارزش متغیر cwnd را برابر یک می‌کند و فرآیند جدید آغاز کُند را فعال می‌نماید. این فرآیند آغاز کُند جدید، هم چنین ارزشی را برای متغیر حالت دوم به نام ssthresh که اصطلاحاً متغیر **آستانه‌ی آغاز کُند** نامیده است، تعریف می‌کند که ارزش آن برابر $\frac{\text{cwnd}}{2}$ می‌باشد. به عبارت دیگر نصف ارزش پنجره‌ی تراکم به هنگام تشخیص و آگاهی از وقوع تراکم. روش دوم که از طریق آن می‌توان فرآیند آغاز کند را به پایان رسانید می‌تواند از طریق وابستگی مستقیم آن با ارزش متغیر ssthresh باشد. از آنجائیکه متغیر ssthresh نیمی از ارزش cwnd را از هنگام آخرین مرتبه دو

برابر نمود به‌ویژه اگر این ارزش به مقدار آستانه‌ی آغاز کند نزدیک و یا حتی از آن عبور نماید. بنابراین هنگامی‌که ارزش متغیر cwnd برابر ssthresh می‌شود فرآیند آغاز کند پایان می‌پذیرد و پروتکل TCP از حالت انتقال به وضعیت خودداری از تراکم مهاجرت می‌کند. در ادامه ملاحظه خواهد شد که TCP افزایش مقدار cwnd را با احتیاط فراوان انجام می‌دهد به‌ویژه هنگامی‌که در فرآیند پرهیز از تراکم قرار می‌گیرد. سرانجام روش آخر است که در آن فرآیند آغاز کند را می‌توان خاتمه داد این مهم در شرایط دریافت چهار پیام ACK که عبارت است از ACK اصلی به همراه سه مرتبه تکرار نسخه‌ی همان ACK می‌باشد. قبلاً اشاره گردید در این حالت TCP عملیات انتقال مجدد پرشتاب را انجام می‌دهد و پس از آن وارد مرحله بازیابی سریع می‌شود. رفتار TCP در قالب یک ماشین FSM برای حالت آغاز کند در شکل ۳-۵۲ نشان داده شده است.



شکل ۳-۵۲ ♦ ماشین FSM کنترل تراکم پروتکل TCP

- **پرهیز از تراکم**

هنگام ورود به حالت پرهیز از تراکم ارزش متغیر cwnd تقریباً نصف مقدار آن قبل از آخرین مرتبه‌ای است که تراکم روی داده است. از آنجائیکه امکان وقوع تراکم هر لحظه امکان‌پذیر است بنابراین پروتکل TCP به جای آنکه در هر RTT ارزش متغیر cwnd را دو برابر نماید از یک رویکرد محافظه‌کارانه بهره می‌جوید و ارزش cwnd را تنها به اندازه‌ی یک MSS برای هر RTT بر مبنای مستند RFC 2581 افزایش می‌دهد. انجام این مهم از چندین روش امکان‌پذیر است. یکی از متداولترین آن‌ها افزایش مقدار cwnd به اندازه‌ی بایت‌های MSS است که از سوی فرستنده‌ی TCP هنگامی‌که یک پیام ACK جدید

وارد می‌شود، انجام می‌پذیرد. این روش افزایش به گونه‌ی $\frac{MSS}{cwnd}$ است، به‌عنوان مثال اگر مقدار MSS برابر 1460 بایت است و cwnd نیز معادل 14600 بایت، آنگاه در بین هر زمان RTT تعداد قطعه‌های ارسالی TCP برابر است با 10 قطعه. اگر برای هر قطعه یک پیام ACK در نظر گرفته شود آنگاه با ورود هر پیام ACK اندازه‌ی پنجره‌ی تراکم به مقیاس MSS $\frac{1}{10}$ افزایش می‌یابد. به عبارت دیگر ارزش پنجره‌ی تراکم به اندازه‌ی یک MSS پس از آن که پیام‌های ACK متعلق به همه‌ی 10 قطعه دریافت گردید، افزایش می‌یابد.

اما چه هنگام این افزایش خطی یک MSS در هر RTT در روش پرهیز از تراکم بایستی خاتمه یابد؟ رفتار الگوریتم پرهیز از تراکم TCP مشابه هنگامی است که زمان خروج پروتکل TCP روی می‌دهد. مانند حالت آغاز کند: ارزش cwnd برابر یک MSS است و ارزش متغیر آستانه‌ی آغاز کند ssthresh هر مرتبه که رخداد مفقودی به وقوع می‌پیوندد به اندازه‌ی نصف cwnd مقداردهی می‌شود. لازم به یادآوری است که رخداد مفقودی ممکن است از طریق دریافت سه پیام ACK تکرار شده نیز فعال گردد. در این حالت شبکه نسبت به تحویل قطعه‌ها از فرستنده به گیرنده مشابه حالت دریافت دوباره‌ی یک پیام ACK عمل می‌کند. بنابراین واکنش TCP نسبت به این گونه رخداد مفقودی رقیق‌تر از زمان خروج در اثر مفقودی بسته می‌باشد. در این حالت TCP ارزش cwnd را نصف کرده و تعداد 3MSS را جهت منظور نمودن سه پیام ACK تکراری دریافت شده اضافه می‌کند و ارزش ثبت شده برای متغیر ssthresh را نیز برابر نصف متغیر cwnd به هنگام دریافت سه پیام ACK تکراری در نظر می‌گیرد. پس از انجام این مهم وارد حالت بازیابی سریع می‌شود.

- **بازیابی سریع**

در حالت بازیابی سریع ارزش متغیر cwnd به اندازه‌ی یک MSS برای هر پیام ACK تکراری دریافت شده به واسطه‌ی قطعه‌ی تلف شده‌ای که سبب ورود TCP به حالت بازیابی سریع می‌گردد، افزایش پیدا می‌کند. هنگامی‌که یک ACK به دلیل مفقودی قطعه وارد می‌شود پروتکل TCP گام در حالت پرهیز از تراکم پس از کاهش متغیر cwnd می‌گذارد. اگر یک زمان خروج روی دهد مهاجرت از وضعیت بازیابی سریع به آغاز کند مشابه عملیات در پرهیز از تراکم انجام می‌گیرد. ارزش متغیر cwnd برابر یک MSS و ارزش متغیر ssthresh نیز برابر نصف آن مقداردهی می‌گردد. بدیهی است این ارزش‌گذاری در صورتی انجام می‌گیرد که رخداد مفقودی اتفاق افتاده است.

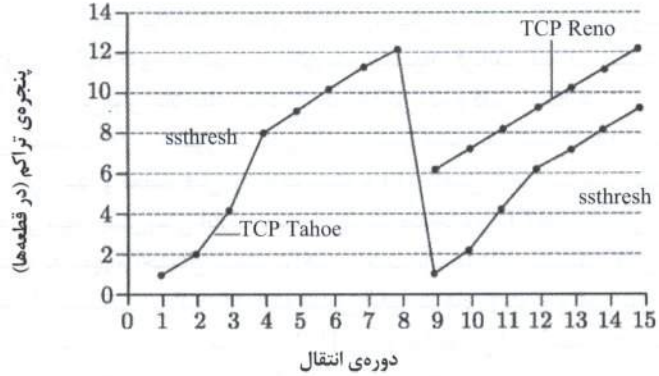
اگرچه بازیابی سریع در پروتکل کنترل تراکم TCP تعریف شده است اما بر اساس مستند RFC 2581 به‌عنوان یک عنصر حیاتی و ضروری قلمداد نمی‌شود. یادآوری این نکته ضروری است که نسخه‌های ابتدایی TCP که اصطلاحاً **تاهُو**[1] نامیده می‌شد بدون پیش شرط اندازه‌ی پنجره تراکم را برابر یک MSS می‌نمود و پس از آن با دریافت زمان خروج و یا سه پیام ACK تکراری وارد مرحله‌ی آغاز کند می‌گردید. نسخه‌ی جدیدتر TCP اصطلاحاً **رِینُو**[2] نامیده شده است و در درون خود بازیابی سریع را لحاظ نموده است.

شکل ۳-۵۳ مراحل تکامل پنجره‌ی تراکم TCP را برای هر دو نسخه‌ی TCP نشان می‌دهد. در این شکل مقداردهی اولیه آستانه برابر 8MSS می‌باشد. برای 8 دوره‌ی انتقال نخستین هر دو نسخه‌ی TCP رویکرد عملیاتی مشابه دارند. اندازه‌ی پنجره‌ی تراکم با سرعت به‌صورت تصاعدی در خلال آغاز کند

1- TCP Tahoe
2- TCP Reno

افزایش می‌یابد و در دور چهارم انتقال به مقدار آستانه در متغیر ssthresh می‌رسد. پس از آن پنجره‌ی تراکم دارای یک افزایش خطی است تا آنکه دریافت تعداد سه پیام ACK که در پایان دور هشتم انتقال روی می‌دهد. لازم به توضیح است که پنجره‌ی تراکم هنگامی‌که رخداد مفقودی ایجاد می‌شود برابر 12 MSS است. ارزش متغیر آستانه‌ی آغاز کند در این هنگام با مقدار cwnd =6 MSS مقداردهی شده و رشد آن نیز به گونه‌ی خطی می‌باشد. در شرایط مربوط به TCP نسخه‌ی تاهو اندازه پنجره‌ی تراکم با ارزش یک MSS مقداردهی شده و رشد آن تا رسیدن به ارزش ssthresh به گونه‌ی تصاعدی انجام می‌شود و پس از آن نقطه‌ی رشد خطی انجام می‌شود.

شکل ۳-۵۲ به توصیف ماشین FSM الگوریتم‌های پنجره کنترل تراکم جهت سه رویکرد آغاز کند، پرهیز از تراکم و بازیابی سریع می‌پردازد. از شکل می‌توان مکان انتقال قطعه‌های جدید و یا مجدداً ارسال شده را ملاحظه نمود. اگرچه تمایز بین دو حالت کنترل خطا و در نتیجه انتقال مجدد با کنترل تراکم در پروتکل TCP دارای اهمیت بالایی می‌باشد اما به همان اندازه نیز پیوند درونی بین این دو حالت TCP نیز مهم است.



شکل ۳-۵۳ ♦ تکامل پنجره‌ی تراکم TCP برای هر دو نسخه‌ی قدیم و جدید TCP

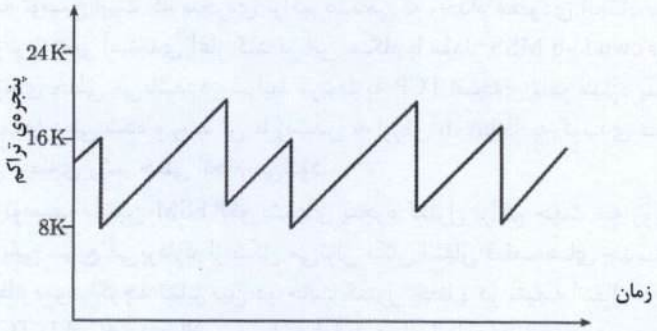
• کنترل تراکم TCP : افزایش انباشتی و کاهش افزایشی

پس از بررسی عمیق جزئیات آغاز کند، پرهیز از تراکم و بازیابی سریع اکنون زمان آن رسیده است که گامی به عقب گذارده و جنگل را از درختان آن ملاحظه کنیم. اگر هنگام آغاز ارتباط TCP از فرآیند آغاز کند صرف‌نظر نمائیم و اتلاف بسته‌ها را فقط در حالتی در نظر بگیریم که پیام‌های تکرار شده‌ی ACK سه مرتبه دریافت شده باشند و اثر زمان خروج نیز منظور نگردد آنگاه کنترل تراکم TCP شامل افزایش خطی و یا به عبارت دیگر انباشتی متغیر cwnd به مقدار یک MSS برای هر RTT و سپس نصف آن یعنی کاهش افزایش برای چنین رخدادی می‌شود. به همین منظور کنترل تراکم TCP اصطلاحاً به **افزایش انباشتی و کاهش افزایشی** [1] (AIMD) معروف شده است.

کنترل تراکم AIMD رفتاری مشابه آنچه در شکل ۳-۵۴ نشان داده شده است را دارد. این نمودار نشان می‌دهد چگونه TCP به جستجوی پهنای باند می‌پردازد. اندازه‌ی پنجره‌ی تراکم TCP و در نتیجه سرعت انتقال آن به گونه‌ی خطی افزایش یافته تا یک رخداد دریافت سه پیام تکراری ACK روی دهد.

1- Additive increase multiplicative decrease (AIMD)

پس از آن اندازه‌ی پنجره با ضریب 2 کاهش می‌یابد اما بار دیگر آغاز به افزایش خطی نموده تا پهنای باند در دسترس افزون‌تر را ارزیابی نماید.



شکل ۳-۵۴ ♦ کنترل تراکم AIMD

قبلاً اشاره گردید اغلب پیاده‌سازی‌های TCP در حال حاضر از الگوریتم رینو بهره می‌گیرند. نسخه‌های فراوانی از گونه‌های مختلف این الگوریتم پیشنهاد گردیده که در مستندهای RFC 3782 و RFC 2018 آن‌ها را می‌توان ملاحظه نمود. از جمله الگوریتم وگاس است که تلاش در پرهیز از تراکم با حفظ نهفتگی مطلوب دارد. اصول بنیادی الگوریتم وگاس عبارتند از: (۱) آشکارسازی تراکم در مسیریاب‌های بین مبدا و مقصد قبل از اتلاف بسته و (۲) کاهش خطی سرعت هنگامی‌که اتلاف بسته تشخیص داده می‌شود. پیش‌بینی اتلاف بسته از طریق مشاهده‌ی RTT انجام می‌شود. طولانی شدن زمان RTT بسته به مفهوم احتمال بیشتر تراکم در مسیریاب‌ها دارد. سیستم عامل لینوکس تعدادی از این الگوریتم‌ها را پشتیبانی می‌کند که البته شامل الگوریتم‌های رینو و وگاس نیز می‌باشد. این سیستم عامل امکانات لازم برای پیکربندی نسخه‌ی TCP را نیز فراهم می‌آورد. نسخه‌ی پیش فرض TCP در سیستم عامل لینوکس 2.6.18 می‌باشد که ویژه‌ی کاربردهای پهنای باند بالا است.

الگوریتم AIMD ویژه‌ی TCP بر مبنای تجارب فراوان بدست آمده از تحلیل کنترل تراکم در TCP حاصل شده است. تقریباً یک دهه پس از توسعه‌ی TCP تحلیل‌های نظری نشان داد که الگوریتم کنترل تراکم TCP می‌تواند به‌عنوان یک الگوریتم بهینه‌سازی توزیع غیر همزمان مورد بهره‌برداری قرار گیرد، نتیجه‌ی آن افزایش کارایی همزمان شبکه و کاربران است.

با توجه به رفتار نشان داده شده از TCP در شکل ۳-۵۴ طبیعی است که میانگین نهفتگی با نرخ حیات تراکم TCP از لحاظ تحلیل سودمند به نظر رسد. در این تحلیل از مراحل آغاز کند که پس از رویداد زمان خروج اتفاق می‌افتد صرف‌نظر می‌شود. این رویدادها اغلب از لحاظ زمان کوتاه می‌باشند زیرا فرستنده با نرخ تصاعدی سریع‌تر از فاز خود خارج می‌گردد. در طول یک تناوب RTT جاری. هنگامی‌که اندازه‌ی پنجره w بایت و زمان دوره‌ی رفت و برگشت جاری نیز RTT ثانیه باشد، سرعت انتقال TCP تقریباً $\frac{W}{RTT}$ است. پس از آن TCP به دنبال پهنای باند اضافه با افزایش ارزش w به مقدار یک MSS برای هر RTT تا زمان رخداد مفقودی می‌باشد. هنگامی‌که این حالت فرا می‌رسد ارزش w با مقدار W بیان می‌شود. اگر فرض شود مقادیر RTT و W هر دو تقریباً در طول ارتباط ثابت می‌باشند آنگاه نرخ انتقال دارای محدوده‌ای از $\frac{W}{2RTT}$ به $\frac{W}{RTT}$ می‌شود.

این مفروضات امکان تعریف ساده‌تری را برای رفتار حالت پایدار TCP فراهم می‌آورد. شبکه

هنگامی به حذف یک بسته در یک ارتباط می‌پردازد که سرعت به میزان $\frac{W}{RTT}$ افزایش پیدا کند، پس از آن مقدار سرعت به نصف آن کاهش و سپس برای هر RTT به اندازه $\frac{MSS}{RTT}$ افزایش می‌یابد تا مجدداً سرعت برابر $\frac{W}{RTT}$ شود. این فرآیند به‌صورت یک حلقه‌ی پایان ناپذیر ادامه می‌یابد. از آنجائیکه نهفتگی ارتباط نرخ افزایش خطی بین دو ارزش فاصله‌دار می‌باشد. آنگاه میانگین نهفتگی ارتباط را می‌توان به گونه‌ی ذیل تعریف نمود:

$$\text{میانگین نهفتگی ارتباط} = \frac{0.75W}{RTT} \qquad (۷-۳)$$

یادآوری این نکته ضروری به نظر می‌رسد که پروتکل‌های کنترل تراکم TCP در یک دوره‌ی چند ساله تکامل یافته‌اند و یقیناً به این روند تکامل خود نیز ادامه خواهند داد. خلاصه‌ای از این سیر تکاملی را می‌توان از دهه 1990 میلادی به بعد در RFC 2581 جستجو نمود. بدیهی است آنچه برای شبکه‌ی اینترنت هنگامی‌که ارتباط‌های TCP محدود به حمل داده‌های SMTP، انتقال فایل FTP و یا ترافیک تلنت می‌شدند، سودمند بود، برای اینترنت امروزی کافی و مفید نخواهد بود. زیرا در حال حاضر اینترنت به شدت تحت تاثیر ترافیک ناشی از HTTP می‌باشد و قطعاً در آینده با ترافیک‌های داده‌هایی رو به رو خواهد بود که هنوز در ذهن‌ها تصور آن ایجاد نشده است.

نیاز برای تکامل پیوسته‌ی TCP را می‌توان در ارتباط‌های TCP پر سرعت برای کاربرد **پردازش‌های آرایه‌ای یا ماتریسی**[1] جستجو نمود. به‌عنوان مثال یک ارتباط TCP با تعداد 1500 قطعه بایت و RTT برابر 100 میلی‌ثانیه که لازم است داده‌ای را از درون یک پیوند با سرعت 10 Gbpsعبور دهد در نظر بگیرید. با استفاده از رابطه‌ی اشاره‌شده در بالا برای دستیابی به نهفتگی 10 Gbpsمیانگین اندازه‌ی پنجره‌ی تراکم مورد نیاز برابر 83333 قطعه می‌باشد. این تعداد قطعه بسیار زیاد است زیرا اگر یکی از این قطعه‌های عبوری در اثر عوامل گوناگون مفقود گردد آنگاه می‌توان بنابر مطالب گذشته تصور نمود که چه اتفاق شگفت‌انگیز و در عین حال ناگواری رخ می‌دهد. به عبارت دیگر مقدار درصد قطعه‌های انتقال یافته که می‌تواند تلف گردد و این امکان را برای الگوریتم کنترل تراکم TCP شکل ۳-۵۲ فراهم آورد که سرعت مطلوب 10 Gbps را حفظ نماید چه مقدار است؟ اگر رابطه‌ی بین نهفتگی ارتباط TCP به‌عنوان تابعی از نرخ مفقودی (L)، زمان RTT و حداکثر اندازه‌ی قطعه MSS در نظر گرفته شود می‌توان اثبات نمود که میانگین نهفتگی یک ارتباط TCP در این حالت برابر است با:

$$\text{میانگین نهفتگی ارتباط} = \frac{1.22MSS}{RTT\sqrt{L}} \qquad (۸-۳)$$

با استفاده از این رابطه می‌توان محاسبه نمود که برای نهفتگی 10Gbps الگوریتم کنترل تراکم TCP در حال حاضر (امروز) فقط می‌تواند احتمال اتلاف قطعه‌ای را معادل 2×10^{-10} تحمل نماید. به عبارت دیگر، یک مفقودی بسته برای هر 5×10^{9} (5,000,000,000) قطعه بایت TCP که مفهوم آن نرخ بسیار پایین است. این ملاحظات سبب شده است تا پژوهشگران این عرصه به دنبال نسخه‌های جدیدتر TCP باشند که در محیط‌های با سرعت بالا می‌تواند کاربرد داشته باشد. در مستند RFC 3649 مشروح این تلاش‌ها را می‌توان یافت.

1- Grid Computing Applications

- **تجزیه‌ی TCP و بهینه‌سازی بهره‌وری خدمات در سیستم‌های رایانش ابری**

برای خدمات ابری مانند جست‌وجو، ایمیل و شبکه‌های اجتماعی امکان مطلوب تأمین یک سطح پیشرفته و وسیع برای پاسخگویی می‌باشد تا برای کاربران این احساس فراهم آید که خدمات موردنظر آنها روی میزبان‌ها یا سیستم‌های انتهایی متعلق به آنها در حال اجرا می‌باشد. بدیهی است این سیستم‌ها شامل گوشی‌های همراه هوشمند و تبلت‌ها نیز می‌شود. این مهم می‌تواند تبدیل به یک چالش بزرگ شود زیرا در اغلب موارد کاربران در مکان‌های به مراتب دورتری از مراکز داده واقع شده‌اند که وظیفه‌ی خدمات‌رسانی پویای محتواها را در ارتباط با خدمات ابری برعهده دارند. یقیناً هنگامی‌که یک میزبان در فاصله‌ی دوری از یک مرکز داده قرار می‌گیرد نتیجه‌ی حاصل طولانی‌تر شدن زمان RTT است که سبب یک زمان پاسخ بسیار ضعیف و آغاز کند TCP می‌شود.

به‌عنوان یک مورد مطالعاتی تأخیر حاصل از دریافت یک پاسخ برای یک تقاضای جست‌وجو رادر یک موتور جست‌وجو درنظر بگیرید. در وضعیت معمولی یک سرور نیازمند سه پنجره‌ی TCP در هنگام شروع کُند برای تحویل پاسخ‌های یک تقاضا می‌باشد. بنابراین زمان لازم از هنگامی که یک سیستم انتهایی آغازکننده‌ی ارتباط TCP می‌شود تا هنگامی‌که آخرین بسته‌ی مورد نیاز از پاسخ را دریافت می‌دارد تقریباً برابر 4RTT خواهد شد. این ارزش شامل یک RTT برای ایجاد ارتباط TCP و تعداد سه RTT برای سه پنجره‌ی داده‌ی مورد نیاز می‌شود. بدیهی است به زمان 4RTT مدت زمان پردازش در مرکز داده را نیز بایستی اضافه نمود. این تأخیرهای RTT می‌تواند سبب یک تأخیر قابل ملاحظه در بازگشت نتایج جست‌وجو برای بخش نسبتاً کوچکی از تقاضاها باشد. علاوه بر این موارد اتلاف قابل ملاحظه‌ی بسته‌ها در شبکه‌های دستیابی نیز دور از انتظار نمی‌باشد و این موضوع خود باعث عملیات انتقال مجدد TCP و تأخیرهای به مراتب بیشتر می‌شود. یک روش به‌منظور تسکین این دشواری بهینه‌سازی کارایی کاربر با توجه به موارد ذیل است:

(۱) به‌کارگیری سرورهای انتهایی در مجاورت کاربران و

(۲) بهره‌گیری از امکان **تجزیه‌ی TCP**[1] از طریق تقسیم ارتباط TCP در سرورهای انتهایی.

بهره‌گیری از روش تجزیه‌ی TCP باعث می‌شود تا برنامه‌ی مشتری یک ارتباط TCP را در نزدیکی سرور انتهایی انجام داده و پس از آن سرور انتهایی یک ارتباط TCP ماندگار را به مرکز داده با کمک تعداد انبوهی از پنجره‌های تراکم TCP نگهداری نماید. در این رویکرد زمان پاسخگویی تقریباً برابر $4RTT_{FE} + RTT_{BE}$ می‌شود که بایستی به این مجموع زمان پردازش را نیز اضافه نمود. در این رابطه RTT_{FE} زمان رفت و برگشت بین برنامه‌ی مشتری و سرور انتهایی و RTT_{BE} زمان رفت و برگشت بین سرور انتهایی و مرکز داده می‌باشد. در صورتی که سرور انتهایی به مشتری نزدیک باشد آنگاه این زمان پاسخگویی تقریباً برابر RTT به اضافه‌ی زمان پردازش می‌شود زیرا RTT_{FE} بسیار کوچک است و RTT_{BE} نیز تقریباً برابر RTT است. به‌صورت خلاصه تجزیه‌ی TCP می‌تواند تأخیر شبکه را از تقریباً 4RTT به‌تنها یک RTT تقلیل دهد که خود باعث بهبود چشمگیر کارایی کاربران می‌شود به‌ویژه برای آنهایی که از نزدیک‌ترین مرکز داده فاصله‌ی چشمگیری دارند. راهکار تجزیه‌ی TCP باعث می‌شود تا تأخیرهای انتقال مجدد TCP که به دلیل اتلاف در شبکه‌های دستیابی نتیجه می‌شود نیز به‌صورت محسوسی کاهش یابد. در حال حاضر شرکت‌های گوگل و Akamai بهره‌گیری گسترده‌ای را از خدمات

1- TCP Spilitting

CDN خود در شبکه‌ها دستیابی می‌کنند تا تجزیه‌ی TCP را برای خدمات ابری که این شرکت‌ها پشتیبانی آنها را برعهده دارند، میسر سازند.

۳-۲۳ تعادل در ارتباط‌های TCP

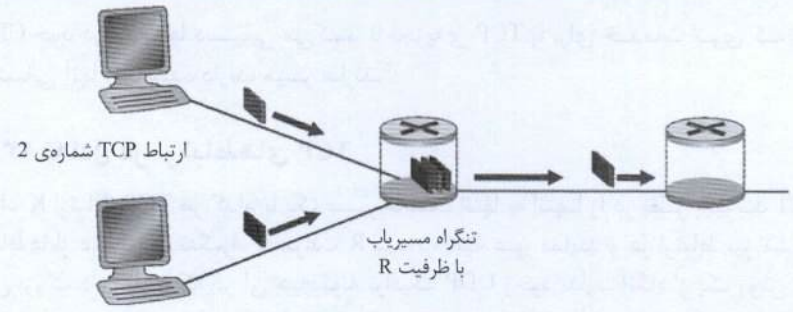
تعداد K ارتباط TCP هر کدام با یک مسیر مختلف انتها به انتها را در نظر بگیرید. اگر همه‌ی این ارتباط‌ها از درون یک تنگ‌راه با سرعت R بیت بر ثانیه عبور نمایند و هر ارتباط نیز مشغول انتقال یک فایل بزرگ داده باشد که در آن هیچگونه ترافیک UDP وجود ندارد. آنگاه از یک روش کنترل تراکم در آن ارتباط‌ها استفاده می‌شود که اصطلاحاً گفته می‌شود **عادلانه**[1] است. قبل از ادامه مطلب ابتدا لازم است دو عبارت که در این بخش از آن‌ها استفاده شده است تعریف گردد. این دو عبارتند از : (۱) پیوند تنگ‌راه و (۲) عادلانه. مفهوم **پیوند تنگ‌راه**[2] آن است که برای هر ارتباط همه‌ی پیوندهای دیگر موجود در مسیر ارتباط دچار تراکم نمی‌باشند و همگی آن‌ها دارای یک ظرفیت آزاد جهت انتقال در مقایسه با ظرفیت انتقال پیوند تنگ‌راه می‌باشند. منظور از عبارت **عادلانه** برای پیوند آن است که نرخ میانگین انتقال از هر ارتباط تقریباً برابر $\frac{R}{K}$ می‌باشد، به عبارت دیگر هر ارتباط دارای سهم برابر از پهنای باند پیوند می‌باشد.

اکنون این پرسش مطرح می‌شود که آیا الگوریتم AIMD می‌تواند عادلانه قلمداد گردد در حالیکه ارتباط‌های گوناگون TCP ممکن است در زمان‌های متفاوت آغاز گردند و بنابراین از اندازه‌ی پنجره‌های مختلف برخوردار باشند؟ پاسخ این پرسش می‌تواند نوری در تاریکی به‌شمار آید که توضیح می‌دهد چرا کنترل تراکم TCP از رویکرد اختصاص سهم برابر پهنای باند از یک پیوند تنگ‌راه در بین ارتباط‌های TCP در رقابت با یکدیگر استفاده می‌کند.

حالت ساده‌ای را در نظر بگیرید که در آن دو ارتباط TCP از یک پیوند انتقال مشترک با سرعت Rbps استفاده می‌کنند. شکل ۳-۵۵ این موضوع را نشان می‌دهد. اگر فرض شود هر دو ارتباط دارای MSS و RTT مشابه می‌باشند به عبارت دیگر اگر هر دو دارای پنجره‌های تراکم با اندازه مشابه می‌باشند آنگاه نهفتگی مشابه یکدیگر را نیز دارند. اگر داده‌ی مورد نظر جهت انتقال نیز بزرگ باشد و ارتباط‌های TCP و یا UDP دیگری از این پیوند مشترک استفاده نمی‌کنند و علاوه بر آن از فرآیند آغاز کند TCP نیز صرفنظر گردد. چنانچه در همه‌ی موارد ارتباط‌های TCP در حالت CA یا AIMD عمل نمایند آنگاه نمودار شکل ۳-۵۶ نهفتگی حاصل از این دو ارتباط TCP را نشان می‌دهد.

اگر TCP سهم برابری از پهنای باند به هر یک از دو ارتباط اختصاص دهد آنگاه نهفتگی حاصل بایستی در مسیر 45 درجه از مبدا نمودار برای این حالت ویژه در شکل ۳-۵۶ باشد. در حالت بهینه جمع دو نهفتگی بایستی برابر R باشد. بدیهی است هر ارتباط سهم برابر اما صفر از ظرفیت پیوند را دریافت می‌نماید که وضعیت مطلوبی نمی‌تواند به‌شمار آید. بنابراین هدف دستیابی به نهفتگی است که در نزدیکی تقاطع خط پهنای سهم برابر پیوند و خط بهره‌وری کامل از آن در شکل ۳-۵۶ واقع گردد.

1- Fair
2- Bottleneck link



شکل ۳-۵۵ ♦ دو ارتباط TCP با استفاده‌ی مشترک از یک پیوند تنگراه

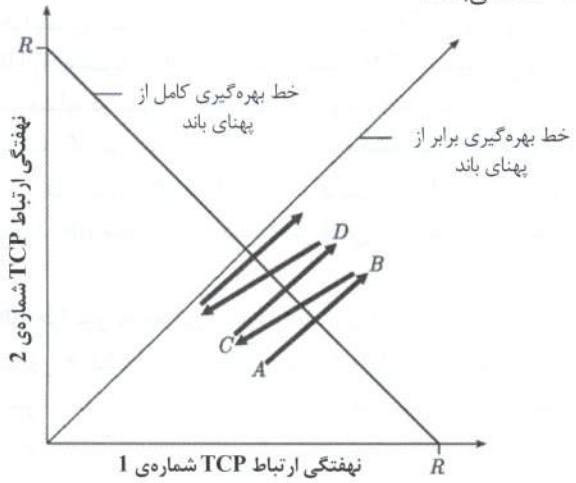
اگر فرض شود اندازه‌های پنجره‌ی TCP به گونه‌ای باشند که در یک نقطه از زمان ارتباط‌های 1 و 2 از نهفتگی نشان داده شده به‌وسیله‌ی نقطه A در شکل ۳-۵۶ برخوردار گردند. از آنجائیکه مقدار پهنای باند پیوند مشترک مورد بهره‌برداری دو ارتباط کمتر از R می‌باشد بنابراین اتلاف بسته روی نمی‌دهد و هر دو ارتباط اندازه‌ی پنجره‌ی خود را به مقدار یک MSS برای هر RTT در نتیجه‌ی الگوریتم پرهیز از تراکم TCP گسترش می‌دهند. بنابراین پیوستن نهفتگی دو ارتباط در مسیر یک خط 45 درجه به مفهوم افزایش برابر برای دو ارتباط یا آغاز از نقطه‌ی A می‌باشد. نتیجه آنکه پهنای باند پیوند مشترکاً از سوی دو ارتباط با سرعت بیش از R بیت بر ثانیه مورد استفاده قرار می‌گیرد و سپس اتلاف بسته‌ها روی می‌دهد. اگر فرض شود ارتباط‌های 1 و 2 هنگامی‌که نهفتگی در نقطه‌ی B را تجربه می‌کنند دچار وضعیت مفقودی بسته‌ها گردند، ارتباط‌های 1و 2 آنگاه با ضریب 2 به کاهش اندازه‌ی پنجره تراکم خود اقدام می‌کنند و بنابراین نهفتگی حاصل در نقطه‌ی C به دست می‌آید که در نیمه‌ی برداری است با آغاز از نقطه‌ی B و پایان در مبداء مختصات نمودار. از آنجائیکه پهنای باند مشترک کوچکتر از R در نقطه‌ی C می‌باشد و دو ارتباط مجدداً نهفتگی خود را در مسیر خط 45 درجه با آغاز از نقطه‌ی C حاصل می‌نمایند. سرانجام مفقودی بسته‌ها مجدداً روی می‌دهد، به‌عنوان مثال در نقطه‌ی D. دو ارتباط اندازه‌های پنجره‌ی تراکم خود را بازهم با ضریب 2 کاهش می‌دهند و این فرآیند تکرار می‌گردد. این بر عهده خواننده است که خود را قانع نماید که دو ارتباط با این رفتار بدون توجه به مکان قرار گرفتن در فضای دو بعدی به یکدیگر می‌پیوندند اگرچه مفروضاتی در پس این رفتارها پنهان شده است. این رویکرد می‌تواند نشان دهد چرا TCP پهنای باند مشترک بین ارتباط‌های خود را به تساوی بین آن‌ها تقسیم می‌کند.

در مطالب بالا فرض بنیادی بر آن بود که ارتباط‌های TCP همگی از طریق پیوند تنگراه انجام می‌شوند و دارای RTT مشابه از لحاظ ارزش می‌باشند و فقط یک ارتباط TCP وابسته به یک جفت میزبان مقصد می‌باشد و در عمل این شرط پیاده‌سازی نمی‌گردد و کاربردهای مشتری – سرور می‌توانند به بخش‌های غیر برابر از پهنای باند پیوند دسترسی داشته باشند. این وضعیت به‌ویژه برای حالت استفاده از چندین ارتباط مشترک از تنگراه اتفاق می‌افتد و آن دسته ارتباط‌های دارای RTT کوچک‌تر، سریع‌تر می‌توانند پهنای باند در دسترس به هنگام آزاد شدن آن‌ها را در اختیار گیرند به عبارت دیگر پنجره‌های تراکم خود را سریع‌تر تشکیل دهند و در نتیجه از نهفتگی بیشتری در مقایسه با ارتباط‌های دارای RTT طولانی‌تر بهره‌مند شوند.

- **پروتکل UDP و تعادل در ارتباط**

در بخش قبل ملاحظه گردید چگونه کنترل تراکمTCP سرعت انتقال یک کاربرد را از طریق روش‌های

پنجره‌ی تراکم تنظیم می‌نمود. بسیاری از کاربردهای چند رسانه‌ای مانند تلفن اینترنتی و کنفرانس تصویری، اغلب از پروتکل TCP به همین دلیل بهره نمی‌گیرند. زیرا آن‌ها مایل به کند و یا محدود شدن سرعت انتقال خود نمی‌باشند حتی اگر شبکه بسیار متراکم باشد. این کاربردها ترجیح می‌دهند بر روی پروتکلی مانند UDP اجرا شوند که دارای ساختار کنترل تراکم نمی‌باشند. هنگامی‌که اجرا بر روی UDP انجام می‌شود کاربردها می‌توانند صدا و تصویر را با یک سرعت ثابت و همراه با اتلاف بسته‌ها در بعضی مواقع به درون شبکه تزریق نمایند. این روش نقطه‌ی مقابل حالت TCP است که رضایت به کاهش سرعت به هنگام تراکم در برابر عدم اتلاف بسته‌ها می‌داد. از نقطه‌نظر TCP کاربردهای چندرسانه‌ای در حال اجرا بر روی UDP عادلانه نمی‌باشند زیرا آن‌ها با دیگر ارتباط‌ها همکاری نکرده و سرعت انتقال را نیز به گونه‌ی عادلانه بین خود تقسیم نمی‌کنند. از آنجائیکه کنترل تراکم TCP سرعت انتقال را به هنگام رو به رو شدن با افزایش تراکم و در نتیجه شرایط مفقودی کاهش می‌دهد، اما پروتکل UDP به میزبان‌های مبداء خود این شرایط را تحمیل نمی‌نمایدبنابراین یک مبداء UDP قادر است ترافیک TCP را متراکم نماید. یکی از نواحی پژوهش در حال حاضر بر روی معرفی روش‌های کنترل تراکم در اینترنت جهت ممانعت از کاهش نهفتگی اینترنت به گونه‌ای بحرانی توسط UDPمی‌باشد.



شکل ۳-۵۶ ♦ نهتگی حاصل از ارتباط‌های TCP یک و دو

• ارتباط‌های موازی TCP و تعادل در ارتباط

حتی اگر ترافیک UDP را بتوان وادار به رفتاری عادلانه نمود چالش تعادل هنوز هم به‌صورت کامل برطرف نمی‌شود زیرا هیچ عاملی وجود ندارد که کاربرد مبتنی بر TCP را از نقطه‌نظر بهره‌برداری ارتباط‌های موازی چندگانه متوقف نماید. به‌عنوان مثال مرورگرهای وب اغلب از ارتباط‌های TCP چندگانه‌ی موازی جهت انتقال چندین منبع داده و یا عنصر در بین یک صفحه‌ی وب استفاده می‌کنند. بدیهی است تعداد دقیق ارتباط‌های چندگانه‌ی موازی در اغلب مرورگرها قابل پیکربندی است. هنگامی که یک کاربرد از ارتباط‌های موازی بهره می‌گیرد آن کاربرد می‌تواند سهم بیشتری از پهنای باند را در یک پیوند متراکم در اختیار گیرد. به‌عنوان مثال پیوندی را تصور نمائید که سرعت آن R بیت بر ثانیه باشد و تعداد 9 کاربرد مشتری – سرور را نیز پشتیبانی نماید و هر یک از کاربردها نیز از یک ارتباط TCP بهره گیرند. اگر یک کاربرد جدید به کاربردهای جاری اضافه گردد و مایل به استفاده از یک

ارتباط TCP باشد آنگاه هر کاربرد تقریباً سرعت انتقال مشابهی معادل $\frac{R}{10}$ بیت بر ثانیه را دراختیار می‌گیرد. اما اگر کاربرد جدید از 11 ارتباط موازی TCP استفاده نماید آنگاه این کاربرد یک بهره‌برداری غیر عادلانه‌ی بیش از $\frac{R}{2}$ را در اختیار گرفته است زیرا ترافیک وب در اینترنت چنان نافذ و فراگیر می‌باشد که ارتباط‌های موازی چندگانه غیرطبیعی به نظر نمی‌رسند.

تمرین‌های فصل سوم

۱- یک ارتباط TCP بین دو میزبان A و B را درنظر بگیرید. فرض کنید قطعه‌های TCP در حال عبور از A به B دارای شماره پورت‌های مبدا و مقصد به ترتیب x و y می‌باشند. شماره پورت‌های قطعه‌های بازگشتی از میزبان‌های B به A چیست؟

۲- توضیح دهید چرا یک طراح ممکن است ترجیح دهد که برنامه‌ی کاربردی طراحی شده از سوی وی از طریق UDP به‌جای TCP به اجرا درآید؟

۳- فرض کنید یک فرآیند در میزبان C دارای شماره پورت 6789 و هر دو میزبان A وB هر کدام یک قطعه‌ی UDP را به میزبان C آدرس پورت مقصد 6789 گسیل می‌دارند. آیا این قطعه‌ها از طریق یک سوکت مشابه در میزبان C هدایت می‌شوند؟ اگر پاسخ آری می‌باشد چگونه یک پردازش در C می‌داند که این دو قطعه از دو میزبان مختلف سرچشمه گرفته‌اند؟

۴- اگر تاخیر RTT بین فرستنده و گیرنده‌ای ثابت و فرستنده از آن آگاهی داشته باشد. آیا به یک تایمر در پروتکل rdt 3.0 هنوز نیاز می‌باشد؟ اگر فرض شود امکان مفقودی بسته‌ها وجود دارد؟ توضیح دهید.

۵- در پروتکل‌های rdt چرا نیاز به معرفی شماره ردیف می‌باشد؟

۶- اگر فرض شود میزبان A قطعه‌ای با شماره ردیف 38 و داده‌ای به طول 4 بایت را از طریق یک ارتباط TCP به میزبان B گسیل می‌نماید در این قطعه نشان دهید که شماره پیام تائید موردنیاز 42می‌باشد؟

۷- اگر فرض شود میزبان A دو قطعه‌ی TCP را به‌صورت پشت به پشت به میزبان B از طریق یک ارتباط TCP گسیل می‌کند و نخستین قطعه شماره ردیف 90 و دومین قطعه شماره ردیف 110 را دارد مقدار داده در قطعه‌ی نخست چقدر است؟ اگر قطعه‌ی اول مفقود گردد و قطعه‌ی دوم به B وارد شود آنگاه پیام تائیدی که B به A گسیل می‌نماید چه شماره‌ی تائیدی دارد؟

۸- با توجه به مکانیزم کنترل تراکم در TCP هنگامی‌که تایمر به انتهای عملیات خود در فرستنده می‌رسد ارزش آستانه برابر چه مقدار قبلی آن می‌شود؟

۹- اگر فرض شود مشتری A آغازگر یک ارتباط تلنت با سرور S باشد و تقریباً همزمان با آن مشتری B نیز ارتباط تلنت را با سرور S آغاز نماید.مطلوب است شماره پورت‌های مبداء و مقصد برای هر یک از حالت‌های ذیل:

۱) قطعه‌ها از A به S ارسال گردد.

۲) قطعه‌ها از B به S ارسال گردد.

۳) قطعه‌ها از S به A ارسال گردد.

۴) قطعه‌ها از S به B ارسال گردد.

۵) اگر A و B دو میزبان مختلف باشند آیا این امکان وجود دارد که شماره پورت مبدا در قطعه‌های S به A مشابه قطعه‌های B به S باشند؟

۶) اگر هر دو میزبان A و B مشابه باشند پاسخ پرسش ۵ چه خواهد شد؟

۱۰- با توجه به شکل ۳-۵ ارزش‌های پورت مبداء و مقصد در قطعه‌های بازگشتی از سرور به پردازش مشتری چیست؟ آدرس‌های IP در داده‌گرام‌های لایه‌ی شبکه که قطعه‌های لایه انتقال را حمل می‌کنند چیست؟

۱۱- اگر 2 بایت به ترتیب 00110100 و 01101001 در اختیار باشد مکمل یک این دو بایت چیست؟ اگر در این بایت‌ها فقط یک بیت از آن‌ها عکس شود و مکمل آن‌ها تغییر نکند آیا می‌توانید نمونه‌ای را برای این حالت بیان کنید.

۱۲- پروتکل تصحیح rdt 2.1 را در مطالب فصل در نظر بگیرید. نشان دهید که گیرنده‌ی نشان داده شده در شکل ۳-۱۱ و به دنبال آن می‌تواند سبب حالت بن‌بست هر دو فرستنده و گیرنده گردد، هنگامی‌که هر یک منتظر رخدادی می‌باشند که هرگز روی نمی‌دهد.

۱۳- ماشین FSM سمت گیرنده پروتکل rdt 3.0 را رسم کنید.

۱۴- مطلوب است رهگیری عملیات پروتکل rdt 3.0 هنگامی‌که بسته‌های داده و تائید در هم ریخته باشند. رهگیری شما بایستی مشابه آنچه در شکل ۳-۱۶ نشان داده شده است، باشد.

۱۵- یک پروتکل مطمئن انتقال داده را در نظر بگیرید که فقط از پیام‌های تائید منفی بهره می‌برد. فرض کنید فرستنده، داده‌ها را تقریباً پی در پی گسیل می‌کند. آیا می‌توان گفت یک پروتکل NAK نسبت به یک پروتکل ACK در این شرایط برتری دارد؟ چرا؟

اگر فرض شود فرستنده حامل مقادیر زیادی داده جهت ارسال می‌باشد و ارتباط انتها به انتها تعداد محدودی مفقودی را تجربه می‌کند، آیا در این حالت پروتکل NAK بر ACK ترجیح دارد؟ چرا؟

۱۶- با توجه به شکل ۳-۱۷ اندازه‌ی پنجره برای بهره‌وری کامل کانال برای آنکه بیش از 90% باشد چقدر بایستی طراحی گردد؟

۱۷- فرض کنید دو برنامه‌ی کاربردی شبکه وجود دارد به نام‌های A و B ، اگر B موظف به تامین پیام‌های داده‌ای باشد که بر اساس شرایط ذیل به A ارسال می‌گردند: هنگامی که A تقاضایی را از لایه‌ی بالاتر برای دریافت داده بعدی به نام D از B دریافت می‌کند A بایستی پیام تقاضای R را به B بر روی پیوند A به B گسیل نماید. فقط هنگامی که B پیام R را دریافت می‌کند می‌تواند پیام D را به A از طریق پیوند B به A بازگشت دهد. A بایستی دقیقاً یک نسخه از هر پیام D را به لایه‌ی بالاتر تحویل نماید. پیام R می‌تواند در پیوند A به B مفقود گردد اما آلوده نشود. پیام D فقط یک بار گسیل می‌شود و همیشه نیز صحیح و بدون عیب دریافت می‌شود و سرانجام تاخیر در مسیر هر دو پیوند متغیر و مجهول می‌باشد. با توجه به موارد بالا FSM یک پروتکل را طراحی نمائید که همه مکانیزم‌های لازم جهت جبران ضعف در برابر مفقودی پیوند A به B را دارا باشد و به پیاده‌سازی روش پیام‌رسانی به لایه‌ی بالاتر کاربرد A بپردازد. فقط از مکانیزم‌های لازم استفاده کنید.

۱۸- در مطالب فصل عنوان گردید یک کاربرد ممکن است UDP را برای یک پروتکل انتقال گزینش نماید زیرا UDP کنترل دقیق‌تری را نسبت به TCP از نقطه‌نظر نوع داده‌ی ارسالی و زمان آن در درون یک قطعه اعمال می‌کند:

۱) به نظر شما چرا یک کاربرد کنترل بیشتری را از نقطه‌نظر نوع داده‌ی ارسالی در یک قطعه اعمال می‌کند؟

۲) چرا این کنترل بیشتر از سوی کاربرد در مورد زمان ارسال قطعه نیز اعمال می‌شود؟

۱۹- دو پروتکل GBN و SR را در نظر بگیرید. اگر فرض شود فضای شماره ردیف برابر K باشد. بزرگترین اندازه پنجره‌ای که یک فرستنده مجاز به تعریف آن جهت جلوگیری از قوع دشواری‌هایی نظیر آنچه در شکل ۳-۲۷ نشان داده شده است، برای هر یک از پروتکل‌ها چه مقدار است؟

۲۰- اگر انتقال یک حجم انبوه از داده در یک فایل به طول L بایت و از طریق میزبان A به B مورد نظر باشد.

۱) حداکثر طول L چقدر بایستی طراحی شود که شماره‌های ردیف TCP دچار کمبود نگردند؟ توجه داشته باشید که دامنه‌ی شماره ردیف TCP برابر 4 بایت است.

۲) جهت ارزش L زمان لازم جهت انتقال فایل چقدر است اگر مجموعاً 66 بایت از پیام مربوط به سرپیام انتقال، شبکه و پیوند داده به هر قطعه قبل از ایجاد بسته جهت ارسال از طریق یک پیوند 10 Mbps افزوده گردد. از تاثیر کنترل جریان و کنترل تراکم جهت قادرسازی میزبان برای تزریق قطعه‌ها به شبکه به گونه پشت به پشت و پی‌درپی صرف نظر نمائید.

۲۱- در مطالب فصل به کوکی‌ها SYN اشاره گردید:

۱) چرا لازم است که سرور از یک شماره ردیف اولیه و ویژه در پیام SYNACK استفاده نماید؟

۲) اگر یک مهاجم آگاه باشد که یک میزبان هدف از کوکی‌های SYN بهره می‌گیرد آیا مهاجم می‌تواند یک ارتباط نیمه‌باز و یا کاملاً باز را از طریق ارسال پیام ACK به هدف ایجاد نماید؟ چرا؟ در صورتی‌که پاسخ منفی است، چرا نه؟

۲۲- دو میزبان A و B مستقیماً از طریق یک پیوند 200 Mbpsبه یکدیگر مرتبط شده‌اند. اگر یک ارتباط TCP بین دو میزبان موجود باشد و میزبان A فایل پرحجمی را از طریق این ارتباط به میزبان B ارسال نماید و میزبان A قادر به ارسال داده‌ی کاربرد به درون پیوند با سرعت 100 Mbps باشد اما B از طریق بافر دریافت TCP خود حداکثر با سرعت 50Mbps قادر به خواندن داده‌ها باشد تاثیر کنترل جریان TCP را در این حالت تشریح نمائید.

۲۳- رابطه‌ی بین متغیر Send Base و متغیر Last Byte RCV چیست؟

۲۴- با توجه به شکل ۳-۴۶ (ب) اگر λ'_{in} فراتر از مقدار $\frac{R}{2}$ افزایش یابد، آیا λ_{out} می تواند فراتر از $\frac{R}{3}$ افزایش پیدا کند؟ توضیح دهید.

اکنون با در نظر گرفتن شکل ۳-۴۶ (ج) اگر λ'_{in} فراتر از $\frac{R}{2}$ افزایش پیدا کند آیا λ_{out} فراتر از $\frac{R}{4}$ افزایش می‌یابد. اگر فرض شود که یک بسته به‌صورت میانگین دو مرتبه از مسیریاب به گیرنده گسیل می‌شود؟ توضیح دهید.

۲۵- با مراجعه به شکل ۳-۵۵ که همگرایی الگوریتم TCPAIMD را نشان می‌دهد، فرض کنید که

در عوض یک کاهش ضرب انباشتی پروتکل TCP اندازه‌ی پنجره را با یک مقدار ثابت کاهش دهد آیا الگوریتم AIMD حاصل از این رویکرد به یک الگوریتم سهم برابر تبدیل می‌شود؟ پاسخ خود را با توجه به شکل ۳-۵۵ توجیح نمائید.

۲۶- در مطالب فصل ملاحظه شد TCP تا هنگام دریافت سه پیام ACK تکرار شده، قبل از انجام یک انتقال مجدد سریع منتظر می‌ماند. به نظر شما چرا طراحان TCP اجرای انتقال مجدد سریع را پس از نخستین پیام ACK از نسخه تکراری برای یک قطعه‌ی دریافتی گزینش نکرده‌اند؟

۲۷- میزبان A در حال گسیل یک فایل پرحجم از طریق یک ارتباط TCP به میزبان B می‌باشد. از طریق این ارتباط هرگز اتلاف بسته و رخداد زمان خروج روی نداده است. اگر سرعت پیوند اینترنت ارتباط دهنده‌ی A نیز R بیت بر ثانیه باشد و فرض شود پردازش در میزبان A قادر به گسیل داده به درون سوکت TCP در سرعت S بیت بر ثانیه می‌باشد که $S = 10R$ و اگر فرض شود بافر گیرنده‌ی TCP به اندازه‌ی کافی بزرگ است که تمامی فایل‌ها را در خود نگاهداری نماید و نیز بافر ارسال کننده تنها یک درصد (1%) از فایل را ذخیره می‌کند، کدامیک از موارد ذیل سبب می‌شود که فرآیندی در A نتواند داده‌ها را به‌صورت مستمر با سرعت S بیت بر ثانیه به درون سوکت TCP خود منتقل نماید؟

۱) کنترل جریان TCP؟

۲) کنترل تراکم TCP؟

۳) عامل دیگری؟

۲۸- با توجه به مطالب بخش‌های پایانی فصل در مورد نهفتگی TCP که در دوره‌ای از زمان سرعت ارتباط آن از مقدار $\frac{W}{2RTT}$ به $\frac{W}{RTT}$ تغییر می‌یافت و فقط یک بسته مفقود می‌گردید آن هم در انتهای دوره نشان دهید:

۱) نرخ اتلاف یعنی درصد مفقودی بسته برابر است با:

$$L = \text{نرخ مفقودی} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

۲) با بهره‌گیری از رابطه‌ی بالا نشان دهید اگر یک ارتباط دارای نرخ مفقودی L باشد آنگاه میانگین نرخ تقریباً برابر است با:

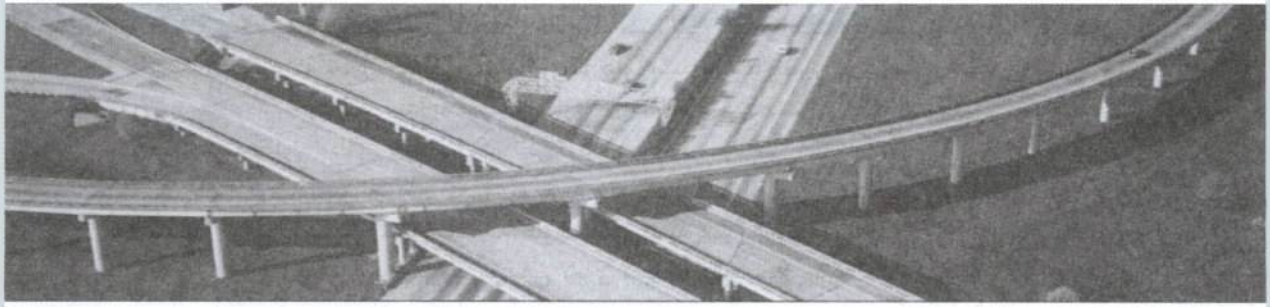
$$\text{میانگین نرخ مفقودی} \approx \frac{1.22MSS}{RTT\sqrt{L}}$$

۲۹- در بحث پیرامون ویژگی‌های TCP در انتهای فصل ۳ ملاحظه شد برای نهفتگی 10 Gbps پروتکل TCP تنها می‌توانست احتمال مفقودی قطعه را به‌صورت 2×10^{-10} تحمل نماید که مفهوم اتلاف تنها یک بسته در بین هر 5 میلیارد قطعه بود. نشان دهید:

۱) احتمال 2×10^{-10} و یک اتلاف در بین 5 میلیون قطعه برای ارزش‌های RTT و MSS داده شده در بخش ۳-۲۲ نیز می‌تواند تعریف شود.

۲) اگر TCP نیازمند پشتیبانی یک ارتباط 100Gbps باشد اندازه‌ی قابل تحمل اتلاف چقدر است؟

۳۰- مفهوم از گروگانگیری یک ارتباط TCP چیست؟ و چگونه این حادثه می‌تواند روی دهد؟



فصل ۴

لایه‌ی شبکه

در فصل قبل آموختیم که لایه‌ی انتقال حالت‌های گوناگونی از ارتباط پردازش به پردازش را از طریق تکیه بر عملیات ارتباط میزبان به میزبان لایه‌ی شبکه فراهم می‌نمود. علاوه بر آن آموختیم لایه‌ی انتقال این امکانات را بدون آگاهی از این که لایه‌ی شبکه چگونه این خدمات را پیاده‌سازی می‌نماید، انجام می‌دهد. ممکن است این پرسش مطرح گردد، چه عاملی در پس پرده‌ی عملکرد میزبان به میزبان لایه‌ی شبکه وجود دارد و چگونه آن را به حرکت وا می‌دارد؟

در این فصل دقیقاً به چگونگی پیاده‌سازی لایه‌ی شبکه از نقطه‌نظر ارتباط میزبان به میزبان پرداخته می‌شود. ملاحظه خواهد شد که برعکس لایه‌ی انتقال قطعه‌ای از لایه‌ی شبکه در هر میزبان و مسیریاب شبکه وجود دارد و به دلیل حضور این قطعه برنامه است که پروتکل‌های لایه‌ی شبکه از جمله دشوارترین و بنابراین شگفت‌انگیزترین موارد در پشته‌ی پروتکلی به شمار می‌آیند.

لایه‌ی شبکه یکی از پیچیده‌ترین لایه‌ها در پشته‌ی پروتکلی است و بنابراین موارد بی‌شماری در آن وجود دارد که نیازمند پژوهش و پیاده‌سازی می‌باشد. در آغاز این فصل ابتدا دیدگاهی کلی از لایه‌ی شبکه و خدمات قابل ارایه به وسیله‌ی آن معرفی می‌گردد و پس از آن دو رویکرد وسیع ساختار تحویل بسته در لایه‌ی شبکه که عبارتند از **داده‌گرام** و **مدار مجازی** که در فصل یک به آن‌ها اشاره رفت، تحلیل می‌شوند. نقش بنیادی آدرس‌دهی در تحویل یک بسته به میزبان مقصد آن از جمله موارد دیگر است.

در این فصل تفاوت مهم بین دو عملکرد **گسیل به جلو**[1] و **مسیریابی**[2] در لایه‌ی شبکه نیز تحلیل می‌شود. رویکرد گسیل به جلو شامل انتقال یک بسته از یک پیوند ورودی است به یک پیوند خروجی و

1- Forwarding
2- Routing

فقط در یک مسیریاب انجام می‌شود در حالی که مسیریابی شامل همه‌ی مسیریاب‌های (روترهای) شبکه می‌شود که تعامل جمعی آن‌ها با پروتکل‌های مسیریابی به تعیین مسیرهایی منجر می‌شود که بسته‌ها پویش خود را از مبدا به مقصد از طریق آن‌ها انجام می‌دهند. این یکی از مهم‌ترین تفاوت‌هایی است که در طول مطالب این فصل در نظر گرفته شده است.

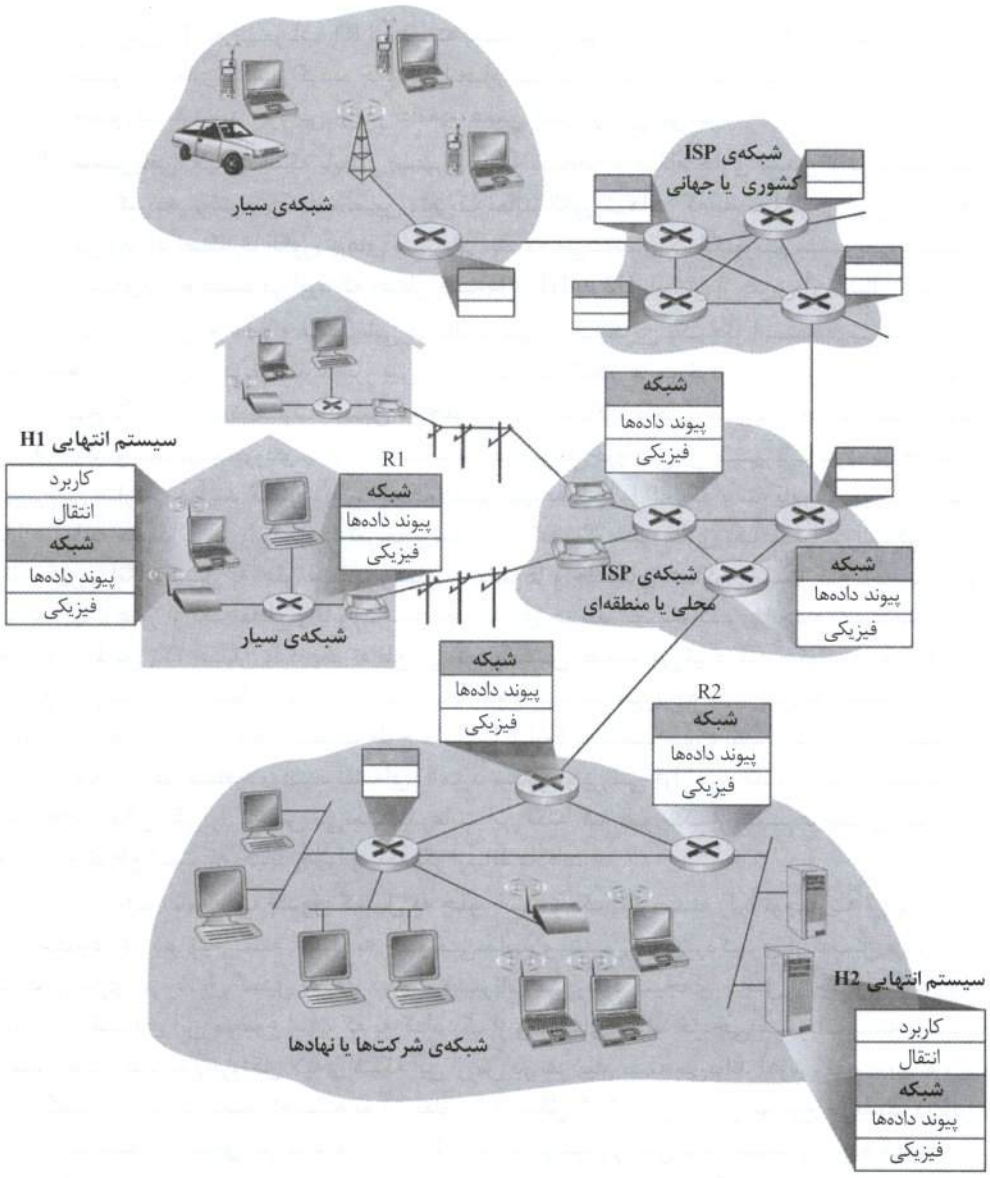
جهت درک عمیق‌تر روش گسیل بسته به جلو یک مسیر از لحاظ عملکرد و ساختار درونی آن بررسی می‌شود. سپس گسیل بسته به جلو در اینترنت به همراه پروتکل معروف آن IP بررسی می‌شود. در ادامه‌ی موضوعات این فصل آدرس‌دهی لایه‌ی شبکه و قالب‌بندی داده گرام‌های IPV4[1] از جمله موارد با اهمیت این فصل به شمار می‌آید. علاوه بر آن ترجمه‌ی آدرس شبکه NAT، تقسیم‌بندی داده‌گرام‌ها، پروتکل کنترل پیام ICMP و کپسول‌بندی در قالب IPV6 از جمله موارد دیگری هستند که در این فصل بررسی می‌شوند.

پس از بررسی موارد یاد شده عملکرد مسیریابی لایه‌ی شبکه از نقطه‌نظر یک الگوریتم مسیریابی جهت شناسایی بهترین مسیر ممکن بین فرستنده‌ها و گیرنده‌ها تحلیل می‌شود. نظریه‌ی الگوریتم‌های مسیریابی و تمرکز بر روی دو گروه بسیار مهم از این الگوریتم‌ها نخستین موردی می‌باشد که در این راستا بررسی می‌شوند. از آنجائیکه پیچیدگی الگوریتم‌های مسیریابی با افزایش تعداد مسیریاب‌ها رشد چشمگیری را به همراه خواهد داشت بنابراین رویکرد سلسله مراتبی مسیریابی با اهمیت می‌باشد. پیاده‌سازی عملی این نظریه‌ها در قالب پروتکل‌های مسیریابی سیستم خود مختار داخلی اینترنت مانند: RIP، OSPE، IS-IS و البته پروتکل‌های مسیریابی سیستم خودمختار بین شبکه مانند BGP به گونه‌ی عمیق بررسی می‌شوند. در انتهای مطالب فصل مسیریابی پخش گسترده و چند پخشی بررسی می‌گردد. به اختصار این فصل شامل سه بخش بسیار مهم است که عبارتند از: خدمات و عملکردهای لایه‌ی شبکه، روش‌های گسیل به جلو مسیریابی.

۴-۱ خدمات و عملکردهای لایه‌ی شبکه

شکل ۴-۱ یک شبکه‌ی ساده با دو میزبان H1 و H2 و چندین مسیریاب را در بین مسیر بین دو میزبان نامبرده نشان می‌دهد. اگر فرض شود میزبان H1 اطلاعات به H2 گسیل می‌نماید و نقش لایه‌ی شبکه در این میزبان‌ها و مسیریاب‌های بین آن‌ها مورد نظر باشد آنگاه لایه‌ی شبکه در H1 قطعه‌های دریافتی از لایه‌ی انتقال در H1 را کپسول‌بندی می‌نماید و هر قطعه را تبدیل به یک **داده‌گرام** می‌کند که یک بسته‌ی لایه‌ی شبکه به شمار می‌آید. سپس داده‌گرام‌ها را به نزدیک‌ترین مسیریاب R1 گسیل می‌کند. در میزبان دریافت کننده H2 ، لایه‌ی شبکه داده‌گرام‌ها را از نزدیک‌ترین روتر R2 دریافت و پس از خارج نمودن قطعه‌های لایه‌ی انتقال آن‌ها را به لایه‌ی انتقال در H2 تحویل می‌نماید. نقش اولیه‌ی مسیریاب‌ها گسیل داده‌گرام‌ها به جلو از پیوند ورودی به پیوند خروجی است. بیان این نکته ضروری است که روترهای شکل ۴-۱ به همراه یک پشته‌ی پروتکلی خلاصه شده نشان داده شده‌اند به عبارت دیگر لایه‌های بالاتر از لایه‌ی شبکه حضور ندارند زیرا به جز برای مقاصد کنترل، مسیریاب‌ها به اجرای پروتکل‌های دو لایه‌ی کاربرد و انتقال نمی‌پردازند. مانند آنچه در فصل‌های ۲ و ۳ ملاحظه گردید.

1- Internet protocol version 4



شکل ۴-۱ ♦ لایه‌ی شبکه

• گسیل به جلو و مسیریابی

نقش لایه‌ی شبکه را می‌توان به گونه‌ای اغراق‌آمیز ساده قلمداد نمود و آن عبارت است از حرکت بسته‌ها از یک میزبان ارسال کننده به یک میزبان گیرنده. برای انجام این امر لایه‌ی شبکه دو کار مهم را انجام می‌دهد:

۱- **گسیل به جلو:** هنگامی که یک بسته به پیوند ورودی یک مسیریاب وارد می‌شود، مسیریاب بایستی بسته را به پیوند خروجی مناسب آن ارسال نماید. به عنوان مثال پس از ورود یک بسته

از میزبان H1 به مسیریاب R1 آن بسته بایستی از طریق مسیری به H2 با بهره‌گیری از مسیریاب بعدی گسیل گردد. چگونگی انجام این مهم یعنی گسیل بسته از پیوند ورودی مسیریاب به پیوند خروجی آن در ادامه‌ی همین فصل بررسی می‌شود.

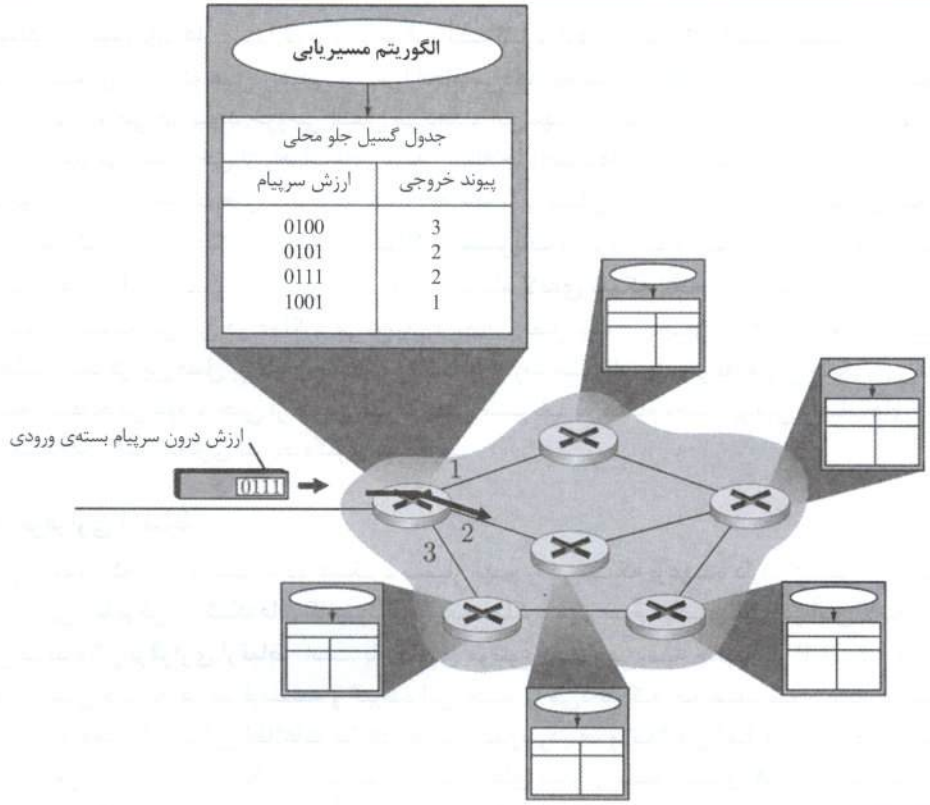
۲- **مسیریابی:** لایه‌ی شبکه بایستی مسیری را که بسته‌ها به هنگام جاری شدن از فرستنده به یک گیرنده پویش می‌کنند، تعیین وتعریف نماید. الگوریتم‌هایی که به محاسبه‌ی این مسیرها می‌پردازند اصطلاحاً الگوریتم‌های مسیریابی نامیده می‌شوند. یک الگوریتم مسیریابی معمولاً مسیری را به دست می‌آورد که جریان بسته‌ها از H1 به H2 بایستی از طریق آن دنبال کنند.

دو عبارت گسیل به جلو و مسیریابی در اغلب موارد جایگزین یکدیگر استفاده می‌شوند. در صورتیکه شایسته است در نوشتارها مرز بین این دو عبارت کاملاً رعایت گردد و در این اثر بر رعایت این موضوع تاکید می‌شود. عبارت **گسیل به جلو** اشاره به عملکرد داخلی مسیریاب جهت انتقال یک بسته از یک واسط پیوند ورودی به واسط مناسبی در پیوند خروجی آن مسیریاب دارد. در حالیکه مسیریابی اشاره به فرآیند گسترده‌تر شبکه که مسیر انتها به انتها را که یک بسته بایستی پویش نماید تا از مبدا به مقصد نهایی وارد گردد، تعریف و تعیین می‌کند. اگر شبیه‌سازی یک عملیات رانندگی را در نظر بگیریم آنگاه گسیل به جلو به مفهوم عبور از تقاطع‌ها و جاده‌های گوناگون است که بایستی هر مرتبه جهت عبور از یک تقاطع و یا وارد شدن به جاده‌ی جدید تصمیم‌گیری انجام شود. بنابراین خودرو از طریق یک خیابان وارد یک تقاطع می‌شود و سپس تصمیم می‌گیرد که چگونه از طریق آن تقاطع و گزینش یک خیابان دیگر از آن عبور و یا خارج شود. در صورتیکه مسیریابی به مفهوم آن است که قبل از خارج شدن از محل سکونت طرح مسافرت از آنجا به مقصد مورد نظر برنامه‌ریزی شود و بنابراین قبل از آغاز مسافرت راننده نقشه‌ای را مورد مشاوره و بررسی قرار می‌دهد و از میان انبوهی از مسیرهای ممکن یکی را گزینش می‌نماید با رعایت این نکته که هر یک از مسیرها شامل تعدادی خیابان و تقاطع است که خیابان‌ها را به یکدیگر ارتباط داده‌اند.

هر مسیریاب شامل یک **جدول گسیل به جلو**[1] است و یک روتر، بسته را با توجه به ارزش درون یک محدوده در سر پیام بسته گسیل به جلو می‌نماید و سپس با بهره‌گیری از همین ارزش به شاخص‌گذاری در جدول گسیل به جلو در یک مسیریاب می‌پردازد. نتیجه‌ی حاصل از جدول گسیل به جلو بیان کننده‌ی این موضوع است که به کدام یک از واسط‌های پیوند خروجی مسیریاب بسته بایستی گسیل گردد. بنابر نوع پروتکل لایه‌ی شبکه این ارزش در سر پیام بسته می‌تواند آدرس مقصد بسته و یا بیان کننده‌ی ارتباطی باشد که بسته به آن تعلق دارد. شکل ۴-۲ مثالی از این موضوع می‌باشد. در این شکل یک بسته با دامنه‌ی سر پیام به ارزش 0111 به مسیریاب وارد می‌شود. مسیریاب به کاوش در جدول گسیل به جلوی خود می‌پردازد و واسط پیوند خروجی برای بسته را گزینش می‌کند. در این جدول واسط شماره‌ی 2 مربوط به این ارزش سرپیام است. روتر آنگاه در درون خود به گونه‌ی داخلی، بسته را به واسط 2 گسیل می‌کند.

ممکن است این پرسش مطرح گردد که جداول نامبرده در مسیریاب‌ها چگونه پیکربندی و مقداردهی می‌شوند. این موضوع یکی از موارد بسیار مهم است که تفاوت مهم عملکرد بین مسیریابی و گسیل به جلو را آشکار می‌سازد. همانگونه که در شکل ۴-۲ نشان داده شده است، الگوریتم مسیریابی ارزش‌های وارد شده به درون جداول گسیل به جلو در مسیریاب‌ها را مشخص می‌کند.

1- Forwarding table



شکل ۴-۲ ♦ الگوریتم مسیریابی به تعیین ارزش‌های جدول‌های گسیل به جلو می‌پردازد.

الگوریتم مسیریابی ممکن است متمرکز عمل نماید به این مفهوم که اجرای الگوریتم بر روی یک سامانه‌ی مرکزی انجام شود و اطلاعات مسیریابی را از آن به هر یک از روترها بارگیری نماید و یا آنکه ممکن است غیرمتمرکز باشد به این مفهوم که قطعه‌ای از الگوریتم مسیریابی به گونه‌ی توزیع شده در هر مسیریاب اجرا گردد. در هر یک از دو حالت، یک مسیریاب پیام‌های پروتکل مسیریابی را دریافت می‌کند که از آن‌ها برای پیکربندی جدول گسیل به جلو بهره می‌گیرد. عملکرد متفاوت و کاملاً متمایز گسیل به جلو و مسیریابی را می‌توان با در نظر گرفتن یک فرضیه که از لحاظ عملی امکان‌پذیر است بهتر توصیف نمود. این فرضیه به حالتی اشاره می‌کند که در آن جدول‌های گسیل به جلو از طریق افراد مستقیماً و با حضور فیزیکی آن‌ها در درون مسیریاب‌ها پیکربندی می‌شوند! بدیهی است در این حالت نیازی به پروتکل‌های مسیریابی نمی‌باشد! و البته هر یک از افراد عمل کننده نیازمند تعامل با یکدیگر می‌باشند که مطمئن شوند جدول‌ها به گونه‌ای پیکربندی می‌شوند که بسته‌ها به مقاصد مورد نظر خود وارد خواهند شد. این احتمال نیز وجود دارد که پیکربندی انسانی سبب بروز خطای بیشتر و واکنش کندتر به تغییرات در برابر چیدمان شبکه گردد تا یک پروتکل مسیریابی. بنابراین شبکه‌ها هر دو مورد مسیریابی و گسیل به جلو را در درون خود استفاده می‌کنند!

دو عبارت دیگر که در اغلب موارد در ارتباط با بحث لایه‌ی شبکه به غلط جایگزین یکدیگر استفاده می‌شوند و در این مقطع لازم است توضیحاتی در مورد آن‌ها ارایه گردد عبارتند از: **راهیاب‌های لایه**

پیوند[1] و **مسیریاب‌ها**[2]. ابتدا لازم است به این نکته اشاره گردد که عبارت **راهیاب بسته**[3] به مفهوم کلی وسیله‌ای است که عمل راهیابی بسته را انجام می‌دهد. به عبارت دیگر یک بسته را از واسط پیوند ورودی به واسطی در پیوند خروجی انتقال می‌دهد و این مهم را با توجه به ارزش درون میدان سرپیام بسته انجام می‌دهد. برخی از راهیاب‌های بسته اصطلاحاً راهیاب‌های لایه‌ی پیوند نامیده شده‌اند زیرا اینگونه راهیاب‌ها تصمیم‌های مربوط به گسیل به جلو را بر مبنای ارزش درون سرپیام **لایه‌ی پیوند** اتخاذ می‌کنند. برخی دیگر از راهیاب‌ها اصطلاحاً **مسیریاب** (روتر) نامیده شده‌اند زیرا تصمیم‌های مربوط به گسیل به جلو را بر مبنای ارزش درون سرپیام **لایه‌ی شبکه** انجام می‌دهند. جهت درک بهتر موضوع و تفاوت بین این دو عملکرد می‌توانید به مطالب فصل یک در بخش ۱-۱۶ مراجعه نمائید. از آنجائیکه تمرکز این فصل بر لایه‌ی شبکه می‌باشد از عبارت مسیریاب یا روتر به عنوان جایگزین راهیاب بسته استفاده می‌شود و حتی از همین عبارت یعنی مسیریاب به هنگام بحث پیرامون راهیاب‌های بسته در شبکه‌های مدار مجازی نیز بهره‌گیری شده است.

- **برقراری ارتباط**

اشاره گردید که لایه‌ی شبکه دو عملکرد بسیار مهم را در شبکه برعهده دارد، گسیل به جلو و مسیریابی. اما برخی از شبکه‌های کامپیوتری از یک عملکرد مهم سوم دیگری نیز در این لایه بهره می‌گیرند و آن **برقراری ارتباط**[4] است. با یادآوری موضوع دست‌دهی سه جانبه در TCP که قبل از جاری شدن داده به هر دو فرستنده و گیرنده این اختیار را می‌دهد که اطلاعات مورد نیاز وضعیت ارتباط را بدست آورند. این اطلاعات مواردی مانند شماره ردیف و اندازه‌ی آغازین پنجره‌ی کنترل جریان می باشند. به جز شبکه‌ی اینترنت اغلب شبکه‌های دیگر از مسیریاب‌های گزینش شده در یک مسیر از مبدا تا مقصد می‌خواهند که جهت ورود به حالت برقراری ارتباط با یکدیگر عملیات دست‌دهی را انجام دهند و پس از آن اجازه می‌دهند بسته‌های داده متعلق به لایه‌ی شبکه در بین یک ارتباط بین مبدا و مقصد آغاز به جریان نمایند. در لایه‌ی شبکه این فرآیند اصطلاحاً برقراری ارتباط نامیده می‌شود.

۲-۴ الگوی خدمات شبکه

قبل از گام نهادن در جزئیات لایه‌ی شبکه ابتدا نگاهی گسترده به انواع خدمات قابل ارایه از سوی لایه‌ی شبکه انداخته می‌شود. هنگامیکه لایه‌ی انتقال در یک میزبان فرستنده بسته‌ای را به درون شبکه منتقل می‌کند که مفهوم آن عبور بسته به لایه‌ی شبکه در میزبان فرستنده می‌باشد آیا لایه‌ی انتقال می‌تواند این اطمینان را داشته باشد که لایه‌ی شبکه بسته را به مقصد تحویل می‌دهد؟ هنگامیکه چندین بسته در حال گسیل می‌باشند آیا همگی آن‌ها به لایه‌ی انتقال مقصد بنابر همان ترتیب ارسال تحویل می‌گردند؟ آیا زمان لازم بین ارسال دو انتقال بسته‌ی پی در پی مشابه، همان زمان در هنگام دریافت آن‌ها می‌باشد؟ آیا شبکه امکانات بازخورد (فیدبک) در مورد وجود تراکم در شبکه را فراهم می‌آورد؟ ویژگی‌های پیوند ارتباط دهنده‌ی لایه‌ی انتقال در میزبان‌های فرستنده و گیرنده چیست؟

1- Link Layer Switch
2- Routers
3- Packet Switch
4- Connection Setup

پاسخ همه‌ی این پرسش‌ها و دیگر معماهای از این دست را می‌توان از طریق الگوی خدمات لایه‌ی شبکه تعریف نمود. **الگوی خدمات لایه‌ی شبکه**[1] مشخصات انتقال انتها به انتهای بسته‌ها را بین سیستم‌های انتهایی فرستنده و گیرنده تعریف می‌کند.

در این جایگاه از مطالب این بخش برخی از خدمات ممکن که لایه‌ی انتقال می‌تواند آن‌ها را تامین نماید، بررسی می‌شوند. در میزبان فرستنده هنگامیکه لایه‌ی انتقال بسته‌ای را به لایه‌ی شبکه تحویل می‌دهد، خدمات ویژه‌ای که از سوی لایه‌ی شبکه می‌تواند قابل ارایه باشد عبارتند از:

۱- **تحویل ضمانت شده:** این خدمت ضمانت لازم جهت تحویل و ورود بسته را به مقصد آن فراهم می‌آورد.

۲- **تحویل ضمانت شده به همراه تاخیر معین:** این خدمت نه تنها ضمانت لازم برای تحویل بسته را فراهم می‌آورد بلکه تحویل آن را در یک محدوده‌ای از زمان تاخیر مشخص بین دو میزبان به عنوان مثال 100 ms ضمانت می‌کند.

علاوه بر موارد بالا خدمات ذیل نیز می‌تواند به یک **جریان از بسته‌ها** بین یک مبدا و مقصد معین نیز ارایه گردد:

۱- **ضمانت تامین حداقل پهنای باند:** این امکانات لایه‌ی شبکه رفتار یک پیوند انتقال را تقلید می‌کند که سرعت مشخصی مانند 1 Mbps را بین میزبان‌های فرستنده و گیرنده دارد. اگر چه مسیر انتها به انتها ممکن است از چندین پیوند فیزیکی واقعی عبور نماید. تا زمانی که میزبان فرستنده به گسیل بیت‌ها به عنوان بخشی از بسته‌ها ادامه می‌دهد که سرعت آن کمتر از نرخ مشخص شده است هیچ یک از بسته‌ها مفقود نخواهند شد و هر یک از آن‌ها نیز در یک محدوده زمان تاخیر معین مانند 40 ms بین دو میزبان مبادله می‌شوند.

۲- **تحویل منظم بسته‌ها:** این سرویس ضمانت لازم جهت تحویل منظم بسته‌ها را در مقصد بنابر نوبت ارسال آن‌ها از مبدا فراهم می‌آورد.

۳- **ضمانت حداکثر اختلال زمانی:** این خدمت مقدار زمان بین انتقال دو بسته‌ی پی در پی در فرستنده را به گونه‌ای ضمانت می‌کند که این مقدار زمان معادل زمان دریافت همان بسته‌ها در مقصد باشد و یا آنکه تغییر ناپایداری زمانی بین دو حالت از یک مقدار معین تجاوز نکند.

۴- **حفاظت‌های امنیتی:** با بهره‌گیری از یک **کلید تعامل سری**[2] (رمز) که فقط از سوی میزبان‌های مبدا و مقصد شناخته شده است، لایه شبکه در میزبان محموله‌ی داده‌ها در همه‌ی داده‌گرام‌ها را رمزگذاری و آماده جهت گسیل به مقصد می‌کند. آنگاه لایه‌ی شبکه در میزبان مقصد مسئولیت بازگشایی محموله‌ها را بر عهده می‌گیرد. با ارایه‌ی چنین خدمتی محرمانه ماندن همه‌ی قطعه‌های لایه‌ی انتقال چه TCP و یا UDP بین دو میزبان مبدا و مقصد فراهم می‌آید. علاوه بر این موضوع لایه‌ی شبکه می‌تواند خدمات یکپارچگی داده و اعتبارسنجی آن‌ها را نیز فراهم آورد.

بدیهی است این بخشی از خدمات قابل ارایه از سوی لایه‌ی شبکه می‌باشد و تعداد بی‌شماری از این موارد قابل تعریف و پیاده‌سازی است.

1- Network Service Model
2- Secret Session Key

لایه‌ی شبکه در اینترنت تنها یک خدمت را فراهم می‌آورد که اصطلاحاً به آن **ارایه با بهترین تلاش**[1] گفته می‌شود. از جدول ۴-۱ ظاهراً می‌توان این برداشت را نمود که خدمات با بهترین تلاش بهترین حالت ممکن از هیچ‌گونه خدمت است. در سرویس بهترین تلاش رعایت زمان‌بندی بین بسته‌ها ضمانت نمی‌شود. به عبارت دیگر ضمانت نظم و ترتیب بسته‌ها و یا تحویل بسته‌های انتقال یافته در مقصد فراهم نمی‌شود. با این تعریف شبکه‌ای که ضمانت تحویل بسته‌ای را به مقصد نمی‌کند به ارایه‌ی خدمت بهترین تلاش در تحویل می‌پردازد. البته در ادامه‌ی همین فصل دلایل لازم جهت وجود چنین امکان بسیار کم ارزش از الگوی لایه‌ی شبکه توضیح داده می‌شود.

جدول ۴-۱ ♦ الگوهای خدمات شبکه‌های گوناگون

معماری شبکه	الگوی خدمت	ضمانت پهنای باند	ضمانت عدم مفقودی	ضمانت ترتیب	ضمانت زمان‌بندی	ضمانت آگاهی از تراکم
اینترنت جهانی	بهترین تلاش	هیچگونه	هیچگونه	به هر ترتیب ممکن	رعایت نمی‌شود	هیچگونه
روش انتقال غیر همزمان(ATM)	CBR	ضمانت نرخ پایدار	آری	رعایت تریب	رعایت می‌شود	تراکم روی نمی‌دهد
روش انتقال غیر همزمان(ATM)	ABR	ضمانت حداقل	هیچگونه	رعایت ترتیب	رعایت نمی‌شود	آری

معماری‌های دیگر شبکه الگوهای خدماتی به مراتب بیشتر از اینترنت را تعریف و پیاده‌سازی نموده‌اند که حتی از خدمت بهترین تلاش هم فراتر رفته است. به عنوان مثال معماری شبکه‌ی ATM الگوهای چندگانه‌ی خدماتی را فراهم می‌آورد که مفهوم آن ارتباط‌های گوناگونی است که در بین یک شبکه اما با گروه‌های خدماتی مختلف امکان‌پذیر می‌شود. بحث پیرامون شبکه‌ای مانند ATM و چگونگی فراهم آوری خدمات از سوی آن در چارچوب اهداف این اثر قرار نمی‌گیرد بلکه منظور اصلی بیان این نکته است که جایگزین‌هایی برای الگوی بهترین تلاش اینترنت نیز وجود دارد. در ادامه‌ی مطالب این بخش دو الگوی مهم خدماتی دیگر شبکه‌ی ATM که عبارتند از: (۱) سرعت ثابت و (۲) سرعت در دسترس، بررسی می‌شوند:

۱- **نرخ ثابت خدمات شبکه‌ی ATM** : این خدمت که اختصاراً *CBR* به مفهوم فارسی **نرخ ثابت**[2] نامیده می‌شود. نخستین الگوی استاندارد خدمات ATM به شمار می‌آید که جهت تامین خواسته‌های شرکت‌های تلفنی تعریف گردید. از CBR جهت حمل ترافیک داده‌های صوتی و تصویری با نرخ ثابت و بی‌درنگ در شبکه‌های تلفنی استفاده می‌شود. هدف از CBR تامین جریانی از بسته‌ها می‌باشد که در واژگان ATM سلول نامیده می‌شوند و این جریان به صورت یک خط لوله‌ی مجازی همانند یک پیوند انتقال با پهنای باند ثابت بین میزبان‌های فرستنده و گیرنده عمل می‌نماید. از طریق CBR سلول‌های ATM در سراسر شبکه به گونه‌ای عبور داده

1- Best effort service
2- Constant Bit Rate (CBR)

می‌شوند که تاخیر انتها به انتهای سلولی، و نیز تغییرات این زمان تاخیرمانند ناپایداری زمانی، درصد احتمال سلول‌های تلف شده و یا تحویل با تاخیر از یک ارزش تعریف شده‌ی ویژه کمتر خواهد بود. این ارزش‌ها از طریق میزبان فرستنده و شبکه‌ی ATM به هنگام برقراری اولین ارتباط CBR بین آن‌ها مورد توافق قرار می‌گیرد.

۲- **سرعت در دسترس خدمات شبکه‌ی ATM** : این خدمات نیز اختصاراً **ABR**[1] نامیده شده است و مربوط به هنگامی است که شبکه‌ی اینترنت خدمت با حداکثر تلاش را ارایه می‌دهد بنابراین ABR را در شبکه‌ی ATM می‌توان به عنوان خدمتی که تا اندازه‌ای بهتر از آنچه اینترنت ارایه می‌دهد، به شمار آورد. مانند الگوی خدماتی اینترنت سلول‌ها در ABR ممکن است مفقود شوند. اما بر عکس اینتزنت سلول‌ها را نمی‌توان به نظم درآورد اگر چه ممکن است تلف شوند. البته یک حداقل **نرخ انتقال سلول** (MCR)[2] جهت ارتباطی که از ABR بهره می‌گیرد، ضمانت می‌شود. اگر شبکه دارای منابع آزاد در مقیاس کافی در یک زمان ویژه باشد فرستنده ممکن است بتواند سلول‌های خود را با موفقیت و سرعت افزونتر از MCR گسیل نماید. علاوه بر این همانگونه که در بخش ۳-۱۹ ملاحظه شد خدمت ABR در شبکه‌ی ATM می‌تواند بازخوردی را برای فرستنده فراهم آورد که فرستنده از طریق آن بتواند به کنترل سرعت بین MCR و یک حداکثر سرعت سلولی مجاز بپردازد. این پیام بازخورد در قالب یک بیت تراکم و یا تعریف یک سرعت ارسال ویژه می‌باشد.

۴-۳ شبکه‌های مدار مجازی و داده‌گرام

در فصل ۳ ملاحظه شد که لایه‌ی انتقال خدمات از نوع اتصال‌گرا و یا غیر اتصال‌گرا را در اختیار کاربردها قرار می‌دهد. به عنوان مثال لایه‌ی انتقال در شبکه‌ی اینترنت گزینه‌ای از سرویس UDP غیر اتصال‌گرا و یا TCP اتصال‌گرا را در اختیار هر کاربرد قرار می‌دهد. به گونه‌ی مشابه لایه‌ی شبکه نیز همین امکانات را می‌تواند فراهم آورد. خدمات اتصال‌گرا و غیر اتصال‌گرا در لایه‌ی شبکه از نقطه‌نظر بسیاری به موازات دو سرویس لایه‌ی انتقال می‌باشد. به عنوان مثال یک ارتباط لایه‌ی شبکه آغاز عملیات خود را با دست‌دهی بین میزبان‌های مبدا و مقصد فعال می‌کند و یک سرویس غیر اتصال‌گرای لایه‌ی شبکه نیاز به انجام عملیات دست‌دهی ندارد.

اگرچه خدمات اتصال‌گرا و غیر اتصال‌گرا در لایه‌ی شبکه در مواردی به موازات خدمات ارایه شده از سوی لایه‌ی انتقال می‌باشد اما تفاوت‌های قابل ملاحظه‌ای نیز بین آن‌ها وجود دارد که عبارتند از:

۱- در لایه‌ی شبکه این خدمات به گونه‌ی میزبان به میزبان از سوی لایه‌ی شبکه برای لایه‌ی انتقال فراهم می‌آید. در لایه‌ی انتقال این خدمات به گونه‌ی فرآیند به فرآیند از سوی لایه‌ی انتقال برای لایه‌ی کاربرد می‌باشد.

۲- در همه‌ی معماری‌های مهم شبکه مانند اینترنت، ATM و غیره، لایه‌ی شبکه سرویس غیر اتصال‌گرای میزبان به میزبان و یا از نوع اتصال‌گرای آن را فراهم می‌آورد اما نمی‌تواند هر دو را شامل شود. شبکه‌های کامپیوتری که فقط سرویس از نوع اتصال‌گرا را در لایه‌ی شبکه فراهم

1- Available Bit Rate (ABR)
2- Minimum Cell Rate

می‌آورند اصطلاحاً **شبکه‌های مدار مجازی** (VC)[1] نامیده می‌شوند و شبکه‌هایی که فقط قادر به تامین سرویس غیر اتصال‌گرا در لایه‌ی شبکه می‌باشند، اصطلاحاً **شبکه‌های داده‌گرام** (DGN)[2] نامیده شده‌اند.

۳- پیاده‌سازی سرویس اتصال‌گرا در لایه‌های انتقال و شبکه کاملاً متفاوت بوده و در فصل ۳ ملاحظه شد که نوع اتصال‌گرا در لایه‌ی انتقال در لبه یا نبش شبکه در سیستم‌های انتهایی پیاده‌سازی می‌شود، در صورتیکه در ادامه‌ی مطالب این فصل ملاحظه خواهد شد که سرویس اتصال‌گرا در لایه‌ی شبکه در درون مسیریاب‌ها که بخشی از هسته‌ی شبکه به شمار می‌آیند و نیز سیستم‌های انتهایی شبکه پیاده‌سازی می‌شوند.

شبکه‌های مدار مجازی و داده‌گرام دو گروه بنیادی مختلف از شبکه‌های کامپیوتری می‌باشند و از اطلاعات کاملاً متفاوتی جهت تصمیم‌گیری فرآیند گسیل به جلو برای داده‌ها استفاده می‌کنند.

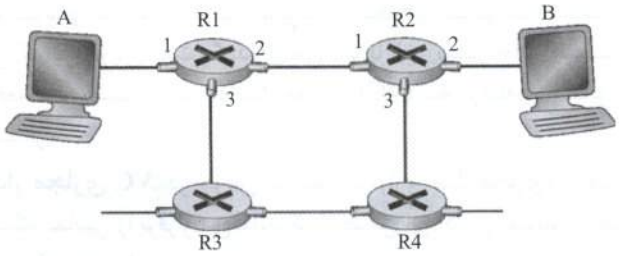
• شبکه‌های مدار مجازی (VC)

ملاحظه شد که اینترنت یک شبکه‌ی داده‌گرام است. اما بسیاری دیگر از معماری‌های شبکه مانند ATM و غیره شبکه‌های از نوع مدار مجازی هستند و بنابراین از ارتباط‌ها در لایه‌ی شبکه بهره می‌گیرند. این ارتباط‌های لایه‌ی شبکه اصطلاحاً **مدارهای مجازی** (VC) نامیده شده‌اند. در ادامه مطالب به توصیف اینکه یک سرویس VC چگونه در یک شبکه‌ی کامپیوتری پیاده‌سازی می‌شود، می‌پردازیم.

یک مدار مجازی (VC) عبارت است از (۱) یک مسیر که مجموعاً شامل تعدادی پیوند و مسیریاب است که بین میزبان‌های مبدا و مقصد مورد بهره‌برداری قرار می‌گیرد. (۲) شماره‌های VC که به مفهوم یک شماره برای هر پیوند در مسیر است و (۳) موجودیت‌های جدول گسیل به جلو در هر روتر در طول مسیر. معمولاً یک بسته که متعلق به یک مدار مجازی (VC) می‌باشد به همراه خود یک شماره‌ی VC دارد که در سر پیام آن حمل می‌شود. از آنجائیکه یک مدار مجازی ممکن است در هر پیوند دارای یک شماره VC متفاوت باشد بنابراین هر روتر واسط در مسیر بایستی شماره VC هر بسته‌ی در حال عبور را با یک شماره VC جدید جایگزین نماید. شماره VC جدید با بهره‌گیری از اطلاعات جدول گسیل به جلو بدست می‌آید.

برای نمایش بهتر این موضوع شبکه‌ای که در شکل ۴-۳ نشان داده شده است را در نظر بگیرید. شماره‌های همجوار پیوندهای مسیریاب R1 در این شکل عبارتند از شماره‌های واسط پیوند. اگر فرض شود میزبان A درخواست یک ارتباط VC را بین خود و میزبان B در یک شبکه نماید و نیز شبکه به گزینش مسیر: $A \rightarrow R1 \rightarrow R2 \rightarrow B$ اقدام نماید و شماره‌های VCبه ترتیب: 12 ، 22 ، 32 را به سه پیوند این مسیر VC اختصاص دهد در این هنگام اگر یک بسته در این VC میزبان A را ترک نماید ارزش درون محدوده‌ی شماره VC در سرپیام بسته برابر 12 خواهد شد و به هنگام ترک R1 این شماره 22 و سرانجام هنگام ترک R2 ارزش VC برابر 32 می‌شود.

1- Virtual Circuit Networks
2- Datagram Networks



شکل ۴-۳ ♦ یک شبکه‌ی ساده مدار مجازی

اکنون این پرسش مطرح می‌گردد که چگونه مسیریاب، شماره VC جایگزین برای یک بسته‌ی عبوری از خویش را بدست می‌آورد؟ در یک شبکه‌ی VC هر جدول گسیل به جلو در مسیریاب شامل سه بخش مُفَسّر شماره VC نیز می‌باشد. به عنوان مثال جدول گسیل به جلو برای مسیریاب R1 ممکن است به گونه‌ی ذیل باشد:

واسط ورودی	شماره VC ورودی	واسط خروجی	شماره VC خروجی
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
---	---	---	---

هنگامیکه یک VC جدید در فراسوی یک مسیریاب ایجاد می‌شود ورودی جدیدی به اطلاعات جدول گسیل به جلو افزوده می‌شود. به گونه‌ی مشابه هنگامی که یک VC خاتمه می‌یابد اطلاعات مربوط به ورودی‌ها از هر یک از جداول گسیل به جلو حذف می‌شود. ممکن است این پرسش نیز در ذهن ایجاد شود چرا بسته یک شماره VC مشابه را بر روی هر یک از پیوندهای مسیرخود نگهداری نمی‌کند. دو علت را می‌توان در این مورد بیان نموده نخست آنکه جایگزینی شماره از پیوند سبب کاهش طول دامنه‌ی VC در سرپیام بسته می‌شود و دوم آنکه ایجاد ارتباط VC از طریق اختصاص شماره VCهای گوناگون در هر پیوند مسیر VC به مراتب ساده‌تر خواهد شد. به ویژه با چندین شماره VC هر پیوند در مسیر می‌تواند به گزینش یک شماره VC، مستقل از شماره VC گزینش شده در پیوندهای دیگر مسیر بپردازد. اگر قرار باشد از یک شماره VCمشترک برای همه‌ی پیوندهای در طول مسیر استفاده گردد آنگاه مسیریاب‌ها بایستی به مبادله و پردازش تعداد قابل توجهی پیام جهت توافق بر روی یک شماره VC مشترک بپردازند که از سوی دیگر ارتباط‌های VC موجود در این مسیریاب‌ها مورد بهره‌برداری برای یک ارتباط قرار نمی‌گیرد.

در یک شبکه‌ی مدار مجازی، مسیریاب‌های شبکه مجبور به نگهداری **اطلاعات وضعیت ارتباط** [1] جهت ارتباط‌های در جریان می‌باشند. هر گاه یک ارتباط جدید در فراسوی یک روتر ایجاد می‌شود، یک ورودی جدید بایستی به جدول گسیل به جلو اضافه گردد و هرگاه یک ارتباط آزاد می‌شود اطلاعات مربوط به آن بایستی از جدول حذف گردد. بدیهی است حتی اگر مکانیزم تفسیر شماره VC وجود نداشته باشد هنوز نیاز به نگهداری اطلاعات وضعیت ارتباط که شماره VCها را در ارتباط با شماره‌ی

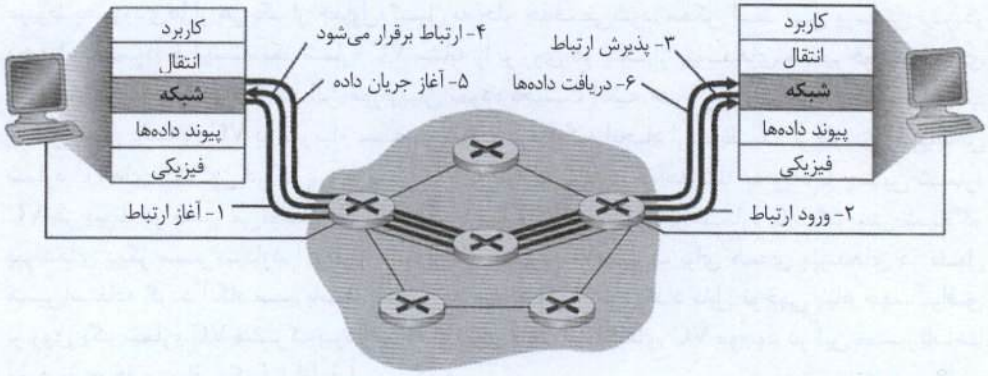
1- Connections information state

واسط‌ها قرار دهد ضروری می‌باشد. موضوع مربوط به اینکه آیا مسیریاب اطلاعات وضعیت ارتباط را جهت هر ارتباطی که در جریان است نگهداری می‌کند یا خیر بسیار حائز اهمیت است و بارها در این کتاب به آن مراجعه شده است. جهت هر مدار مجازی (VC) سه مرحله‌ی متمایز، قابل شناسایی است. این مراحل عبارتند از:

۱- **ایجاد مدار مجازی VC.** در جریان مرحله‌ی برقراری مدار مجازی، لایه‌ی انتقال فرستنده با لایه‌ی شبکه تماسی را برقرار می‌نماید که از طریق آن آدرس گیرنده را مشخص می‌کند، سپس به انتظار شبکه جهت ایجاد مدار مجازی می‌نشیند. لایه شبکه آنگاه مسیر بین فرستنده و گیرنده که عبارت است از مجموعه‌ای از پیوندها و مسیریاب‌ها که همه‌ی بسته‌ها از آن‌ها عبور می‌نمایند را تعریف می‌کند. لایه‌ی شبکه هم چنین شماره VC هر پیوند در طول مسیر را بدست می‌آورد و سرانجام لایه‌ی شبکه اطلاعات مربوط به شماره VC را وارد جدول گسیل به جلو در هر مسیریاب در طول مسیر می‌نماید. به هنگام ایجاد مدار VC لایه‌ی شبکه ممکن است منابع طول مسیر VC را که شامل پهنای باند (سرعت) و نظایر آن می‌شود از پیش ذخیره (رزرو) نماید.

۲- **انتقال داده.** همانگونه که در شکل ۴-۴ نشان داده شده است، هنگامیکه یک مدار VC ایجاد می‌شود بسته‌ها می‌توانند شروع به حرکت در مسیر VC نمایند.

۳- **قطع ارتباط مدار مجازی.** این عملیات هنگامی فعال می‌شود که فرستنده و یا گیرنده لایه‌ی شبکه را از قصد خود جهت خاتمه‌ی VC آگاه می‌کند. لایه ی شبکه سپس سیستم انتهایی طرف دیگر شبکه را آگاه نموده و جداول گسیل به جلو در هر یک از مسیریاب‌های بسته را در مسیر به روز رسانی نموده و از این طریق عدم ادامه‌ی حیات مدار VC را به آگاهی آن‌ها می‌رساند.



شکل ۴-۴ ♦ ایجاد ارتباط مدار مجازی

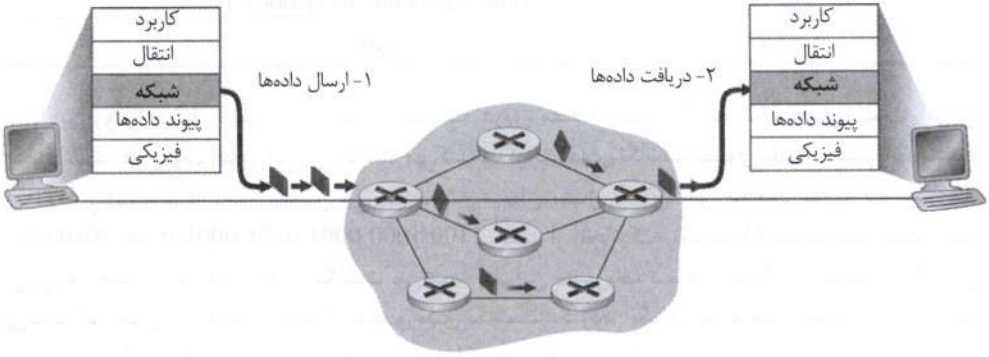
بدیهی است تمایز مهمی بین ایجاد یک ارتباط VC در لایه‌ی شبکه و معادل آن در لایه‌ی انتقال می‌باشد. به عنوان مثال دست‌دهی سه جانبه‌ی TCP که در فصل ۳ مورد مطالعه قرار گرفت از جمله‌ی این موارد است. ایجاد ارتباط در لایه‌ی انتقال شامل فقط دو سیستم انتهایی است. در هنگام ایجاد ارتباط لایه‌ی انتقال دو سیستم انتهایی به تنهایی پارامترهای ارتباط لایه‌ی انتقال خویش را مانند شماره ردیف اولیه و اندازه‌ی پنجره‌ی کنترل جریان بدست می‌آورند. اگرچه هر دو سیستم انتهایی آگاه

از ارتباط لایه‌ی انتقال می‌باشند اما مسیریاب‌های بین شبکه کاملاً از آن بی‌خبر می‌باشند. از طرف دیگر در یک ارتباط VC لایه‌ی شبکه مسیریاب‌های حاضر در بین مسیر دو سیستم انتهایی در ایجاد چنین ارتباط VC سهیم بوده و هر روتر کاملاً از وضعیت تمامی VC های عبوری از درون خود آگاه می‌باشد.

پیام‌هایی که سیستم‌های انتهایی جهت شروع و یا خاتمه‌ی یک VC به درون شبکه گسیل می‌کنند و نیز پیام‌هایی که بین مسیریاب‌ها جهت ایجاد VC برای تغییر و به روزرسانی وضعیت ارتباط در جداول آن‌ها مبادله می‌شود، در واژگان شبکه‌های کامپیوتری **پیام‌های سیگنالینگ (علامت‌دهی)** نام‌گذاری شده‌اند و پروتکل‌های به کار گرفته جهت مبادله‌ی این پیام‌ها نیز به **پروتکل‌های سیگنالینگ** معروف شده‌اند.شکل ۴-۴ ایجاد VC را به گونه‌ای تصویری نشان می‌دهد.

- **شبکه‌های داده‌گرام (DGN)**

در شبکه‌های داده‌گرام هرگاه یک سیستم انتهایی مایل به ارسال به یک بسته می‌باشد. بسته را از طریق آدرس مقصد سیستم انتهایی نشان‌گذاری می‌نماید و سپس آن را به درون شبکه تزریق می‌کند. بر اساس آنچه در شکل ۴-۵ نشان داده شده است این فرآیند بدون ایجاد ارتباط VC صورت می‌گیرد. مسیریاب‌های موجود در یک شبکه‌ی داده‌گرام هیچگونه اطلاعات وضعیت از VCها را نگهداری نمی‌کنند زیرا مدار مجازی وجود ندارد. به هنگام انتقال یک بسته از مبدا به مقصد آن بسته از میان مجموعه‌ای از مسیریاب‌ها عبور می‌نماید و هر یک از این مسیریاب‌ها با بهره‌گیری از آدرس مقصد بسته آن را به سمت جلو هدایت می‌کنند. هر مسیریاب دارای یک جدول گسیل به جلو می‌باشد که آدرس مقصد را به واسط‌های پیوند نگاشت می‌کند. هنگامیکه یک بسته وارد مسیریاب می‌شود با استفاده از آدرس مقصد بسته، روتر به جستجوی واسط پیوند خروجی مناسب برای آن بسته در جدول گسیل به جلو می‌پردازد. مسیریاب آنگاه بسته را با نیت قبلی به سمت جلو به آن واسط پیوند خروجی هدایت می‌کند.



شکل ۴-۵ ♦ شبکه‌ی داده گرام

جهت درک بهتر از فرآیند عملیات جستجو به مثالی که در ادامه‌ی بحث آمده است دقت نمائید. اگر فرض شود همه‌ی آدرس‌های مقصد، 32 بیت طول داشته باشند. بدیهی است این واقعیت در رابطه با آدرس‌های IP در داده‌گرام‌های اینترنت کاملاً درست می‌باشد. با استفاده از یک پیاده‌سازی با بهره‌گیری از الگوریتم ناآگاهانه جداول گسیل به جلو دارای یک ورودی برای هر آدرس مقصد ممکن

خواهند شد. از آنجائیکه بیش از 4 میلیارد آدرس می تواند تعریف شود امکان پیاده‌سازی این تعداد چشمگیر از آدرس میسر نمی‌باشد و نیازمند یک جدول گسیل به جلو به گونه‌ی کلان می‌باشد.

اکنون اگر مسیریابی در نظرگرفته شود که دارای 4 پیوند به شماره‌های صفر تا 3 باشد و بسته‌ها نیز بایستی به واسط پیوند گسیل به جلو شوند آنگاه:

محدوده‌ی آدرس مقصد	واسط پیوند
1100 1000 000101110001 0000 0000 0000 تا 1100 1000 0001011100010111 1111 1111	0
1100 1000 0001011100011000 0000 0000 تا 1100 1000 0001011100011000 1111 1111	1
1100 1000 0001011100011001 0000 0000 تا 1100 1000 0001011100011111 1111 1111	2
غیره	3

آشکار است برای این مثال لازم نیست که تعداد 4 میلیارد ورودی در جدول گسیل به جلو در مسیریاب تعریف گردد و به عنوان مثال جدول گسیل به جلو را می‌توان به تعداد چهار ورودی ذیل خلاصه نمود:

نگاشت پیش‌وند	واسط پیوند
1100 1000 0001 0111 0001 0	0
1100 1000 0001 0111 0001 1000	1
1100 1000 0001 0111 0001 1	2
غیره	3

با این روش جدول گسیل به جلو، مسیریاب آدرس مقصد بسته را به یک پیش‌وند از آدرس مقصد بسته در ورودی‌های جدول نگاشت می‌کند و اگر نتیجه‌ی نگاشت تساوی باشد، مسیریاب بسته را به یک پیوند وابسته به آن گسیل می‌نماید. به عنوان مثال اگر آدرس مقصد بسته به صورت: 1100 1000 0001 0111 0001 0110 1010 0001 باشد از آنجائیکه تعداد 21 بیت پیش‌وند این آدرس به نخستین ورودی جدول نگاشت می‌شود، بنابراین مسیریاب بسته را به واسط پیوند 0 گسیل می‌کند. اگر پیش‌وند به هیچ یک از سه ورودی نگاشت نگردد آنگاه مسیریاب بسته را به واسط پیوند شماره 3 گسیل می‌نماید. اگرچه این عملکرد ساده به نظر می‌رسد اما نکته‌ی ظریف و پیچیده‌ای در پس آن نهفته است. با اندکی تامل ملاحظه می‌شود این امکان وجود دارد که آدرس مقصد به بیش از یک ورودی در جدول نگاشت شود. به عنوان مثال اگر 24 بیت نخست آدرس به صورت: 1100 1000 0001 0111 0001 1000 1010 1010 باشد این بیت‌ها به ورودی دوم جدول نگاشت می‌شود. هنگامیکه تعداد نگاشت‌ها بیش از یک باشد مسیریاب از **قاعده‌ی نگاشت پیش‌وند**

طولانی‌تر[1] استفاده می‌کند. به عبارت دیگر بلندترین ورودی قابل نگاشت به جدول را گزینش و بسته را به سوی واسط پیوند وابسته به آن گسیل می‌نماید. هنگامیکه آدرس‌دهی اینترنت در همین فصل بررسی می‌شود علت این رویکرد بیشتر شناخته می‌شود.

اگرچه مسیریاب‌ها در شبکه‌های داده‌گرام به نگهداری اطلاعات وضعیت ارتباط نمی‌پردازند اما در هر حال اطلاعات مربوط به وضعیت گسیل به جلو را در جدول خود نگهداری می‌کنند. مقیاس زمانی تغییر اطلاعات وضعیت به جلو نسبتاً کند می‌باشد. در یک شبکه‌ی داده‌گرام تغییرات اطلاعات جدول گسیل به جلو از طریق الگوریتم‌های مسیریابی اعمال می‌گردد. این تغییرات به گونه‌ی ویژه‌ای در فاصله‌ی زمانی یک تا 5 دقیقه یک بار به جداول گسیل به جلو اعمال می‌شوند. در یک شبکه‌ی VC جدول گسیل به جلو در مسیریاب هنگامی تغییر می‌کند که یک ارتباط جدید از طریق روتر ایجاد شده باشد و یا آنکه ارتباط موجود از درون آن مسیریاب قطع گردد. این وضعیت در مقیاس زمانی یک میکرو ثانیه در شبکه‌ی زیر ساخت سطح اول در مسیریاب‌ها رخ می‌دهد.

از آنجائیکه جداول گسیل به جلو در شبکه‌های داده‌گرام در هر لحظه ممکن است تغییر نمایند بنابراین مجموعه‌ای از بسته‌ها از یک سیستم انتهایی به سیستم انتهایی دیگر ممکن است مسیرهای گوناگونی را از درون دنبال نموده و در نتیجه خارج از نوبت گسیل آن‌ها وارد مقصد شوند.

- **سرچشمه شبکه‌های مدار مجازی و داده‌گرام**

چگونگی تکامل شبکه‌های VC و داده‌گرام بازگو کننده‌ی سرچشمه‌ی آن‌ها می‌باشد. مفهوم مدار مجازی ریشه‌ی خود را از ارتباطات تلفنی که از مدارهای فیزیکی واقعی بهره می‌گیرند، گرفته است. از طرف دیگر مفهوم داده‌گرام ریشه در عوامل انتقال داده به عنوان عناصر اصلی اطلاعات دارد که به ارتباط کامپیوترها با یکدیگر به عنوان اجزا اصلی تولید داده نیازمند می‌باشد و از آنجائیکه نخستین امکانات انتقال داده از طریق خطوط تلگراف به عنوان تلگرام میّسر گردید بنابراین از واژه داده‌گرام در برابر تلگرام استفاده شد. به دلیل پیچیدگی روزافزون تجهیزات سیستم‌های انتهایی معماران شبکه‌ی اینترنت ضمن گزینش الگوی خدمات لایه‌ی شبکه برای این منظور تلاش در ساده نگهداشتن این الگو نموده‌اند. همانگونه که در فصل‌های ۲ و ۳ توضیح داده شد عملکردهای افزون‌تر مانند تحویل منظم، انتقال مطمئن داده، کنترل تراکم و بازیابی نام (DNS) در لایه‌های بالاتر سیستم‌ها پیاده‌سازی شده است. این موارد الگوی شبکه‌ی تلفنی را به نوعی وارونه با نتایج چشمگیر نشان می‌دهد:

۱- از آنجائیکه الگوی خدمات لایه‌ی شبکه در اینترنت حداقل ضمانت را شامل می‌شود بنابراین نیازهای اندکی را نیز به لایه‌ی شبکه تحمیل می‌نماید. این مهم سبب می‌شود ارتباط بین شبکه‌ها که از فناوری‌های لایه‌ی پیوند گوناگون بهره می‌گیرند ساده‌تر گردد. این فناوری‌ها معمولاً شامل رادیو، الیاف نوری، اترنت و یا ماهواره می‌باشند که مشخصات سرعت انتقال و اتلاف آن‌ها متفاوت است. ارتباط بین شبکه‌های IP با جزئیات کامل در ادامه‌ی مطالب همین فصل تحلیل می‌شوند.

۲- در فصل دوم ملاحظه گردید کاربردهایی مانند پست الکترونیک (e-mail)، وب و حتی DNS در میزبان‌هایی (سرورها) پیاده‌سازی شده‌اند که در نبش یا لبه شبکه حضور دارند. توانمندی افزایش خدمات جدید به سادگی از طریق پیوست یک میزبان به شبکه و تعریف یک پروتکل

1- Longest prefix matching rule

لایه‌ی کاربرد جدید مانند HTTP این امکان را برای کاربردهای جدیدی مانند وب در اینترنت در یک فاصله‌ی زمانی بسیار کوتاه امکان‌پذیر نموده است.

در فصل ۷ ملاحظه خواهد شد که در جامعه‌ی اینترنت مباحث فراوانی در ارتباط با چگونگی تکامل معماری لایه‌ی شبکه در جریان است تا پشتیبانی خدمات بی‌درنگ مانند امکانات چند رسانه‌ای را میسر نماید.

۴-۴ ساختمان درونی مسیریاب (روتر)

در بخش‌های قبلی عملکرد و خدمات لایه‌ی شبکه بررسی گردید. در ادامه چگونگی فعالیت به جلو در لایه‌ی شبکه بررسی می‌شود که از طریق آن انتقال واقعی بسته‌ها از پیوندهای ورودی مسیریاب به پیوندهای خروجی مناسب آن انجام می‌گیرد. در این بخش به اختصار معماری روتر جهت انتقال بسته‌ها بررسی می‌شود و تحلیل از واژگان به جلو و راهیابی از این نقطه‌نظر ارایه می‌شود. بر اساس معماری نشان داده شده در شکل ۴-۶ از یک مسیریاب، عناصر اصلی در ساختار یک روتر عبارتند از موارد چهارگانه‌ی ذیل:

۱- **پورت‌های ورودی.** عملیات گوناگونی از طریق پورت‌های (درگاه‌های)ورودی انجام می‌شود. از جمله عملیات مربوط به لایه‌ی فیزیکی است که مربوط به پیوند فیزیکی ورودی به مسیریاب می‌باشد و از طریق درگاه‌های ورودی و خروجی در دو طرف شکل ۴-۶ پیاده‌سازی می‌شود. علاوه بر این مورد عملیات لایه‌ی پیوند داده در شکل ۴-۶ در قالب چهارگوش‌های میانی در درگاه‌های ورودی و خروجی نیز نشان داده شده‌اند که در مسیریاب انجام می‌شود. این عملیات مورد نیاز لایه‌ی پیوند داده‌ها در طرف دیگر سیستم انتهایی دریافت کننده و در پیوند ورودی آن می‌باشد. به جز دو مورد اشاره رفته عملیات انجام شده در چهارگوش سمت راست درگاه ورودی و سمت چپ درگاه خروجی عبارت است از جستجو و گسیل به جلو به گونه‌ای که یک بسته به درون ساختار راهیاب درون روتر تزریق شده و در یک درگاه خروجی مناسب ظاهر می‌شود. بسته‌های کنترل مانند آن‌هایی که اطلاعات پروتکل مسیریابی را حمل می‌کنند از طریق درگاه ورودی به پردازشگر مسیریابی منتقل می‌شوند. در عمل درگاه‌های چندگانه همگی بر روی آنچه اصطلاحاً **کارت خط**[1] خوانده می‌شود در درون مسیریاب مجتمع می‌شوند.

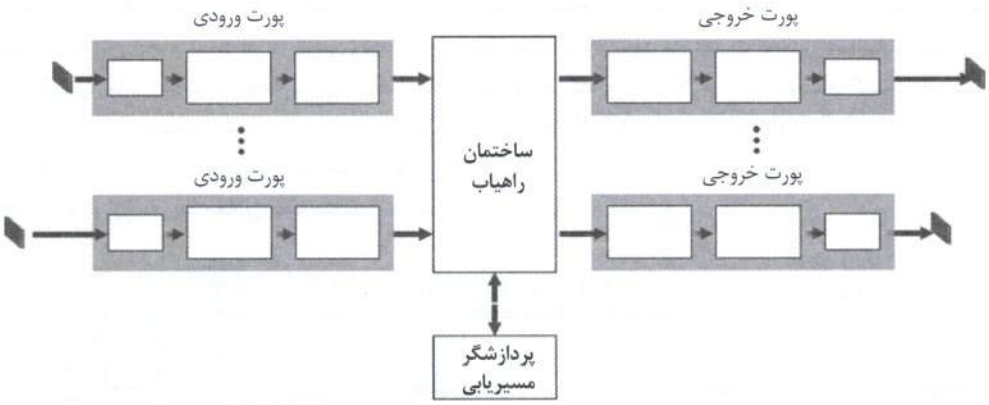
۲- **ساختمان راهیاب روتر.** عنصر راهیاب درگاه‌های ورودی مسیریاب را به درگاه‌های خروجی آن ارتباط می‌دهد. این بخش از ساختار در قالب یک نوع شبکه در درون مسیریاب سازمان‌دهی شده است.

۳- **درگاه‌های خروجی.** یک درگاه خروجی به ذخیره‌ی بسته‌های ارسال شده به آن از طریق راهیاب می‌پردازد و بسته‌ها را از سوی خود به پیوند خروجی روتر منتقل می‌کند. درگاه خروجی عکس عملیات آنچه در درگاه ورودی رخ می‌دهد را پیاده‌سازی می‌کند. هنگامیکه یک پیوند دو طرفه است به حمل ترافیک در دو سو می‌پردازد یک درگاه خروجی به پیوند در قالب یک زوج درگاه ورودی برای آن پیوند روی کارت خط مشابه قرار می‌گیرد.

۴- **پردازشگر مسیریابی.** پردازشگر مسیریابی پروتکل‌های مسیریابی را اجرا می‌کند. این

1- Line Card

پروتکل‌ها در بخش‌های بعدی همین فصل مطالعه می‌شوند. این پردازشگر همچنین اطلاعات مسیریابی و جداول گسیل به جلو را نگهداری و نیز عملیات مدیریت شبکه را از نقطه‌نظر عملیات درون مسیریاب انجام می‌دهند.



شکل ۴-۶ ♦ معماری مسیریاب

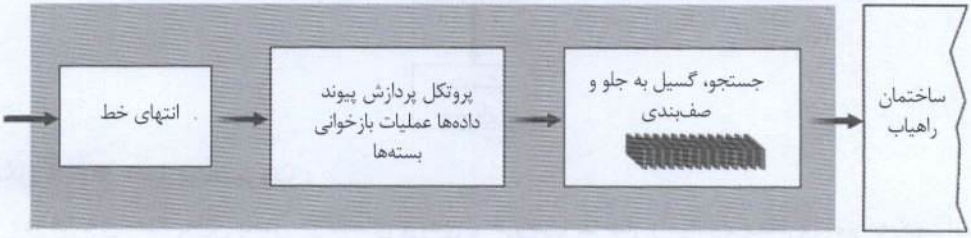
در ادامه‌ی مطالب این بخش درگاه‌های ورودی و خروجی، ساختار راهیاب و معماری روتر به همراه جزئیات بیشتر تحلیل شده‌اند. در مطالبی که به دنبال می‌آید فرض کلی بر آن است که شبکه‌های کامپیوتری صرفاً از نوع شبکه‌های داده‌گرام می‌باشند و بنابراین تصمیم‌گیری‌ها را بر مبنای اطلاعات آدرس مقصد نه شماره‌ی مدار VC آن‌ها انجام می‌دهند.

۴-۵ درگاه‌های ورودی

شکل ۴-۷ تصویر کلی از عملکرد درگاه ورودی را نشان می‌دهد. همانگونه که در مطالب پیشین عنوان شد بخش کاوش و گسیل به جلو در درگاه ورودی مهمترین عامل عملیات انتقال به جلو در یک مسیریاب به شمار می‌آید. در بسیاری از روترها در این نقطه از عملیات است که مسیریاب درگاه خروجی بسته‌های وارد شده به درون خود را مشخص و تعریف می‌کند. بدیهی است گسیل بسته‌ها از درگاه ورودی به خروجی با بهره‌گیری از اطلاعات جدول گسیل به جلو میّسر می‌شود. اگرچه این جدول همواره از سوی پردازشگر روتر محاسبه و به روزرسانی می‌شود اما همواره یک نسخه‌ی سایه از این جدول در هر درگاه ورودی ذخیره و از طریق پردازشگر مسیریاب در صورت نیاز به روزرسانی می‌شود. با قرار گرفتن جداول مسیریابی به گونه‌ی محلی تصمیم‌گیری های مورد نیاز نیز به صورت مستقل در هر درگاه ورودی بدون نیاز مراجعه به پردازشگر اصلی مسیریاب انجام می‌پذیرد. این رویکرد غیر متمرکز مانع از ایجاد گلوگاه‌های پردازش در یک نقطه از مسیریاب خواهد شد.

در مسیریاب‌هایی که از لحاظ توان‌مندی پردازش در درگاه‌های ورودی دچار کمبود می‌باشند، درگاه ورودی بسته‌ها را به سوی پردازشگر مرکزی مسیریاب روانه می‌کند و در آنجا عملیات مربوط به کاوش در جدول گسیل به جلو و سپس ارسال بسته به درگاه خروجی مناسب انجام می‌پذیرد. این رویکرد از سوی یک میزبان مشتری یا یک سرور که به عنوان مسیریاب عمل می‌نماید نیز صورت می‌پذیرد. در این حالت پردازشگر مسیریاب همان CPU در میزبان است و درگاه ورودی نیز یک کارت شبکه مانند کارت اترنت است.

با توجه به وجود جدول گسیل به جلو جستجوی درون آن به سادگی قابل تصور است. کافی است جدول مورد نظر با توجه به نگاشت طولانی‌ترین پیش‌وند بنابر آنچه در بخش ۴-۳ توضیح داده شده است مورد کاوش قرار گیرد. بدیهی است در عمل مسیریاب‌های زیر ساخت شبکه بایستی این عملیات را در سرعت فوق‌العاده بالا و در مقیاس چندین میلیون جستجو در هر ثانیه انجام دهند. در نتیجه لازم است پردازش انجام شده در درگاه ورودی در سرعت پیوند انجام شود به عبارت دیگر عملیات کاوش جدول در زمانی کمتر از آنچه برای دریافت یک بسته در درگاه ورودی لازم است، انجام گیرد. جهت درک بهتر نیاز عملکرد عملیات جستجو در یک پیوند مانند OC-48 با سرعت 2.5Gbps را تصور نمائید. اگر هر یک از بسته‌های وارد شده طولی معادل 256 بایت را در اختیار نمایند آنگاه سرعت جستجوی جدول بایستی تقریباً معادل یک میلیون جستجو در ثانیه باشد.



شکل ۴-۷ ♦ پردازش در گذرگاه ورودی مسیریاب

با توجه به نیاز سرعت بسیار بالا در پیوندهای فعلی فرآیند جستجوی خطی از درون یک جدول مسیریابی پر حجم غیر ممکن به نظر می‌رسد. راه کار منطقی و مطلوب آن است که ورودی‌های چنین جدولی در قالب یک ساختمان داده‌ی درختی ذخیره شوند. هر سطح از درخت را می‌توان در ارتباط با یک بیت در آدرس مقصد تصور نمود. بنابراین برای جستجوی یک آدرس کافی است از ریشه‌ی درخت عملیات جستجو آغاز و اگر نخستین بیت آدرس مقصد صفر می‌باشد آنگاه زیرشاخه‌ی سمت چپ شامل آدرس مقصد در جدول گسیل به جلو خواهد شد در غیر این صورت زیر شاخه‌ی سمت راست حامل اطلاعات آدرس مقصد است. زیر شاخه‌ی مربوط آنگاه با استفاده از دنباله‌ی بیت‌های آدرس مقصد پویش می‌شود. اگر بیت بعدی در آدرس صفر می‌باشد زیر شاخه‌ی فرعی سمت چپ گزینش می‌شود و در غیر این صورت زیر شاخه‌ی فرعی سمت راست. با این روش فرآیند جستجوی جدول گسیل به جلو به گونه N مرحله‌ای خواهد شد که N عبارت است از تعداد بیت‌های موجود در آدرس. بدیهی است از آنجائیکه درخت جستجوی دودویی (باینری) است فضای حالت جستجوی یک آدرس برابر 2^N خواهد شد.

بدیهی است حتی برای آدرس به طول 32 بیت مانند آدرس‌های IP در اینترنت سرعت جستجوی یک درخت باینری به اندازه‌ی کافی مطلوب نیازهای مسیریاب‌های زیر ساخت شبکه در حال حاضر نمی‌باشد. به عنوان مثال اگر فرض شود یک دستیابی به حافظه برای هر مرحله نیاز باشد آنگاه اندکی کمتر از یک میلیون جستجوی آدرس در هر ثانیه برای یک دستیابی حافظه‌ای با طول زمانی 40 نانو ثانیه بایستی انجام گیرد. بنابراین روش‌های متعددی جهت افزایش سرعت جستجوی آدرس‌ها پیشنهاد شده است.

حافظه‌های آدرس‌دهی توده‌ای[1] که اختصاراً CAM نامیده می‌شود این امکان را فراهم می‌آورد که یک آدرس 32 بیت مانند IP هنگام ارایه به CAM اطلاعاتی را از ورودی جدول گسیل به جلو برای آن آدرس بازگشت می‌دهد که تقریباً از لحاظ زمانی ثابت می‌باشد. مسیریاب‌هایی مانند سری 8500 از شرکت سیسکو از یک CAM با گنجایش 64k در هر درگاه ورودی بهره می‌گیرند.

روش دیگر تسریع در جستجوی اطلاعات جدول گسیل به جلو نگهداری آن گروه از ورودی‌های جدول در یک حافظه‌ی نهان است که به تازگی دستیابی شده‌اند. در این روش نگرانی اصلی حجم گنجایش بالقوه‌ی حافظه‌ی نهان است. ساختمان داده‌های سریع اجازه‌ی دستیابی به اطلاعات ذخیره شده در جدول را در قالب گام‌های Log(N) می‌دهند. یک رویکرد سخت‌افزاری جهت جستجوی آدرس‌های با طول 24 بیت و یا کمتر نیز پیشنهاد شده است. هنگامی که درگاه خروجی برای یک بسته مشخص شود آنگاه بسته به سوی شبکه‌ی راهیابی روتر گسیل می‌شود. بدیهی است ممکن است موقتاً از ورود بسته به درون شبکه‌ی راهیابی به دلیل آنکه بسته‌های متعلق به درگاه‌های ورودی دیگری در حال استفاده از آن شبکه می‌باشند جلوگیری به عمل آید. یک بسته‌ی مسدود شده بایستی در صف انتظار درگاه ورودی قرا گیرد و سپس در نوبت بعدی تعیین شده‌ای از شبکه‌ی راهیابی عبور نماید. موارد مربوط به انسداد، صف‌بندی و نوبت‌بندی عبور بسته‌ها در هر درگاه ورودی و خروجی در درون یک مسیریاب در بخش ۴-۷ تحلیل می‌شوند.

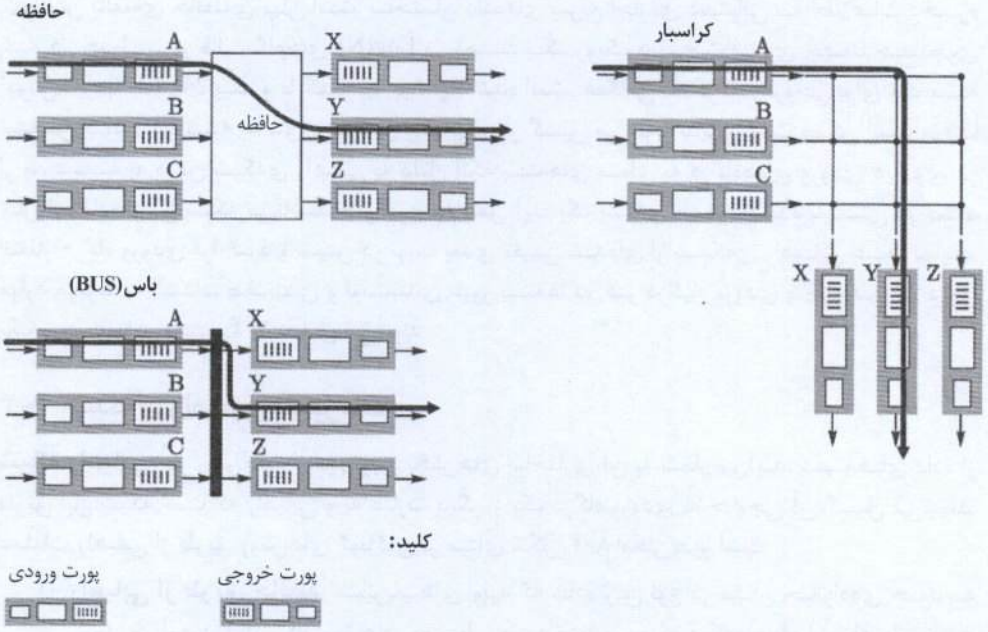
۴-۶ شبکه‌ی راهیابی مسیریاب

شبکه راهیابی[2] در روتر یکی از مهم‌ترین بخش‌های ساختاری آن به شمار می‌آید. بسته‌های داده از طریق این شبکه است که راهیابی و به عبارت دیگر از یک درگاه ورودی به خروجی آن گسیل می‌شوند. عملیات راهیابی از طریق روش‌های گوناگون بر مبنای شکل ۴-۸ امکان‌پذیر است:

۱- **راهیابی از طریق حافظه.** مسیریاب‌های اولیه که ساده‌ترین نوع در میان خانواده‌ی خود به شمار می‌آمدند. اغلب کامپیوترهای معمولی همه منظوره بودند که عملیات راهیابی بین درگاه‌های ورودی و خروجی با نظارت مستقیم CPU و به عبارت دیگر پردازشگر مسیریاب انجام می‌شد. در یک سیستم‌عامل سُنَتی درگاه‌های ورودی و خروجی به گونه‌ی تجهیزات عادی ورودی خروجی (I/O) عمل می‌نمودند. یک درگاه ورودی با دریافت نخستین بسته‌ی داده وقفه‌ای را به سوی CPU ارسال می‌نمود. بسته آنگاه از درگاه ورودی در حافظه‌ی پردازشگر نسخه برداری می‌شد، سپس CPU اطلاعات آدرس مقصد را از سرپیام بازشناسی می‌نمود و به کاوش پیرامون درگاه خروجی مناسب در جدول گسیل به جلو می‌پرداخت و آنگاه بسته‌ی نسخه‌برداری شده را به بافرهای درگاه خروجی منتقل می‌نمود. بدیهی است اگر سرعت حافظه‌ی مورد نظر به گونه‌ای باشد که در هر ثانیه تعداد B بسته را بتوان در آن نوشت و یا از آن خواند آنگاه نهفتگی کلی که عبارت است از سرعت انتقال بسته‌ها از درگاه‌های ورودی به خروجی بایستی کمتر از $\frac{B}{2}$ باشد. بسیاری از مسیریاب‌های مدرن نیز راهیابی را از طریق حافظه انجام می‌دهند. تفاوت عمده‌ی آن‌ها با روترهای قدیم در چگونگی جستجوی آدرس مقصد و سپس ذخیره‌سازی بسته‌ها در مکان‌های مناسب حافظه است که پردازشگرهای مستقر

1- Content Addressable Memories
2- Switching Networks

بر روی کارت‌های خطوط ورودی این مهم را انجام می‌دهند و CPU اصلی در مسیریاب عهده‌دار وظایف مهم‌تری می‌گردد. در اغلب مواقع مسیریاب‌هایی که راهیابی را از طریق حافظه انجام می‌دهند بسیار شبیه به چند پردازشگر با یک حافظه‌ی مشترک عمل می‌کنند. به عبارت دیگر پردازشگرهای مستقر بر روی کارت خطوط (پیوندها) راهیابی بسته‌ها را به حافظه‌ی مناسب درگاه خروجی هدایت می‌کنند. به عنوان نمونه راهیاب‌های سری: [Cisco 8500 2009] گسیل بسته‌ها را از طریق یک حافظه‌ی مشترک انجام می‌دهند.



شکل ۴-۸ ♦ روش‌های راهیابی در مسیریاب

۲- راهیابی از طریق گذرگاه (BUS). در این رویکرد درگاه‌های ورودی، یک بسته را مستقیماً به درگاه خروجی از طریق یک گذرگاه مشترک بدون دخالت پردازشگر مسیریاب انجام می‌دهند. توجه به این نکته ضروری است که در راهیابی از طریق حافظه، بسته باز هم از طریق گذرگاه به سوی حافظه‌ی سیستم و یا از آن عبور می‌نمود. اگرچه پردازشگر مسیریاب در عملیات انتقال گذرگاه شرکت ندارد اما به دلیل آنکه گذرگاه، مشترک مورد استفاده قرار می‌گیرد تنها یک بسته‌ی داده در هر زمان می‌تواند از طریق آن منتقل گردد. هنگامی که یک بسته‌ی داده به یک گذرگاه ورودی وارد می‌شود در صورتی که گذرگاه مشغول انتقال بسته‌ی دیگری در آن لحظه باشد آنگاه عبور بسته‌ی ورودی از درون شبکه‌ی راهیابی مسدود می‌شود و آن بسته در یک صف درگاه ورودی قرار می‌گیرد. از آنجائیکه هر بسته بایستی تنها از آن یک گذرگاه عبور نماید سرعت راهیابی بسته‌ها در مسیریاب محدود به سرعت گذرگاه می‌شود.

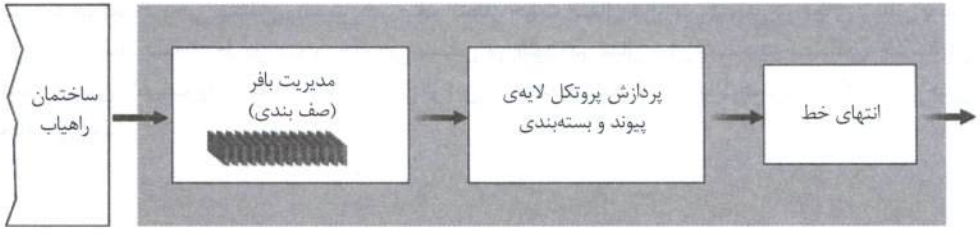
به دلیل آنکه در حال حاضر گذرگاه‌های با پهنای باند بیش از 1 Gbit امکان‌پذیر می‌باشد، راهیابی از طریق یک گذرگاه برای مسیریاب‌های مورد استفاده در اغلب شبکه‌های محلی کامپیوتری تجاری و غیره کافی می‌باشد. روش راهیابی از طریق گذرگاه در حال حاضر در بسیاری از مسیریاب‌ها مورد

استفاده قرار گرفته است. از جمله‌ی این مسیریاب‌ها سری [Cisco 5600,2009] می‌باشند که راهیابی بسته‌ها را از طریق یک گذرگاه 32 Gbps انجام می‌دهند.

۳- **راهیابی از طریق ارتباط بین شبکه‌ای.** یک روش جهت چیرگی بر دشواری محدودیت سرعت راهیابی در مسیریاب‌هایی که از یک گذرگاه مشترک بهره می‌گیرند از طریق ارتباط بین شبکه‌ای می‌باشد، مانند روش‌های گذشته که چندین پردازشگر در یک معماری کامپیوتر چند پردازشگری دارای ارتباط با یکدیگر می‌شدند. به عنوان نمونه یک **راهیاب تقاطعی**[1] یک ارتباط بین شبکه‌ای است که شامل تعداد 2^n گذرگاه است که n درگاه ورودی را به n درگاه خروجی بر اساس شکل ۴-۸ متصل می‌کند. یک بسته ی وارد شده در یک درگاه ورودی، پویش مسیر خود را از طریق گذرگاه افقی متصل شده به درگاه ورودی انجام می‌دهد تا آنکه با یک گذرگاه عمودی که آن را به درگاه خروجی مطلوب هدایت می‌کند برخورد نماید. چنانچه گذرگاه عمودی مرتبط با درگاه خروجی آزاد باشد بسته به آن درگاه منتقل می‌شود. در غیر این صورت اگر گذرگاه عمودی مشغول انتقال بسته‌ای از یک درگاه ورودی دیگر به همان درگاه خروجی باشد آنگاه بسته‌ی وارد شده مسدود و در صف درگاه ورودی قرار می‌گیرد. روش‌های راهیابی از نوع دِلتا و اُمگا نیز پیشنهاد شده‌اند و به عنوان مثال راهیاب سری [Cisco 12000 2009] از روش ارتباط بین شبکه‌ای با سرعت 60 Gbps بهره می‌گیرند. رویکرد اخیری که در طراحی روش ارتباط بین شبکه‌ای مورد مطالعه است، قطعه‌بندی یک بسته‌ی IP با طول متغیر در سلول‌های با طول ثابت می‌باشد که پس از آن که سلول‌ها دارای برچسب شده و از طریق عبور از ارتباط‌های بین شبکه‌ای به مقصد مطلوب خود هدایت می‌شوند. سلول‌ها آنگاه در قالب بسته‌ی اصلی در درگاه خروجی سازماندهی می‌شوند. بهره‌گیری از سلول‌های با طول ثابت به همراه برچسب‌های داخلی تسریع راهیابی بسته‌ها را از درون ارتباط بین شبکه‌ای به همراه دارد.

۴-۷ درگاه‌های خروجی

چگونگی پردازش درگاه خروجی در شکل ۴-۹ نشان داده شده است. در این فرآیند بسته‌هایی که در حافظه‌ی درگاه خروجی دخیره شده‌اند از طریق پیوند خروجی منتقل می‌شوند. پروتکل‌های پردازشگر پیوند و انتهای خط در شکل ۴-۹ تعامل لازم را با درگاه ورودی و دیگر بخش پیوند خروجی بر اساس آنچه در بخش ۴-۵ توضیح داده شد، انجام می‌دهند. عملیات مربوط به تشکیل صف و نیز مدیریت بافرهای مورد نیاز شبکه‌ی راهیابی روتر، جهت تحویل بسته‌ها به درگاه خروجی با سرعت بالاتر از نرخ پیوند خروجی می‌باشد. این موارد با جزئیات بیشتری در ذیل بررسی می‌شوند.



شکل ۴-۹ ♦ پردازش درگاه خروجی مسیریاب

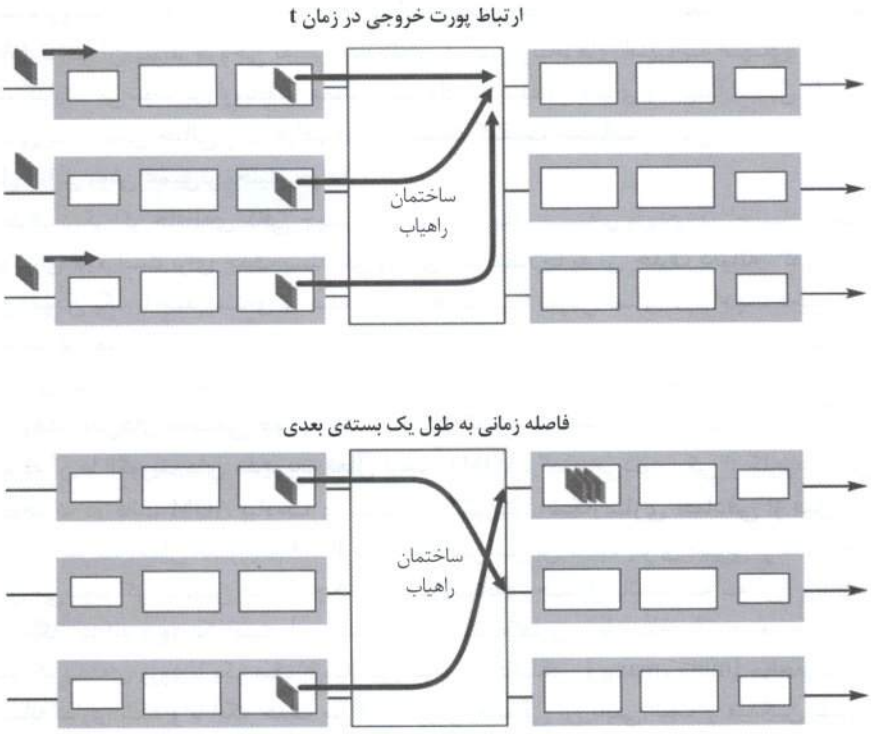
1- Crossbar Switch

- **مکان تشکیل صف**

بر اساس آنچه از شکل ۴-۸ می‌توان مشاهده نمود تشکیل صف‌های بسته می‌تواند در درگاه‌های ورودی و خروجی انجام شود. آنچه مهم است بررسی ساختار و عملکرد این صف‌ها با جزئیات بیشتر است، زیرا هرچه رشد این صف‌ها افزایش یابد فضای بافر مسیریاب دچار محدودیت شدید می‌شود و در نتیجه‌ی آن مفقودی بسته رخ می‌دهد. قبلاً اشاره گردید مفقودی و یا حذف بسته در بین عملیات شبکه چگونه انجام می‌گیرد. در واقع در این صف‌ها بسته‌ها حذف و یا تلف می‌شوند. مکان واقعی مفقودی بسته در صف‌های درگاه ورودی و یا خروجی می‌باشد و این رخداد در اثر بار ترافیک و سرعت سنجی شبکه‌ی راهیابی و نیز سرعت خط می‌باشد.

اگر فرض شود سرعت خط ورودی و خروجی برابر یکدیگر و تعداد درگاه‌های ورودی و خروجی نیز n باشد آنگاه سرعت شبکه‌ی راهیابی عبارت است از نرخ جابجایی بسته‌ها از درگاه ورودی و خروجی، اگر این سرعت به اندازه n مرتبه بیشتر از سرعت خط ورودی باشد آنگاه در درگاه ورودی صف تشکیل نخواهد شد. زیرا در بدترین حالت ممکن که همه‌ی تعداد n خط ورودی در حال دریافت بسته می‌باشند راهیاب قادر به انتقال n بسته از درگاه ورودی به خروجی در هنگامه‌ای است که هر n درگاه ورودی همزمان فقط یک بسته را دریافت می‌کند. اما در این شرایط چه وضعیتی برای درگاه‌های ورودی و خروجی روی می‌دهد؟ فرض کنید هنوز سزعت شبکه‌ی راهیابی n برابر سرعت خط باشد آنگاه در بدترین حالت ممکن بسته‌های وارد شده در هر یک از n درگاه ورودی به سمت درگاه خروجی مشابهی هدایت می‌شوند. در این حالت، در زمان دریافت و ارسال تنها یک بسته تعداد n بسته به درگاه خروجی وارد می‌شود. از آنجائیکه درگاه خروجی قادر به ارسال تنها یک بسته در هر لحظه از زمان می‌باشد یعنی زمان انتقال بسته، آنگاه تعداد n بسته ی ورودی بایستی در انتظار (صف) جهت انتقال از طریق پیوند خروجی قرار گیرند. این امکان وجود دارد که در همان لحظه تعداد n بسته‌ی بیشتر نیز وارد شوند و این اتفاق در زمانی روی دهد که عملیات انجام انتقال فقط یک بسته از n بسته ی از قبل در صف قرار گرفته و مشابه آن، در جریان است.

سرانجام تعداد بسته های در صف تا آن اندازه‌ای گسترش می‌یابد که فضای حافظه در درگاه خروجی عملاً پُر می‌شود و در نتیجه بایستی بسته‌های وارد شده حذف گردند. چگونگی تشکیل صف در درگاه خروجی در شکل ۴-۱۰ نشان داده شده است. در زمان t یک بسته به هر یک از درگاه‌های ورودی وارد شده و هر یک نیز به مقصد مطلوب‌ترین درگاه خروجی هدایت می‌شوند. اگر سرعت‌های برابر برای خط در نظر گرفته شود و یک راهیاب نیز با سرعت سه برابر نرخ خط عمل نماید. به اندازه‌ی یک یکای زمانی بعد و به عبارت دیگر در زمان لازم جهت ارسال و دریافت یک بسته همگی سه بسته‌ی اصلی به درگاه خروجی منتقل و سپس در صف انتظار جهت انتقال قرار می‌گیرند. در یکای زمانی بعد، یکی از این سه بسته از طریق پیوند خروجی منتقل می‌شود. در مثال بیان شده دو بسته‌ی جدید در سمت ورودی راهیاب وارد شده و گسیل یکی از این بسته‌ها به مقصد این مطلوب‌ترین درگاه خروجی انجام می‌شود.



شکل ۴-۱۰ ♦ صف‌بندی در درگاه خروجی

با توجه به نیاز برای بافرهای مسیریاب جهت جذب نوسان‌های حاصل در بار ترافیک، پرسش طبیعی آن است، مقدار بافر مورد نیاز چقدر است؟ محاسبه‌ی اندازه‌ی بافر (B) به گونه‌ی حاصل ضرب میانگینی از RTT (به عنوان مثال 250 ms) و ظرفیت پیوند (C) برای سال‌های طولانی در حال محاسبه است. این نتیجه بر مبنای تحلیلی از تغییرات صف نسبت به تعداد کمی از جریان‌های TCP بدست می‌آمد. به عبارت دیگر در یک پیوند 10 Gbps با RTT برابر 250 ms اندازه‌ی بافر مورد نیاز برابر است با:

$$B = RTT \cdot C \qquad (۱-۴)$$

در نتیجه در چنین شرایطی اندازه‌ی بافر B برابر 2.5 Gbit است. به طور کلی هنگامی که تعداد جریان‌های TCP در حال عبور از یک پیوند N باشد و اندازه‌ی N نیز قابل توجه در نظر گرفته شود آنگاه اندازه‌ی بافر مورد نیاز با استفاده از رابطه (۴-۲) محاسبه می‌شود.

$$B = RTT \cdot \times \frac{C}{\sqrt{N}} \qquad (۲-۴)$$

یک نتیجه‌ی حاصل از صف‌بندی در درگاه خروجی را می‌توان گزینش یک بسته از میان آن گروه از بسته‌ها که در صف قرار گرفته‌اند از سوی یک **زمان‌بند بسته**[1] در درگاه خروجی دانست. گزینش ممکن است بر یک مبنای ساده صورت پذیرد مانند زمان‌بندی از نوع **نخستین ورودی اولین جهت**

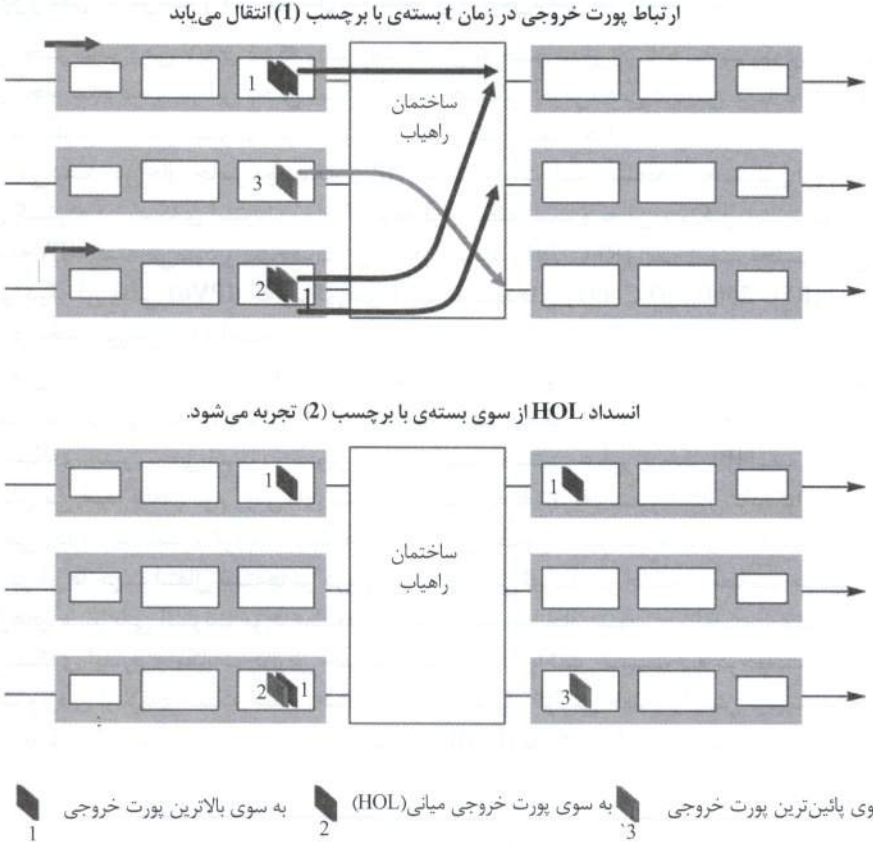
1- Packet Scheduler

دریافت سرویس[1] (FCFS) است و یا یک نظام زمان‌بندی پیچیده‌تر مانند **صف‌بندی توزین عادلانه**[2] (WFQ) که در آن پیوند خروجی به طرز عادلانه‌ای در بین ارتباط‌های انتها به انتهای گوناگون به اشتراک گذارده می‌شود. این ارتباط‌ها همواره بسته‌های به صف درآمده‌ای را جهت انتقال دارند. روش زمان‌بندی بسته نقش حیاتی را در فراهم آوری **ضمانت کیفیت خدمات** از سوی شبکه فراهم می‌آورد. در فصل ۷ این روش عمیق‌تر تحلیل خواهد شد.

از طرف دیگر اگر حافظه‌ی کافی جهت ذخیره‌سازی یک بسته‌ی ورودی در اختیار نباشد یک تصمیم‌گیری لازم است برای حذف بسته‌های ورودی که اصطلاحاً به آن **حذف دنباله**[3] گفته می‌شود و یا خارج نمودن یک یا چند بسته‌ی به صف درآمده جهت مهیا نمودن فضای لازم جهت ذخیره‌ی بسته های جدید. در بعضی موارد بهینه است که یک بسته حذف گردد و یا آنکه سرپیام آن جهت حذف نشانه‌گذاری شود قبل از آنکه بافر درگاه خروجی پُر شود و در نتیجه‌ی آن سیگنال تراکم به فرستنده گسیل گردد. روش‌های مختلفی جهت حذف و یا نشانه‌گذاری بسته‌ها تا کنون پیشنهاد شده اند که مجموعاً به آن‌ها الگوریتم‌های **مدیریت فعال صف**[4] (AQM) گفته می‌شود. یکی از کاربردی‌ترین این الگوریتم‌ها که در قالب AQM پیاده‌سازی شده است الگوریتم **آشکارسازی تصادفی از قبل**[5] (RED) می‌باشد. در شرایط اعمال الگوریتم RED یک میانگین توزین شده برای طولی از صف خروجی نگهداری می‌شود. اگر میانگین صف کوچکتر از یک آستانه‌ی حداقل باشد که به آن $\min_{th}$ گفته می‌شود آنگاه هنگام ورود یک بسته آن بسته در قالب نشانه‌گذاری و یا حذف شده تعریف می‌شود. سرانجام اگر بسته‌ی ورود با یک صف با میانگین طولی در فاصله‌ی $[\min_{th}, \max_{th}]$ مواجه گردد آنگاه بسته نشانه‌گذاری شده و یا آنکه حذف می‌گردد. ارزش احتمالی آن تابعی است از میانگین طول صفِ $\min_{th}$ و $\max_{th}$.

چنانچه ساختار شبکه‌ی راهیابی در حدّ کافی نسبت به سرعت‌های خط ورودی سریع نباشد که همه‌ی بسته‌های ورودی را بدون تاخیر از درون شبکه‌ی راهیابی خود انتقال دهد آنگاه صف‌بندی بسته می‌تواند در درگاه‌های ورودی نیز روی دهد زیرا که بسته‌ها بایستی به صف‌های درگاه ورودی جهت انتظار پیوسته، تا نوبت انتقال آن‌ها از درون شبکه‌ی راهیابی به سوی درگاه خروجی فرا رسد. جهت نشان دادن تاثیر مهم این صف‌بندی یک شبکه راهیابی کِراسبار (تقاطعی) را در نظر بگیرید که در آن (۱) سرعت همه‌ی پیوندها مشابه می‌باشد ،(۲) یک بسته می‌تواند از هر درگاه ورودی به یک درگاه خروجی در طولی زمان مشابه برای آنکه یک بسته بر روی یک پیوند ورودی دریافت گردد، انتقال یابد و (۳) بسته‌ها در قالب الگوریتم FCFS از یک صف ورودی مفروض به صف خروجی مطلوب خویش منتقل می‌شوند. انتقال موازی چندین بسته تا زمانیکه درگاه‌های خروجی آن‌ها متفاوت می‌باشند نیز میسّر است. در هر صورت اگر دو بسته در جلوی دو صف ورودی منتظر گسیل به صف خروجی مشابهی می‌باشند آنگاه یکی از این دو بسته مسدود شده و بایستی در صف ورودی بازهم در حالت انتظار قرار گیرد. شبکه‌ی راهیابی یک مسیریاب فقط می‌تواند یک بسته را در هر لحظه از زمان به یک درگاه خروجی انتقال دهد.

1- First Come First Serve
2- Weighted Fair Queuing
3- Drop Tail
4- Active Queue Management
5- Random Early Detection



شکل ۴-۱۱ ♦ انسداد سر خط در صف ورودی راهیاب

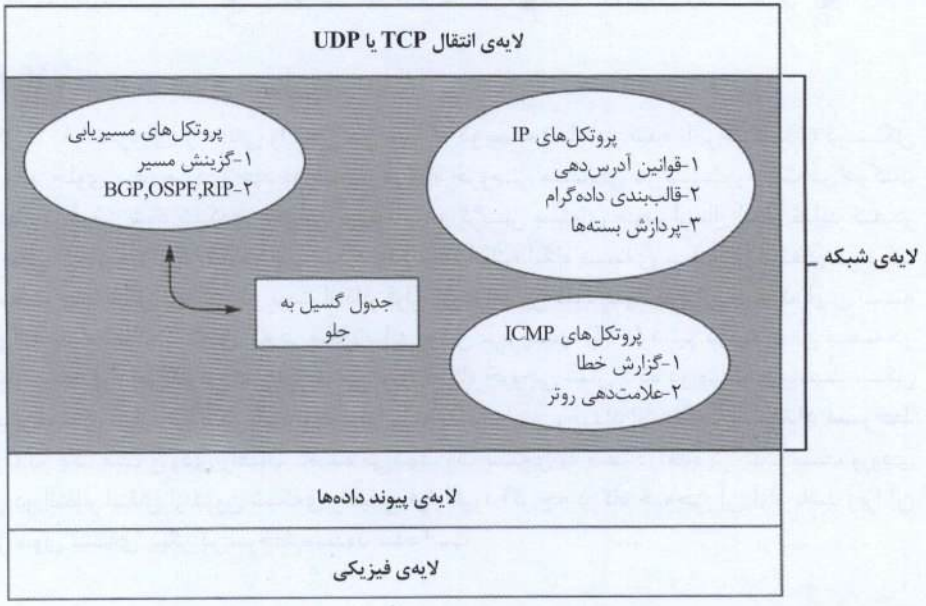
شکل ۴-۱۱ نمونه‌ای از حالتی را نشان می‌دهد که دو بسته (سایه‌دار شده با برچسب(1) در شکل (۴-۱۱) در جلوی صف ورودی خود به سمت درگاه خروجی مشابهی در سمت راست در حرکت می‌باشند. اگر فرض شود شبکه‌ی راهیابی مسیریاب به گزینش بسته‌ای جهت ارسال اقدام نماید که در جلوی صفی که در بالا و در سمت چپ شکل واقع شده است، آنگاه بسته‌ی سایه‌دار شده‌ی در صف پائین سمت چپ شکل بایستی در نوبت انتظار قرار گیرد. اما این فقط تنها کاری نیست که این بسته بایستی انجام دهد بلکه بسته‌ای که در شکل سایه روشن با برچسب (2) دارد نیز در پشت آن بسته در صف اشاره شده قرار می‌گیرد. حتی اگر مانعی برای درگاه خروجی نشان داده در میانه‌ی راست شکل که مقصد بسته‌ی اشاره شده می‌باشد وجود نداشته باشد. این رخداد اصطلاحاً **انسداد سرخط** (HOL)[1] در یک صف ورودی راهیاب نامیده می‌شود. یک بسته‌ی به صف درآمده در یک صف ورودی بایستی در انتظار انتقال از درون شبکه‌ی راهیابی قرار گیرد اگر چه درگاه خروجی آن آزاد باشد زیرا آن بسته از سوی بسته‌ی دیگر در سرِخط مسدود شده است.

1- Head of the line blocking

۴-۸ پروتکل اینترنت (IP): گسیل به جلو و آدرس‌دهی در اینترنت

تاکنون بحث آدرس‌دهی لایه‌ی شبکه و نیز روند گسیل به جلو بدون اشاره به شبکه‌ی کامپیوتری ویژه‌ای انجام شده است. در این بخش تمرکز بر روی چگونگی آدرس‌دهی در این لایه و نیز رویکرد گسیل به جلودر شبکه‌ی اینترنت خواهد بود. دو عنصر اشاره شده اجزای اصلی پروتکل اینترنت (IP) را تشکیل می‌دهند. در حال حاضر دو نسخه از IP ارایه شده است. ابتدا نسخه‌ی نخستین آن را که به گونه‌ی گسترده در شبکه‌ی اینترنت مورد استفاده قرار گرفته است و به آن پروتکل اینترنت نسخه‌ی چهارم (IPV4) گفته می‌شود و مشخصات آن نیز در مستند [RFC 791] آمده است، تحلیل می‌شود. نسخه‌ی دیگر آن یعنی (IPV6) که ویژگی‌های آن در مستندهای [RFC 2460 , RFC 491] آمده است در انتهای بخش بررسی شده است.

اما قبل از گام نهادن در بحث IP اندکی به عقب بازگشت نموده و عناصر تشکیلاتی لایه‌ی شبکه‌ی اینترنت را بررسی می‌کنیم. در شکل ۴-۱۲ سه عنصر مهم پروتکل IP که در واقع عناصر تشکیلاتی لایه‌ی شبکه‌ی اینترنت می‌باشند به تصویر در آمده است. نخستین عنصر پروتکل IP است که موضوع این بخش می‌باشد. عنصر دوم روش‌های مسیریابی است که به تعریف مسیر انتقال داده‌گرام از مبداء به مقصد می‌پردازد. قبلاً اشاره گردید پروتکل‌های مسیریابی جداول مسیر به جلو را محاسبه و از نتایج حاصل در آن‌ها جهت انتقال بسته‌ها در درون شبکه بهره می‌گیرند. در ادامه‌ی مطالب همین فصل پروتکل‌های مسیریابی اینترنت مورد مطالعه قرار می‌گیرند. سرانجام سومین و آخرین عنصر اساسی لایه‌ی شبکه‌ی اینترنت امکانات گزارش خطا در داده‌گرام‌ها و واکنش نسبت به درخواست‌ها جهت اطلاعات ویژه‌ای از لایه‌ی شبکه است. بحث مربوط به پروتکل گزارش خطا و واکنش نسبت به تقاضا در بخش مربوط به آن که پروتکل کنترل پیام اینترنت (ICMP) نام گرفته است، بررسی شده است.



شکل ۴-۱۲♦ تصویر درونی لایه‌ی شبکه‌ی اینترنت

• قالب‌بندی داده‌گرام

قبلاً اشاره شد که بسته ی تشکیل شده در لایه‌ی شبکه داده‌گرام خوانده می‌شود. مطالعه‌ی پیرامون IP را از نقطه‌نظر داده‌گرام IPV4 آغاز می‌کنیم. از آنجائیکه داده‌گرام نقش اساسی و مهمی را در اینترنت بر عهده دارد بنابراین ضروری است یک پژوهشگر و یا متخصص شبکه کاملاً آن را شناسایی، جذب و در بکارگیری آن مهارت ویژه‌ای را کسب نماید. قالب یک داده‌گرام از نوع IPV4– در شکل ۴-۱۳ نشان داده شده است. دامنه‌های کلیدی داده‌گرام IPV4 عبارتند از:



شکل ۴-۱۳ ♦ قالب داده‌گرام IPV4

۱- **شماره نسخه.** این دامنه‌ی به طول 4 بیت، نسخه‌ی پروتکل IP را تعریف می‌کند. با بررسی شماره نسخه‌ی یک مسیریاب شبکه می‌تواند به چگونگی تفسیر دیگر اطلاعات درون داده‌گرام IP بپردازد. نسخه های مختلف IP داده‌گرام‌های گوناگون را از لحاظ قالب‌بندی تعریف می‌کنند.

۲- **طول سرپیام.** از آنجائیکه یک داده‌گرام IPV4 شامل تعدادی متغیر در سرپیام خویش است این 4 بیت جهت تعیین آغاز واقعی یک داده‌گرام IP ضروری می‌باشد. اغلب داده‌گرام‌های IP این بخش از اطلاعات را در درون خود ندارند و بنابراین اندازه یا طول یک سرپیام داده‌گرام IP برابر 20 بایت است.

۳- **نوع سرویس.** بیت‌های متعلق به نوع سرویس (TOS) در سرپیام IPV4 اجازه‌ی تعریف انواع گوناگونی از داده‌گرام‌های IP را می‌دهند. با استفاده از اطلاعات درون این محدوده می‌توان داده‌گرام‌ها را از یکدیگر تفکیک نمود. این میدان در سرپیام به عنوان نمونه داده‌گرام‌هایی که نیازمند حداقل تاخیر در زمان انتقال، نهفتگی بالا و یا قابلیت اطمینان بالا می‌باشند را از دیگر داده‌گرام‌ها متمایز می‌کند. به عنوان مثال ممکن است، سودمند باشد که داده‌گرام‌های زمان واقعی و به عبارت دیگر بی‌درنگ مانند کاربرد تلفن IP از نوع غیر بی‌درنگ آن‌ها مانند FTP تفکیک شوند. بدیهی است سطح یا میزان سرویس فراهم آمده از طریق برنامه‌ریزی مسیریاب تعیین و اعمال می‌گردد. فصل ۷ این موضوع را بررسی می‌نماید.

۴- **طول داده‌گرام**. این محدوده به تعریف طول کلی داده‌گرام شامل سرپیام و محموله‌ی داده در آن می‌پردازد. از آنجائیکه طول این میدان 16 بیت است بنابراین حداکثر اندازه‌ی یک داده‌گرام IP می‌تواند برابر 65535 بایت باشد اگرچه در عمل داده‌گرام‌ها به ندرت دارای طولِ فراتر از 1500 بایت را اختیار می‌کنند.

۵- **شناسه‌گر، نشانه‌ها و متعادل‌کننده قطعه**[1]. این سه محدوده در ارتباط با عملیات قطعه بندی IP می‌باشند که در ادامه‌ی همین بخش بررسی شده است. لازم به یادآوری است که نسخه‌ی جدید IP یعنی IPV6 اجازه‌ی قطعه بندی در مسیریاب‌ها را نمی‌دهد.

۶- **زمان خروج (TTL)**. این میدان در سرپیام داده‌گرام IP به منظور جلوگیری از چرخش دایمی داده‌گرام‌ها در شبکه می‌باشد. این میدان هر مرتبه که داده‌گرام در یک مسریاب پردازش می‌شود با کاهش برابر یک رو به رو می‌باشد و بنابراین هنگامی که ارزش TTL به مقدار صفر می‌رسد داده‌گرام حذف می‌شود.

۷- **پروتکل**. از این محدوده هنگامی بهره‌برداری می‌شود که یک داده‌گرام IP وارد مقصد نهایی خود می‌شود. ارزش این محدوده مشخص کننده‌ی پروتکل لایه‌ی انتقال ویژه‌ای است که بخش داده از یک داده‌گرام بایستی عبور نماید.به عنوان مثال یک ارزش 6 به مفهوم عبور بخش داده از پروتکل TCP است، در حالیکه یک ارزش 17 به مفهوم عبور داده از پروتکل UDP است. برای مشاهده‌ی فهرست همه‌ی ارزش‌های ممکن به پروتکل IANA شماره‌ی 2012 مراجعه شود. لازم به یادآوری است که شماره‌ی پروتکل در داه‌گرام IP نقشی را بازی می‌کند که مشابه میدان شماره در درگاه (پورت) در قطعه های لایه‌ی انتقال است. شماره‌ی پروتکل همانند یک گیره پروتکل‌های لایه‌های انتقال و شبکه را به یکدیگر مرتبط می‌کند. این در حالی است که شماره‌ی درگاه همانند یک گیره دو لایه‌ی انتقال و کاربرد را به یکدیگر اتصال می‌دهد. در فصل ۵ نشان داده شده است که قالب لایه‌ی پیوند با استفاده از یک میدان ویژه لایه‌ی پیوند را به لایه‌ی شبکه مرتبط می‌کند.

۸- **کُد مجموع مقابله (Checksum) سرپیام**. از کُد مجموع مقابله‌ی سرپیام یک مسیریاب جهت شناسایی بیت‌های خطا در داده‌گرام دریافت شده استفاده می‌کند. این کد از طریق در نظر گرفتن هر 2 بایت در سرپیام به عنوان یک عدد و سپس جمع این اعداد با استفاده از مکمل یک آن‌ها محاسبه می‌شود. در بخش ۳-۶ توضیح داده شد که مکمل یکِ این حاصل جمع کد مجموع مقابله‌ی اینترنت نامیده می‌شود و در میدان مربوط به خود با همین نام ذخیره می‌شود. یک مسیریاب کد مجموع مقابله‌ی سرپیام را برای هر یک از داده‌گرام‌های IP دریافت شده محاسبه و وضعیت خطا را آشکارسازی می‌کند. اگر کد حمل شده در سرپیام داده‌گرام با آنچه مسیریاب محاسبه نموده یکسان نباشد آنگاه مسیریاب آن داده‌گرام را دچار خطا قلمداد نموده و از آن صرف‌نظر می‌کند. بدیهی است کد مجموع مقابله بایستی مجدداً محاسبه و در هر روتر ذخیره گردد، زیرا ارزش درون دو محدوده‌ی TTL و احتمالاً اختیار ممکن است تغییر نماید. مستند [RFC 1011] الگوریتم محاسبه‌ی سریع کد مجموع مقابله‌ی اینترنت را تشریح نموده است. در این نقطه از بحث ممکن است این پرسش مطرح شود چرا TCP/IP عملیات مربوط به تشخیص خطا را در هر دو لایه‌ی شبکه و انتقال انجام می‌دهد؟

1- Fragment Offset

چند علت را می‌توان برای این تکرار برشمرد. نخست آن که فقط سرپیام IP در لایه‌ی IP مجموع مقابله می‌شود در حالیکه مجموع مقابله در TCP/IP برای تمامیت قطعه‌های TCP و یا UDP محاسبه می‌شود. دوم دو پروتکل TCP و یا UDP و نیز IP لزوماً نبایستی هر دو متعلق به یک بسته ی پروتکلی باشند. پروتکل TCP در اصل می‌تواند بر روی پروتکل‌های گوناگون نظیر ATM به اجرا در آید در حالیکه IP می‌تواند به حمل داده‌هایی بپردازد که از TCP و یا UDP عبور نمی‌کنند.

۹- **آدرس‌های IP مبداء و مقصد.** هنگامی که یک مبداء داده‌گرام را خلق می‌کند، آدرس IP آن را در دامنه‌های آدرس IP مبدا و مقصد نهایی قرار می‌دهد. در اغلب مواقع میزبان آدرس مقصد را از طریق کاوش DNS بر اساس آنچه در فصل ۲ توضیح داده شد انجام می‌دهد. بحث مربوط به ادرس‌دهی IP در بخش ۴-۹ به تفصیل بررسی شده است.

۱۰- **دامنه‌ی اختیار (Option).** این دامنه توسعه و گسترش یک سرپیام IP را امکان‌پذیر می‌سازد. از این محدودیت به ندرت استفاده می‌شود و علت تعریف آن نیز از آنجا ناشی می‌شود که سرپیام‌های داده‌گرام ممکن است طول‌های متغیر اختیار کنند و از قبل نمی‌توان پیش‌بینی نمود که محدوده‌ی داده از کجا و کدام نقطه آغاز خواهد شد. همچنین آیا برخی از داده‌گرام‌ها نیازمند این فرآیند می‌باشند؟ و یا برخی دیگر نیازی به آن ندارند. زمان لازم برای پردازش یک داده‌گرام در یک مسیریاب می‌تواند بسیار متغیر باشد. رعایت همگی این موارد به ویژه برای پردازش IP در مسیریاب‌ها و میزبان‌های با کارایی بسیار بالا اهمیت دارد. به همین منظور این بخش از میدان در سرپیام IP در نسخه‌ی IPV6 حذف شده است.

۱۱- **محدوده‌ی داده یا محموله (Payload).** سرانجام مهم‌ترین محدوده‌ی حیاتی در یک داده‌گرام بخش داده‌ی اصلی آن است. در بسیاری از اوقات، میدان داده در داده‌گرام IP حامل قطعه‌ی TCP و یا UDP از لایه‌ی انتقال است که بایستی به مقصد تحویل گردد. یادآوری این نکته لازم است که محدوده‌ی محموله‌ی داده ممکن است انواع داده‌های دیگری غیر از TCP و یا UDP مانند ICMP را نیز حمل نماید.

به طور کلی ملاحظه گردید یک داده‌گرام IP جمعاً شامل 20 بایت بدون در نظر گرفتن محدوده‌ی اختیار در سرپیام است. چنانچه داده‌گرام به حمل یک قطعه‌ی TCP می‌پردازد آنگاه هر داده‌گرام که قطعه‌بندی نشده است مجموعاً دارای 40 بایت سرپیام شامل 20 بایت سرپیام IP و 20 بایت سرپیام غیر IP است که به همراه پیام لایه‌ی کاربرد آن را حمل می‌کند.

- **قطعه‌بندی داده‌گرام IP.**

در فصل ۵ ملاحظه خواهد شد که همگی پروتکل‌های لایه‌ی پیوند داده قادر به حمل بسته های لایه‌ی شبکه با اندازه‌های مشابه نیستند. برخی از پروتکل‌ها قادرند داده‌گرام‌های بزرگ از لحاظ طول را حمل نمایند در حالیکه برخی دیگر فقط می‌توانند داده‌گرام‌های کوچک حمل نمایند. به عنوان مثال قاب‌های اطلاعاتی اِترنت حداکثر قادرند 1500 بایت از داده را حمل نمایند در حالیکه قاب‌های اطلاعاتی برخی از پیوندهای گسترده (WAN) بیش از 576 بایت داده را نمی‌توانند حمل نمایند. حداکثر مقدار داده‌ی قابل حمل از سوی یک قاب لایه‌ی پیوند داده اصطلاحاً **حداکثر جزء انتقال**[1] **(MTU)** نامیده شده

1- Maximum Transmission Unit

است. از آنجائیکه هر داده‌گرام IP در بین قاب لایه‌ی پیوند بسته‌بندی جهت انتقال از یک مسیریاب به روتر دیگر می‌شود بنابراین MTU از پروتکل پیوند داده محدودیت قاطعی را بر اندازه‌ی طول یک داده‌گرام IP اعمال می‌کند. بدیهی است دشواری ممکن مربوط به اعمال این محدودیت نمی‌باشد بلکه علت آن چالشی است که از هر پیوند در مسیربین مسیریاب فرستنده و گیرنده نتیجه می‌شود و آن عبارت است از بهره‌گیری از پروتکل‌های پیوند لایه‌ی گوناگون که هر یک از این پروتکل‌ها می‌توانند دارای MTU مختلف نیز باشند.

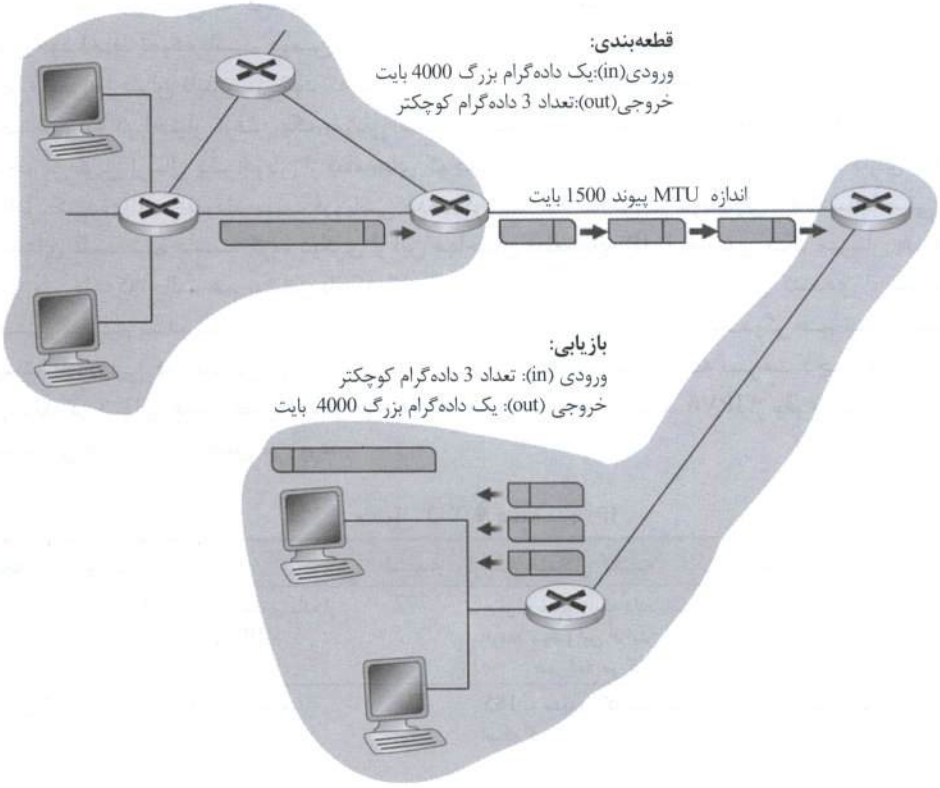
جهت درک بهتر موضوع تصور کنید شما مسیریابی باشید که چندین پیوند را به یکدیگر ارتباط می‌دهید و هر یک از این پیوندها پروتکل‌های لایه‌ی پیوند گوناگونی را با MTU های متفاوت به اجرا در می‌آورند. اگر فرض شود شما یک داده‌گرام IP را از یک پیوند دریافت نموده‌اید آنگاه ابتدا به ارزیابی جدول مسیریابی به جلو جهت تعیین پیوند خروجی می‌پردازید و این پیوندی است که MTU برای آن کوچکتر از طول داده‌گرام IP است. نگرانی در چگونگی فشرده‌سازی یک داده‌گرام IP است که بزرگتر از اندازه‌ی معمول است و به محدوده‌ی داده‌ای از قاب پیوند بایستی اعمال شود. راه‌کار در قطعه بندی داده به دو و یا چندین داده‌گرام IP کوچکتر است و بسته‌بندی هر یک از این داده‌گرام‌های کوچکتر به یک قاب مجزای لایه‌ی پیوند و گسیل این قاب‌ها از طریق پیوند خروجی. هر یک از این داده‌گرام‌های کوچکتر اصطلاحاً **قطعه**[1] نامیده شده است.

قطعه‌ها نیازمند سازماندهی مجدد قبل از رسیدن آن‌ها به لایه‌ی انتقال در مقصد می‌باشند. بدیهی است هر دو پروتکل TCP و UDP انتظار قطعه‌های کامل و تقسیم نشده را از لایه‌ی شبکه دارند. طراحان IPV4 بر این باور بودند که عملیات مربوط به سازماندهی مجدد قطعه در قالب داده‌گرام‌ها در مسیریاب‌ها سبب پیچیدگی چشمگیر در پروتکل‌ها و اعمال محدودیت در کارایی روترها می‌شود. بدیهی است اگر شما به جای یک مسیریاب می‌بودید حاضر به انجام این عملیات علاوه بر وظایف محوله‌ی دیگر نمی‌شدید. با توجه به اصل ساده نگهداشتن هسته‌ی شبکه تصمیم طراحان IPV4 بر آن شد که وظیفه‌ی سازماندهی مجدد قطعه‌ها به یک داده‌گرام تقسیم نشده را بر عهده‌ی سیستم‌های انتهایی شبکه و نه مسیریاب‌های شبکه بگذارند.

هنگامی که یک میزبان مقصد مجموعه‌ای از داده‌گرام‌ها را از یک منبع مشابه دریافت می‌کند آن میزبان نیازمند تعیین تکلیف این نکته است که آیا هر یک از این داده‌گرام‌ها قطعه‌ای از یک داده‌گرام بزرگتر هستند یا نه؟ اگر برخی از داده‌گرام‌ها قطعه‌هایی از یک داده‌گرام اصلی می‌باشند آن میزبان بایستی مشخص نماید چه هنگام آخرین داده‌گرام دریافت خواهد شد و چگونه داده‌گرام‌های دریافتی بایستی تبدیل به داده‌گرام اصلی خود شوند. جهت ایجاد امکانات لازم برای میزبان مقصد که آن را قادر به انجام این عملیات نماید طراحان IPV4 **محدوده‌های شناسه‌گر، نشانه‌ها (flags) و متعادل کننده‌ی قطعه** را در سرپیام داده‌گرام IP قرار داده‌اند. هنگامیکه یک داده‌گرام خلق می‌شود میزبان فرستنده داده‌گرام را دارای یک شماره شناسه و نیز آدرس مبدا و مقصد می‌کند. این شماره‌ی شناسه برای هر داده‌گرام ارسالی از سوی میزبان فرستنده هر بار با مقدار یک افزایش می‌یابد. هنگامیکه یک مسیریاب نیازمند قطعه‌بندی یک داده‌گرام است هر یک از داده‌گرام‌های نتیجه شده در اثر قطعه‌بندی با آدرس مبدا و مقصد و نیز شماره شناسه‌ی از داده‌گرام اصلی مقداردهی می‌شوند. هنگامی که میزبانی در مقصد مجموعه‌ای از داده‌گرام‌ها را از یک میزبان ارسال کننده‌ی مشابه دریافت نماید به ارزیابی

1- Fragment

شماره‌های شناسایی هر یک از داده‌گرام‌ها می‌پردازد تا مشخص نماید در حقیقت کدام یک از این داده‌گرام‌ها بخشی و یا بخش‌هایی از داده‌گرام مشابه بزرگتر است. از آنجائیکه پروتکل IP یک سرویس نامطمئن است یک یا چندین قطعه ممکن است هرگز به مقصد خود تحویل نگردند. به همین دلیل برای اطمینان میزبان گیرنده از دریافت آخرین قطعه از داده‌گرام اصلی معمولاً آخرین قطعه در میدان بیت نشانه‌ی خود دارای ارزش صفر می‌شود در حالیکه بقیه‌ی قطعه‌ها حامل ارزش یک در همان بیت می‌باشند. همچنین برای آنکه میزبان مقصد قادر باشد قطعه‌ها و در نتیجه داده‌گرام‌ها را به نظم درآورد و نیز از مفقود شدن آن‌ها آگاه شود از محدوده‌ی متعادل کننده‌ی قطعه جهت مشخص نمودن مکان مناسب قطعه در بین داده‌گرام IP اصلی استفاده می‌شود.



شکل ۴-۱۴ ♦ قطعه‌بندی و سازماندهی مجدد داده‌گرام IP

شکل ۴-۱۴ مثالی در این زمینه را به تصویر کشیده است. یک داده‌گرام به حجم 4000 بایت شامل 20 بایت سرپیام IP و 3980 بایت محموله‌ی IP به یک مسیریاب وارد شده است و بایستی از طریق یک پیوند یا MTU برابر 1500 بایت به جلو گسیل گردد. این به مفهوم آن است که 3980 بایت داده در داده‌گرام اصلی بایستی در سه قطعه هر کدام با ظرفیت مناسب که هر یک نیز یک داده‌گرام IP قلمداد می‌شود تفکیک گردند. فرض کنید داده‌گرام اصلی با شماره شناسه 777 مشخص شده باشد. مشخصات سه قطعه در جدول ۴-۲ توصیف شده است. ارزش‌های این جدول منعکس‌کننده‌ی این واقعیت است که اندازه‌ی محموله‌ی داده در همه‌ی قطعه ها به غیر از آخرین قطعه بایستی ضریبی از 8 بایت باشد و ارزش متعادل کننده در اجزاء 8 بایت تعریف گردد.

در مقصد محموله‌ی داده‌گرام فقط پس از آنکه لایه‌ی IP به گونه‌ی کامل داده گرام IP اصلی را بازسازی نمود تحویل لایه‌ی انتقال می‌شود. چنانچه یک یا چند قطعه وارد مقصد نگردند از داده‌گرام ناقص صرف‌نظر می‌شود و در نتیجه به لایه‌ی انتقال گسیل نمی‌شود. بدیهی است اگر از پروتکل TCP به عنوان لایه‌ی انتقال استفاده شود آنگاه TCP به بازیابی این مفقودی از طریق اجبار مبداء به انتقال مجدد داده‌ها در داده‌گرام اصلی اقدام می‌کند.

با توضیحات ارایه شده به اهمیت قطعه‌بندی IP به عنوان گیره‌ای که بسیاری از پروتکل‌های گوناگون لایه‌ی پیوند را به یکدیگر ارتباط می‌دهد، آشنا شدیم. بدیهی است قطعه‌بندی نیز هزینه‌ی ویژه‌ی خود را به همراه دارد. نخستین هزینه‌ی قطعه‌بندی پیچیده نمودن مسریاب‌ها و سیستم‌های انتهایی است که بایستی به گونه‌ای طراحی شوند که فرآیند قطعه‌بندی و بازسازی مجدد آن‌ها را در درون خود لحاظ نموده باشند. دومین هزینه عبارت است از آن که قطعه‌بندی سبب خلق تهاجم‌های DoS قطعی و جبران ناپذیر می‌شود که در آن مهاجم با ارسال تعدادی قطعه‌ی غیر منتظره و خیالی به هدف خود که در اختیار گرفتن یک میزبان است، می‌رسد. یک نمونه از این تهاجم‌ها Joltz بود که مهاجم از طریق ارسال یک جریان از قطعه‌های کوچک به میزبان هدف که هیچ یک از آن‌ها دارای یک متعادل کننده‌ی صفر نبودند سبب فروپاشی هدف گردید که تلاش در بازسازی داده‌گرام اصلی از بسته‌های فاسد شده داشت. گروه دیگری از این مهاجمان قطعه‌های IP را به سوی یک میزبان هدف نشانه می‌روند که حالت هم پوشانی با یکدیگر دارند به این مفهوم که ارزش متعادل کننده‌ی آن‌ها به گونه‌ای تنظیم شده است که قطعه‌ها نتوانند به صورت مناسب با یکدیگر هم تراز شوند. سیستم‌عامل‌هایی که آمادگی مقابله با این گونه تهدیدها را نداشته باشند به سرعت دچار فروپاشی می‌شوند. در انتهای همین فصل آشکار می‌شود که چگونه نسخه ی جدید IPV6 از وقوع چنین نقطه ضعف‌هایی جلوگیری به عمل می‌آورد.

جدول ۴-۲ ♦ قطعه‌بندی IP

قطعه	بایت	شناسه	ارزش متعادل کننده	نشانه
قطعه‌ی 1	1480بایت در محدوده‌ی داده از داده‌گرام IP	777	صفر به مفهوم آنکه داده بایستی افزوده شود و این افزایش از بایت صفر آغاز می‌گردد	یک به مفهوم داده گرام‌های بیشتر می‌باشد
قطعه‌ی 2	1480 بایت از داده	777	185 به مفهوم آنکه داده بایستی اضافه گردد و آغاز آن نیز از بایت 1480 است زیرا: $185 \times 8 = 1480$	یک به مفهوم داده گرام‌های بیشتر می‌باشد
قطعه‌ی 3	1020 بایت داده یعنی (3980 - 1480 –1480)	777	370 به مفهوم افزودن داده از آغاز بایت 2960 زیرا: $370 \times 8 = 2960$	صفر به مفهوم اینکه این آخرین قطعه است که گسیل می‌شود

۴-۹ آدرس‌دهی IPV4

در این بخش آدرس پروتکل IP نسخه چهارم تحلیل می‌شود. اگرچه ممکن است آدرس‌دهی و چگونگی انجام آن ساده به نظر رسد اما آدرس‌دهی اینترنت از ظرافت و پیچیدگی ویژه‌ای برخوردار است که نیاز به مهارت و دانش کافی را در نهان خود طالب است. قبل از ورود به بحث آدرس‌دهی IP ابتدا لازم است چگونگی اتصال میزبان‌ها و مسیریاب‌ها در شبکه‌های کامپیوتری بررسی شود. یک میزبان در اساس

فقط از یک پیوند در شبکه بهره می‌گیرد و هنگامی که IP در میزبان مایل به ارسال یک داده‌گرام است آن میزبان از طریق این پیوند عملیات گسیل داده‌گرام را انجام می‌دهد. مرز بین میزبان و پیوند فیزیکی اصطلاحاً **واسط**[1] نامیده می‌شود. اکنون یک مسیریاب و واسط‌های آن را در نظر بگیرید. از آنجائیکه وظیفه‌ی اصلی یک روتر دریافت داده‌گرام از طریق یک پیوند و گسیل آن به پیوندهای دیگر است، یک مسیریاب لزوماً دارای دو یا چندین پیوند است که به آن متصل شده‌اند. مرز بین مسیریاب و هر یک از پیوندهای آن نیز واسط نامیده می‌شود. یک مسیریاب بنابراین شامل چندین واسط است به عبارت دیگر یک واسط برای هر پیوند. از آنجائیکه هر میزبان و روتر قادر به ارسال و دریافت داده‌گرام‌های IP می‌باشند بنابراین پروتکل IP لزوماً از هر میزبان و مسیریاب می‌خواهد که برای هر واسط خود آدرس IP ویژه‌ی آن را معرفی نماید. بنابراین یک آدرس IP از نقطه‌نظر عملی وابسته به یک واسط است تا آنکه معرف میزبان و یا مسیریابی باشد که واسط شامل آن‌ها می‌باشد.

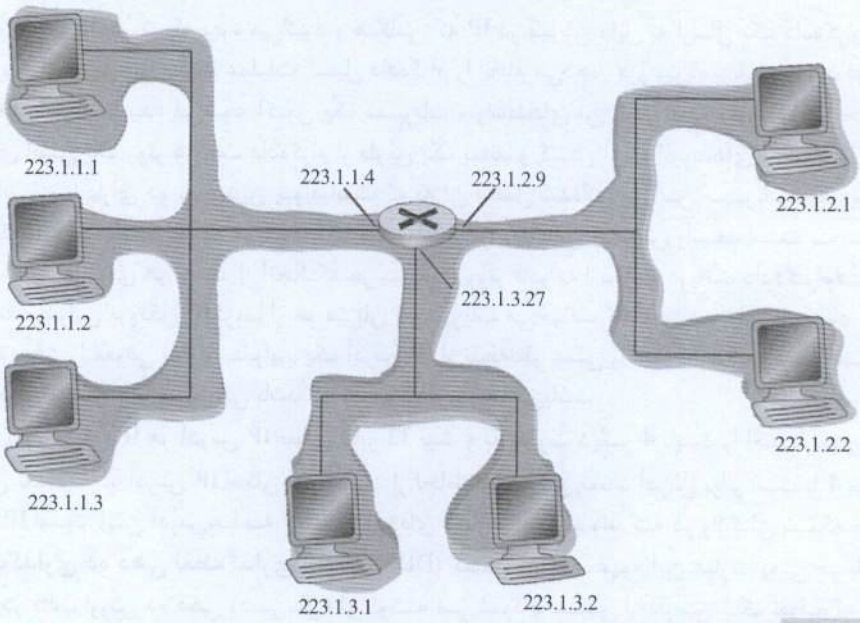
در پروتکل IPV4 هر آدرس IP طولی برابر 32 بیت و به عبارت دیگر 4 بایت را اختیار می‌کند. بنابراین تعداد 2^{32} آدرس IP امکان‌پذیر است . از لحاظ تقریب این تعداد آدرس برابر است با 4 میلیارد آدرس IP است. این آدرس‌ها به گونه‌ی ویژه‌ای نگارش می‌شوند که در واژگان شبکه به آن **علامت‌گذاری ده دهی نقطه‌گذاری شده**[2] (DDN) گفته می‌شود. مفهوم این عبارت یعنی هر بایت از آدرس در قالب ارزش ده دهی (دسی‌مال) آن نوشته می‌شود و سپس از طریق یک نقطه از دیگر بایت‌های آدرس متمایز می‌شود. به عنوان مثال اگر آدرس IP به صورت 193.32.216.9 در نظر گرفته شود، آنگاه ارزش 193 معادل ده دهی نخستین 8 بیت از آدرس IP می‌باشد. ارزش‌های بعدی مانند 32 ، 216 و 9 به ترتیب معادل ده دهی دومین، سومین و چهارمین 8 بیت از آدرس IP اشاره شده هستند. بنابراین آدرس 193.32.216.9 در قالب علامت‌گذاری دودویی یا باینری آن به صورت ذیل تعریف می‌شود:

1100 0001 0010 0000 1101 1000 0000 1001

هر واسط بر روی هر میزبان و مسیریاب در شبکه‌ی اینترنت جهانی بایستی دارای یک آدرس IP باشد که در سراسر گیتی منحصر به فرد است. بدیهی است این انفرادی و اختصاصی بودن آدرس IP در شبکه‌ی اینترنت برای واسط‌های نهان شده در NAT که در انتهای این بخش توضیح داده می‌شوند لازم به اجرا نمی‌باشد. بنابراین گزینش آدرس‌های IP تابع شرایط و قواعد ویژه‌ای می‌باشند. بخشی از یک آدرس IP برای واسط از طریق آنچه اصطلاحاً **زیر شبکه**[3] نامیده شده است انجام می‌پذیرد که واسط به آن زیر شبکه مرتبط است.

شکل ۴-۱۵ نمونه‌ای از واسط‌ها و آدرس‌های IP را به تصویر کشیده است. در این شکل یک مسیریاب با تعداد سه واسط جهت ارتباط با هفت میزبان مورد استفاده قرار گرفته است. با تأمل بر روی آدرس‌های IP اختصاص یافته به واسط‌ها، میزبان‌ها و مسیریاب‌ها چندین موضوع مهم آشکار می‌شود.

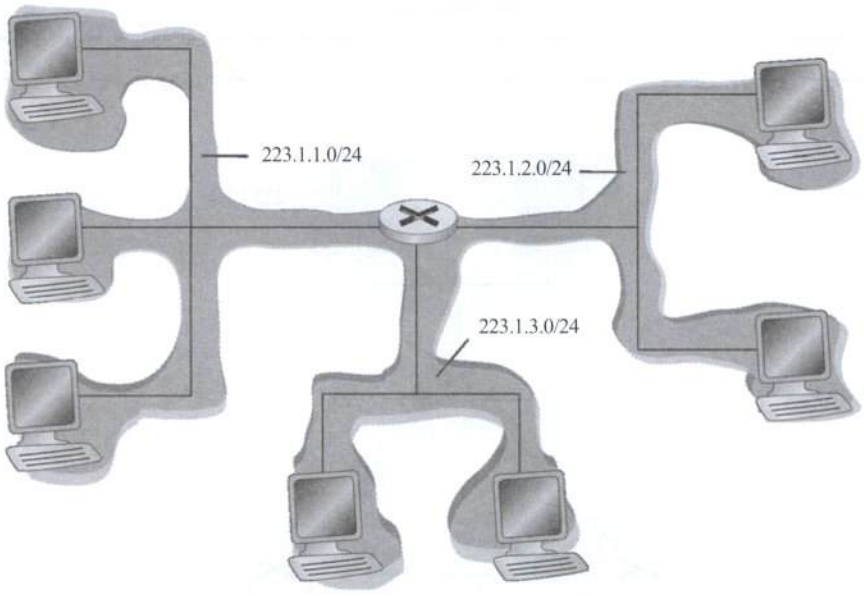
1- Interface
2- Dotted Decimal Notation
3- Subnets



شکل ۴-۱۵ ♦ واسط آدرس‌ها و زیر شبکه

سه میزبان بالا در سمت چپ تصویر ۴-۱۵ و واسط مسیریاب که این میزبان‌ها به آن متصل شده‌اند همگی دارای آدرس‌های IP به صورت: 223.1.1.xxx می‌باشند به عبارت دیگر 24 بیت سمت چپ همگی آن‌ها در آدرس IP مشابه می‌باشد. چهار واسط اشاره شده هم چنین به یکدیگر از طریق یک شبکه که شامل مسیریاب‌هایی نمی‌شود متصل شده‌اند. بدیهی است این شبکه‌ها به عنوان مثال می‌توانند شامل شبکه‌ی اِترنت محلی باشند که در این حالت واسط‌ها از طریق یک راهیاب اترنت و مانند آن به یکدیگر متصل می‌شوند. در واژگان IP شبکه‌ای مانند شبکه‌ی ۴-۱۵ که سه واسط میزبان و یک واسط مسیریاب را به یکدیگر ارتباط داده است زیر شبکه نامیده می‌شود. یک زیر شبکه بر اساس مستند [RFC 950] هم چنین یک **شبکه‌ی IP** نامیده می‌شود. آدرس‌دهی IP یک آدرس را به این زیرشبکه به صورت : 223.1.1.0/24 اختصاص داده است که علامت 24/ در واژگان اینترنت **پوشش (نقاب) زیر شبکه**[1] نامیده شده است و به مفهوم آن است که 24 بیت سمت چپ آدرس IP که 32 بیت می‌باشد مُعرف آدرس زیر شبکه است. زیر شبکه‌ی 223.1.1.0/24 بنابراین شامل سه واسط میزبان است که آدرس‌های آن‌ها عبارتند از: 223.1.1.1 ، 223.1.1.2 و 223.1.1.3 و آدرس واسط مسیریاب نیز 223.1.1.4 می‌باشد. هر میزبان اضافه شده به زیرشبکه‌ی 223.1.1.0/24 الزاماً بایستی یک آدرس اختیار نماید که شکل آن آدرس به گونه‌ی 233.1.1.xxx می‌باشد. دو زیرشبکه‌ی اضافی دیگر نیز در شکل ۴-۱۵ نشان داده شده است که به ترتیب آدرس‌های زیر شبکه‌ی آن‌ها عبارتند از: 223.1.2.0/24 و 223.1.3.0/24 . شکل ۴-۱۶ سه زیرشبکه‌ی حاضر در شکل ۴-۱۵ را نشان می‌دهد.

1- Subnet Mask



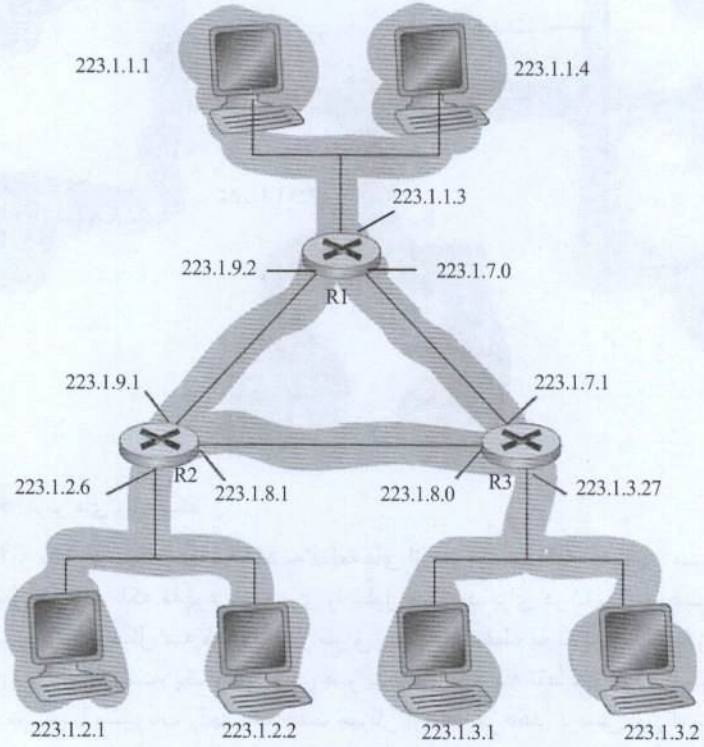
شکل ۴-۱۶ ♦ آدرس‌های زیر شبکه

تعریف IP از یک زیرشبکه تنها محدود به قطعه‌های اترنت نیست که چندین میزبان را به یک مسیریاب متصل می‌کند بلکه مفهوم عمیق‌تری را شامل می‌شود. برای درک بهتر موضوع شکل ۴-۱۷ را ملاحظه کنید در این شکل سه مسیریاب از طریق پیوندهای نقطه به نقطه به یکدیگر متصل شده‌اند. هر روتر دارای سه واسط است، یک واسط برای هر پیوند نقطه به نقطه و یک واسط برای پیوند فراپخشی که مستقیماً مسیریاب را به یک جفت میزبان ارتباط می‌دهد. پرسش این است که چه زیر شبکه‌هایی در این چیدمان حضور دارند؟ با اندکی مطالعه آشکار می‌شود سه زیر شبکه با آدرس‌های 223.1.1.0/24 ، 223.1.2.0/24 و 223.1.3.0/24 وجود دارند مشابه آنچه در شکل ۴-۱۵ نشان داده شده‌اند. اما در شکل ۴-۱۷ تعداد سه زیر شبکه‌ی اضافه نیز وجود دارند که یکی از این زیر شبکه‌ها با آدرس IP برابر 223.1.9.0/24 متعلق به واسط ارتباطی مسیریاب‌های R1 و R2 است و زیر شبکه‌ی دیگر با آدرس IP برابر 223.1.8.0/24 است که متعلق به واسط ارتباطی مسیریاب‌های R2 و R3 می‌باشد و سرانجام زیرشبکه‌ی سوم با آدرس 223.1.7.0/24 که به واسط ارتباطی R3 و R1 تعلق دارد. به طور کلی برای یک سیستم ارتباطی بین روترها و میزبان‌ها از عبارت ذیل می‌توان جهت تعریف زیر شبکه در یک سیستم بهره گرفت:

"به منظور یافتن زیرشبکه ابتدا هر واسط میزبان یا مسیریاب را مشخص نموده و از طریق آن‌ها جزایری از شبکه‌های کاملاً مستقل از یکدیگر به همراه واسط‌هایی که انتهای ارتباط آن‌ها سیستم‌های انتهایی شبکه است تشکیل می‌شود. هر یک از این شبکه‌های مستقل اکنون یک **زیرشبکه** است."

اگر این تعریف به سیستم‌های ارتباطی شکل ۴-۱۷ اعمال گردد تعداد زیرشبکه‌های حاصل برابر شش می‌شود. از موارد اشاره شده در بالا آشکار است یک نهاد مانند یک شرکت یا دانشگاه که دارای تعدادی قطعه‌ی اترنت و پیوندهای نقطه به نقطه است شامل چندین زیرشبکه می‌باشد. همه‌ی تجهیزات موجود در یک شبکه دارای آدرس زیرشبکه‌ی مشابه خواهند شد. زیرشبکه‌های گوناگون می‌توانند آدرس‌های زیرشبکه‌ی کاملاً گوناگونی نیز اختیار نمایند. بدیهی است در عمل در آدرس‌های

زیر شبکه‌ی آن‌ها به گونه‌ای گزینش می‌شوند که وجوه اشتراک بین آن‌ها وجود داشته باشد. جهت یافتن علت این مهم بر روی چگونگی روش آدرس‌دهی در شبکه‌ی اینترنت جهانی بررسی عمیق‌تری صورت می‌گیرد.



شکل ۴-۱۷ ♦ تعداد سه مسیریاب جهت ارتباط شش زیرشبکه

- **مسیریابی بین محدوده**

در شبکه‌ی اینترنت اختصاص آدرس‌های IP بر مبنای یک استراتژی صورت می‌گیرد که اصطلاحاً آن را **مسیریابی بین محدوده بدون کلاس[1] (CIDR)** نام نهادند. مستند [RFC 4632] ضمن تشریح مفصّل این روش آدرس‌دهی IP و مکانیزم‌های آن در شبکه‌ی اینترنت از آن به اختصار به عنوان **سایدر[2]** نام برده است. در ادامه‌ی مطالب این بخش بعضی از مواقع از CIDR و در پاره‌ای دیگر از اوقات از آن به عنوان سایدر نام برده می‌شود. به طور کلی سایدر یک گونه روش علامت‌گذاری آدرس‌های زیرشبکه می‌باشد. مانند آدرس‌های معمولی زیرشبکه تعداد 32 بیت آدرس IP به دو بخش تقسیم می‌گردد و از آن قالب ده دهی نقطه‌گذاری شده به صورت a.b.c.d/x جهت معرفی استفاده می‌شود. بدیهی است x/ به مفهوم تعداد بیت‌های متعلق به بخش نخست آدرس است. با ارزش‌ترین بیت‌های یک آدرس a.b.c.d/x همان x می‌باشد که شامل بخشی از آدرس IP است و در اغلب موارد از آن به

1- Classless Inter Domain Routing
2- Cider

عنوان **پیش‌وند**[1] و یا ارزش سایدر شبکه نام برده می‌شود. معمولاً در شبکه‌ی اینترنت به یک نهاد یک توده آدرس IP که شامل محدوده‌ای از آدرس‌ها با یک پیش‌وند مشترک است اختصاص می‌یابد. در این حالت آدرس‌های IP تجهیزات درون نهاد از پیش‌وند مشترکی بهره می‌گیرند. در ادامه‌ی مطالب این فصل ملاحظه خواهد شد که یک مسیریاب خارج از شبکه‌ی یک نهاد معمولاً فقط همین x بیت پیش‌وند را با اهمیت تلقی می‌کند. به عبارت دیگر هنگامی که مسیریاب خارج از شبکه‌ی نهاد، داده‌گرامی را به جلو گسیل می‌دارد که آدرس مقصد آن درون نهاد می‌باشد، فقط x بیت پیش‌وند آدرس را بایستی تحلیل نماید. این رویکرد سبب می‌شود جدول گسیل به جلو در این مسیریاب‌ها از لحاظ اندازه کاهش یابند زیرا یک ورودی به گونه‌ی a.b.c.d/x کافی است تا بسته‌ها را به هر مقصدی در بین نهاد گسیل به جلو نماید.

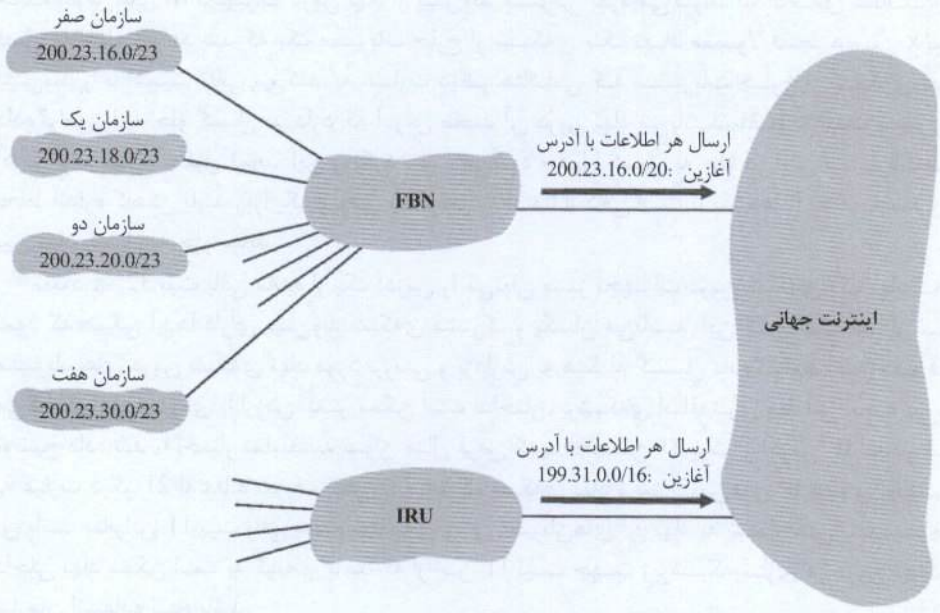
تعداد 32- x بیت باقی مانده از یک آدرس را می‌توان ممیز تجهیزات درون شبکه‌ی یک نهاد تصور نمود که همگی آن‌ها دارای پیش‌وند شبکه‌ی مشترک و یکسان می‌باشند. این گروه بیت‌ها از سوی مسیریاب‌های درون شبکه‌ی نهاد مورد بررسی و پردازش به هنگام گسیل داده‌گرام‌ها به جلو قرار می‌گیرند. این بیت‌های با ارزش کمتر ممکن است ساختار زیرشبکه‌ی اضافه‌تری مطابق آنچه در بالا توضیح داده شد را اختیار نمایند. به عنوان مثال فرض کنید نخستین 21 بیت از آدرس IP سایدر شود به عبارت دیگر a.b.c.d/21 تعریف پیش‌وند مشترک شبکه‌ی نهاد و نیز آدرس‌های IP همه‌ی تجهیزات آن باشد. بنابراین 11 بیت باقی مانده شناسه‌ی ویژه‌ی میزبان‌های آن نهاد به شمار می‌آیند. ساختار داخلی نهاد ممکن است به گونه‌ای باشد که از این 11 بیت جهت زیرشبکه‌سازی در درون نهاد یا سازمان استفاده شده باشد.

اکنون به یک مثال عملی از چگونگی این رویکرد می‌پردازیم. این مثال یک ISP را به تصویر می‌کشد که تعداد 8 سازمان را به اینترنت مرتبط می‌کند. اگر فرض شود بنابر آنچه در شکل ۴-۱۸ نشان داده شده است آن ISP به دنیای خارج از خود آگهی دهد که فقط داده‌گرام‌هایی را می‌پذیرد که نخستین 20 بیت آدرس آن‌ها با آدرس IP : 200.23.16.0/20 نگاشت می‌شوند. شبکه‌های خارج از دنیای این شبکه نیازی ندارند که از وجود 8 سازمان دیگر و شبکه‌ها و حتی زیرشبکه‌های آن‌ها در بین توده آدرس 200.23.16.0/20 آگاه شوند. این توانمندی در بهره گرفتن از یک پیش‌وند جهت آگهی دادن چندین شبکه و زیرشبکه در واژگان اینترنت به **اختصار کردن آدرس** و یا **توده کردن مسیر** و یا **اختصار کردن مسیر** معروف شده است.

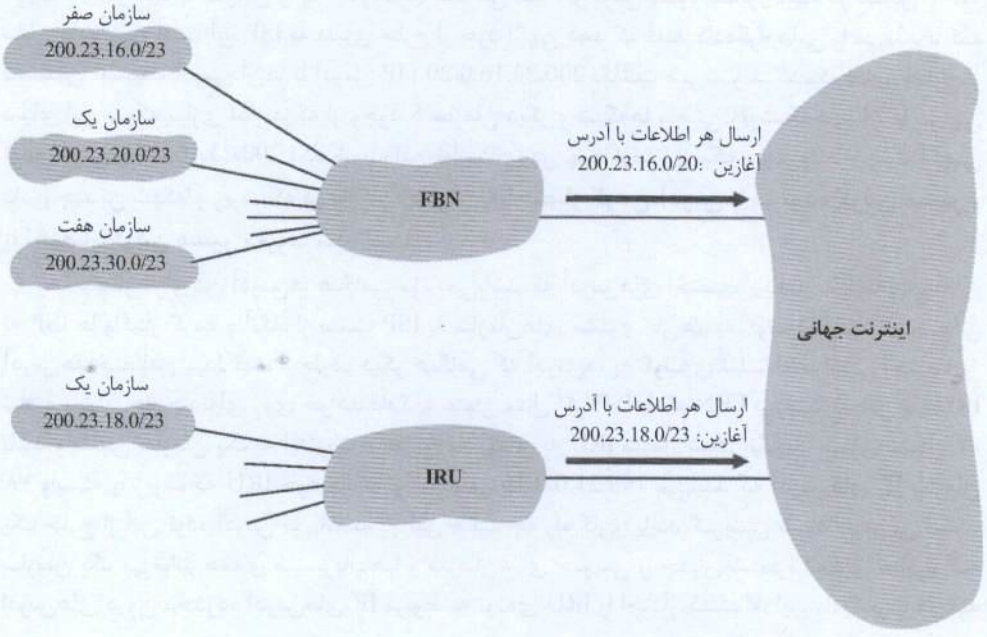
به اختصار درآوردن آدرس‌ها هنگامی مؤثر می‌باشد که آدرس‌های اختصاص یافته به گونه‌ی توده‌ای به ISP ها واگذار گردد و آنگاه از سمت ISP به سازمان‌های مشتری باز هم به گونه‌ی اختصار نمودن آدرس‌ها اختصاص پیدا کند. از طرف دیگر هنگامی که آدرس‌ها به گونه‌ی سلسله مراتبی اختصاص نیافته باشند چه حادثه‌ای روی خواهد داد؟ به عنوان مثال اگر ISP با نام FBN نیازمند تماس با IRU باشد و سپس سازمان یک به اینترنت از طریق وابسته‌ای به IRU متصل شود. آنگاه بر مبنای شکل ۴-۱۸ وابسته یا زیرشاخه IRU خود مالک توده آدرس 199.31.0.0/16 می‌باشد که آدرس‌های IP سازمان یک خارج از این توده آدرس می‌باشند. در این حالت چه راه کاری باید گزینش شود؟ بدیهی است سازمان یک می‌تواند همه‌ی مسیریاب‌ها و میزبان‌های خویش را چنان شماره‌گذاری نماید که ادرس‌های درون محدوده آدرس‌های IP مربوط به توده‌ی IRU را اختیار کنند. اما این راه کار پرهزینه

1- Prefix

است و سازمان یک ممکن است در آینده به زیر شاخه‌ی دیگری واگذار گردد. آنچه سازمان یک می‌تواند انجام دهد نگهداری آدرس‌های IP خود در محدوده‌ی آدرس 200.23.18.0/23 است.



شکل ۴-۱۸ ♦ آدرس‌دهی سلسله‌مراتبی و مسیردهی اختصاری



شکل ۴-۱۹ ♦ شبکه‌ی IRU دارای تعداد بیشتری از مسیرهای ویژه در سازمان یک

در این حالت مانند آنچه در شکل ۴-۱۹ نشان داده شده است شبکه‌ی FBN به آگهی توده آدرس 200.23.16.0/23 ادامه می‌دهد و IRU نیز همین عمل را با توده آدرس 199.31.0.0/16 انجام می‌دهد. اما IRU اکنون می‌تواند توده آدرس 200.23.18.0/23 متعلق به سازمان یک را نیز آگهی نماید. هنگامی که روترهای دیگر در مقیاس بزرگتر از شبکه‌ی اینترنت توده‌های آدرس 200.23.16.0/20 از FBN و 200.23.18.0/23 از IRU را ملاحظه می‌کنند و مایل هستند به انجام مسیریابی به یک آدرس در توده آدرس 200.23.18.0/23 بپردازند آنگاه آن‌ها از نگاشت طولانی‌ترین پیش‌وند بر اساس بخش ۳-۴ بهره می‌گیرند و مسیر را به سمت IRU همزمان با آگهی طولانی‌ترین پیش‌وند آدرس که با آدرس مقصد قابل نگاشت است، هدایت می‌کنند.

قبل از گزینش روش CIDR یا سایدر بخش شبکه‌ی یک آدرس IP محدود به 8 ، 16 و 24 بیت از لحاظ طول می‌شد، به این روش آدرس‌دهی اصطلاحاً **آدرس‌دهی کامل** گفته می‌شد زیرا زیرشبکه‌های با 8 ، 16 و 24 بیت آدرس زیرشبکه به ترتیب معروف به آدرس‌های کلاس A ، B و C بودند. این نیاز که بخش زیرشبکه از یک آدرس IP برابر یک، دو و یا سه بایت طول داشته باشد سبب دشواری‌هایی جهت پشتیبانی از تعداد روزافزون شبکه‌های سازمانی با زیرشبکه‌های کوچک و متوسط گردید. به عنوان مثال یک آدرس IP از نوع C که سایدر آن برابر (24/) است فقط می‌تواند حداکثر تا $2^8 - 2 = 254$ میزبان را پشتیبانی نماید. این تعداد آدرس IP میزبان برای بسیاری از سازمان‌های تجاری و غیره کافی نمی‌باشد. بدیهی است از میان 2^8 یعنی 256 میزبان دو آدرس IP آن عملاً جهت کاربردهای ویژه ذخیره می‌شدند. در آدرس‌دهی IP از نوع B از طرف دیگر ارزش CIDR برابر (16/) تعداد $2^{16} - 2 = 65534$ میزبان را پشتیبانی می‌نمودکه تعداد بسیار قابل توجهی به شمار می‌آمد و عملاً کاربردی نداشت. در شرایط آدرس‌دهی کامل به یک شرکت تجاری با تعداد 2000 میزبان آدرس‌های IP از نوع B اختصاص می‌یافت که عملاً تعداد زیادی از آدرس‌های قابل استفاده تلف می‌شدند. این موضوع سبب نقصان شدید در فضای آدرس‌دهی گروه B و بهره‌وری بسیار ناچیز از این گروه آدرس‌های IP بود. به عنوان نمونه در مثال بالا از میان 65534 آدرس IP قابل بهره‌برداری تنها 2000 آدرس آن مورد استفاده قرار می‌گیرند و تعداد 63000 آدرس IP دیگر عملاً بدون استفاده می‌باشند و از آن‌ها نیز نمی‌توان جهت شبکه‌های دیگری استفاده نمود.

علاوه بر همگی دلایل بالا نوع دیگر آدرس IP اصطلاحاً آدرس فراپخشی با شناسه‌ی 255.255.255.255 می‌باشد. هنگامیکه میزبانی یک داده‌گرام را با آدرس IP مقصد برابر 255.255.255.255 گسیل می‌نماید. این پیام به همه‌ی میزبان‌های همان زیرشبکه تحویل داده می‌شود. بدیهی است مسیریاب‌ها به صورت اختیاری این پیام را به زیرشبکه‌های مجاور نیز گسیل می‌نمایند. اکنون که آدرس‌دهی IP را با جزئیات آن مطالعه نمودیم لازم است چگونگی دستیابی میزبان‌ها و زیرشبکه‌ها را به این گونه آدرس‌ها بررسی نمائیم به عبارت دیگر چگونه یک سازمان یک توده آدرس IP مورد نیاز تجهیزات خود را تامین می‌کند؟ و چگونه یک وسیله مانند یک میزبان از میان آن توده آدرس دارای فقط یک آدرس IP می‌شود؟

• اخذ یک توده‌ی آدرس IP

جهت اخذ یک توده از آدرس‌های IP برای استفاده در درون یک زیرشبکه‌ی سازمانی معمولاً مدیر شبکه با ISP خود تماس برقرار می‌کند. معمولاً ISP آدرس‌هایی را از میان یک توده‌ی بزرگتر از آدرس‌هایی که در اختیار دارد به آن سازمان اختصاص می‌دهد. به عنوان مثال یک ISP ممکن است

خود دارای توده آدرس IP به صورت 200.23.16.0/20 باشد. این ISP می‌تواند توده آدرس خود را به 8 گروه آدرس مجاور و برابر یکدیگر تقسیم نموده و هر یک از این توده‌های برابر را به یک سازمان و حداکثر به تعداد 8 سازمان اختصاص دهد. بدیهی است همگی این آدرس‌ها از جانب ISP مورد نظر پشتیبانی می‌شوند. این تقسیم‌بندی می‌تواند در قالب آنچه در جدول ۴-۳ آمده است صورت پذیرد. در این قالب آدرس زیرشبکه‌ها با خطی در زیر آن‌ها مشخص شده‌اند:

جدول ۴-۳ ♦ تقسیم‌بندی آدرس‌های IP از بین یک توده آدرس

نام	گروه آدرس IP	آدرس و زیرشبکه
توده آدرس ISP	200.23.16.0/20	11001000 00010111 00010000 00000000
سازمان 1	200.23.16.0/23	11001000 00010111 00010000 00000000
سازمان 2	200.23.18.0/23	11001000 00010111 00010010 00000000
سازمان 3	200.23.20.0/18	11001000 00010111 00010100 00000000
سازمان 4	---	---
سازمان 5	---	---
سازمان 6	---	---
سازمان 7	---	---
سازمان 8	200.23.30.0/23	11001000 00010111 00011110 00000000

اگرچه تامین یک مجموعه از آدرس‌های IP از طریق یک ISP یکی از روش‌های ممکن می‌باشد اما تنها راه نیست. آشکار است که ISP نیز توده آدرس‌های متعلق به خود را از طریق دیگری به دست آورده است. بنابراین پرسش مطرح آن است که آیا می‌تواند یک نهاد جهانی جهت اختصاص و مدیریت فضای آدرس‌های IP به دیگران وجود داشته باشد؟ پاسخ یقیناً آری است. چگونگی مدیریت و اختصاص توده آدرس‌های IP سال‌ها است که از طریق یک نهاد بین‌المللی به نام سازمان بین‌المللی اختصاص نام و شناسه یا به اختصار (ICANN)[1] انجام می‌شود. بر مبنای مجموعه‌ی چهارم دستورالعمل‌هایی که در مستند [RFC 2050] تشریح شده‌اند چگونگی اختصاص اسامی و شناسه‌ها به شبکه‌های وابسته به اینترنت تعریف شده است. نقش اساسی این سازمان غیر انتفاعی فقط اختصاص آدرس‌های IP نیست بلکه مدیریت محدوده نام‌های DNS را در سرورهای ریشه نیز بر عهده دارد. این سازمان وظیفه‌ی دشوار و حساس اختصاص نام قلمروها و نیز اختلاف‌های حقوقی ایجاد شده بر اثر این نام‌ها را نیز بر عهده دارد. معمولاً ICANN اختصاص توده آدرس‌ها را به دفاتر اسناد رسمی محلی اینترنت مانند: ARIN ، RIPE و LACNIC واگذار می‌کند، و این سه به همراه یکدیگر سازمان پشتیبانی آدرس‌های اینترنت (ASO)[2] را تشکیل می‌دهند که وظیفه‌ی این سازمان اختصاص و مدیریت آدرس‌ها در درون قلمرو هر یک از آن‌ها می‌باشد.

- **اخذ یک آدرس IP میزبان**

هنگامی که یک سازمان توده آدرس IP مورد تقاضا را دریافت نمود آنگاه می‌تواند هر یک از این

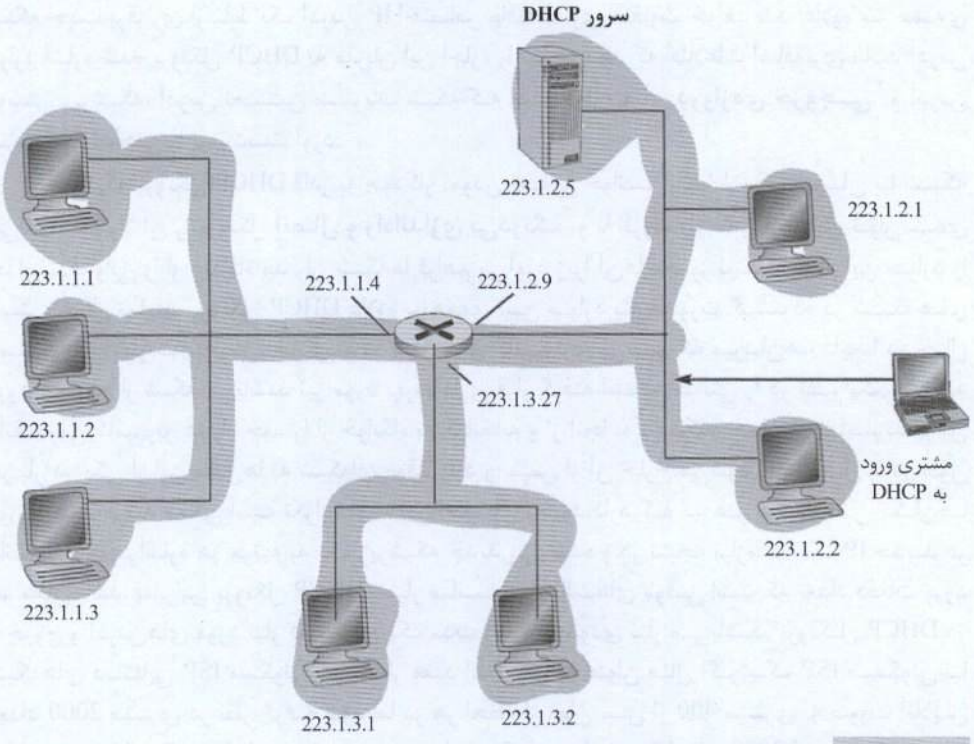
1- International corporation for assigned names and address
2- Address supporting organization

آدرس‌های IP را به میزبان و واسط مسیریاب‌ها در آن نهاد اختصاص دهد. یک مدیر سیستم با کمک گرفتن از یک ابزار مدیریت شبکه (نرم‌افزار) می‌تواند اختصاص این آدرس‌ها و پیکربندی آن‌ها را به مسیریاب‌ها اصطلاحاً به گونه‌ی دستی انجام دهد. علاوه بر آن اختصاص آدرس‌ها به میزبان‌ها نیز به گونه‌ی دستی قابل پیکربندی می‌باشند اما معمولاً این وظیفه به گونه‌ی خودکار بر عهده **پروتکل پیکربندی پویای میزبان**[1] **(DHCP)** گذارده می‌شود. ویژگی‌های این پروتکل در مستند [RFC 2131] تشریح شده است. پروتکل DHCP به یک میزبان این اجازه را می‌دهد که به صورت خودکار نسبت به اخذ یک آدرس IP اقدام نماید. یک مدیر شبکه DHCP را می‌تواند به‌گونه‌ای پیکربندی نماید که یک میزبان با هر مراجعه به شبکه جهت برقراری ارتباط یک آدرس IP مشابه را اخذ نماید و یا آنکه آدرس IP اختصاص یافته به میزبان موقتی باشد در این صورت با هر مراجعه به شبکه جهت برقراری ارتباط یک آدرس IP اختصاص یافته به وی متفاوت خواهد شد. علاوه بر همه‌ی موارد اشاره شده پروتکل DHCP به میزبان این اجازه را نیز می‌دهد که اطلاعات اضافه‌تری مانند آدرس پوشش زیرشبکه، آدرس نخستین مسیریاب شبکه که اصطلاحاً به آن **دروازه‌ی خروجی**[2] و آدرس سرور DNS محلی را نیز بدست آورد.

از آنجائیکه پروتکل DHCP قادر به خودکار نمودن تمامی جوانب ارتباط یک میزبان با شبکه می‌باشد معمولاً آن را پروتکل **اتصال و راه‌اندازی بی‌درنگ**[3] و یا افزونه نیز نامیده‌اند. این توان‌مندی جذابیت فراوانی را از دیدگاه مدیران شبکه‌ها فراهم می‌آورد زیرا آن‌ها مجبور نیستند تمام این موارد را دستی اعمال نمایند. پروتکل DHCP علاوه بر همه‌ی این موارد به صورت گسترده در شبکه‌های دستیابی به اینترنت شخصی، خانگی و در شبکه‌های کامپیوتری بی‌سیم که میزبان‌ها دایماً در حال ورود و خروج از شبکه می‌باشند نیز مورد بهره‌برداری قرار گرفته است. حالتی را در نظر بگیرید که دانشجویی کامپیوتر همراه خود را از خوابگاه به کتابخانه و از آنجا به دانشکده حمل می‌نماید بنابراین هر بار در یکی از این مکان‌ها به شبکه‌ای موقتاً وارد و سپس از آن خارج می‌شود و این عمل را در طول روز چندین مرتبه ممکن است تکرار نماید. این احتمال وجود دارد که در هر یک از این مکان‌ها دانشجوی مورد اشاره هر مرتبه به یک زیرشبکه جدید وارد شده و در نتیجه نیازمند یک IP جدید در هر مکان باشد. بنابراین پروتکل DHCP بسیار مناسب این حالت‌های موقتی است که تعداد دفعات ورود و خروج و آدرس‌های مورد نیاز فقط برای یک مدت زمان محدودی نیاز می‌باشد. پروتکل DHCP در شبکه‌های دستیابی ISP مسکونی نیز بسیار مفید است زیرا به عنوان مثال اگر یک ISP مسکونی با تعداد 2000 مشتری در نظر گرفته شود اما در هر لحظه از زمان بیش از 400 مشتری به صورت آنلاین شبکه را دستیابی کنند در این حالت به جای نیاز به یک توده آدرس IP برابر 2048 آدرس یک سرور DHCP کافی است که اختصاص آدرس‌های پویای مورد نیاز را در یک توده‌ی 512 آدرس IP انجام دهد به عنوان مثال یک توده‌ی آدرس IP در قالب: a.b.c.d\23 هر مرتبه که میزبانی به سرور DHCP وارد و یا از آن خارج می‌شود نیازمند به روزرسانی صورت آدرس‌های IP قابل دسترس می‌باشد. بنابراین با هر مرتبه ورود یک میزبان سرور DHCP یک آدرس تصادفی IP را از میان مجموعه آدرس‌های جاری در دسترس خود به آن میزبان اختصاص می‌دهد و هر مرتبه میزبان شبکه را ترک می‌کند آدرس IP وی به آن مجموعه عودت داده می‌شود.

1- Dynamic host configuration protocol
2- Gateways
3- Plug and play protocol

پروتکل DHCP به گونه‌ی مشتری – سرور عمل می‌نماید. یک مشتری یک میزبان وارد شده ی جدید است که نیازمند دستیابی به اطلاعات پیکربندی شبکه از جمله یک آدرس IP برای خویش می‌باشد. در ساده‌ترین حالت هر زیرشبکه در قالب آدرس‌دهی شکل ۴-۱۷ دارای یک سرور DHCP است. اگر سروری در زیرشبکه حضور ندارد آنگاه نیاز به یک عامل تکرار DHCP مانند یک مسیریاب می‌باشد که از آدرس یک سرور DHCP برای آن شبکه آگاه است. شکل ۴-۲۰ نشان می‌دهد یک سرور DHCP متصل به زیرشبکه‌ای است مانند 223.1.2/24 با مسیریابی که به عنوان عامل تکرار برای ورود مشتری‌های متصل به زیرشبکه‌های 223.1.1/24 و 223.1.3/24 است برای یک میزبان جدید وارد شده، پروتکل DHCP یک فرآیند چهار مرحله‌ای است.



شکل ۴-۲۰ ♦ چگونگی عملکرد مشتری – سرور **DHCP**

شکل ۴-۲۱ این فرآیند چهار مرحله‌ای را جهت پیکربندی شبکه‌ی ۴-۲۰ نشان می‌دهد. در این شکل yiaddr[1] به مفهوم آدرس اختصاص یافته به مشتری جدید وارد شده، می‌باشد. این چهار مرحله عبارتند از:

۱- **بازیابی سرور DHCP.** نخستین وظیفه‌ی میزبان جدیداً وارد شده به شبکه آن است که یک سرور DHCP را جهت تعامل شناسایی نماید. این وظیفه از طریق بهره‌گیری از یک **پیام کشفِ DHCP** که یک مشتری آن را از طریق یک بسته‌ی UDP بر روی درگاه 67 ارسال می‌نماید، انجام می‌شود. بسته‌ی UDP در قالب یک داده‌گرام IP بسته‌بندی می‌شود. اما پرسش آن است

1- Your internet address

که این داده‌گرام را به چه کسی گسیل نماید؟ میزبان حتی آدرس IP شبکه‌ای که به آن متصل است را نیز نمی‌داند تا چه رسد به آدرس یک سرور DHCP برای آن شبکه. با توجه به این واقعیت، مشتری DHCP داده‌گرامی را خلق می‌کند که شامل پیام کشف DHCP در مجاورت آدرس فراپخش IP در قالب 255.255.255.255 و یک آدرس IP منبع: "این میزبان" به صورت 0.0.0.0 : است. مشتری DHCP داده‌گرام IP را به لایه‌ی پیوند انتقال می‌دهد و سپس این قاب را به همه‌ی گره‌های متصل به زیرشبکه فراپخش می‌کند. در فصل ۵ روش فراپخش بررسی شده است.

۲- **پاسخگویی سرور DHCP.** یک سرور DHCP که پیام کشف یک مشتری را دریافت می‌دارد. واکنش خود را در قالب یک **پیام پیشنهاد DHCP** به همه‌ی کارت‌های شبکه در زیرشبکه فراپخش می‌نماید و این عمل را با بهره‌گیری از آدرس فراپخش IP یعنی 255.255.255.255 مجدداً انجام می‌دهد. شاید علت پاسخ سرور به گونه‌ی فراپخش نیز جالب به نظر آید. از آنجائیکه چندین سرور DHCP می‌تواند در یک زیرشبکه حاضر باشند، برای مشتری ممکن است این حس پدیدار شود که در یک موقعیت دشواری جهت گزینش از میان چندین پیشنهاد ارایه شده قرار گرفته است. هر پیام پیشنهاد از یک سرور شامل شناسه‌ی (ID) تراکنش پیام کشف دریافتی است که شامل آدرس IP پیشنهادی برای مشتری، آدرس پوشش شبکه و **یک آدرس زمان اعتبار IP**[1] می‌باشد. آدرس زمان اعتبار IP مدت زمانی است که آن IP دارای اعتبار است. معمولاً این زمان از سوی سرور شامل چندین ساعت و یا چندین روز می‌شود.

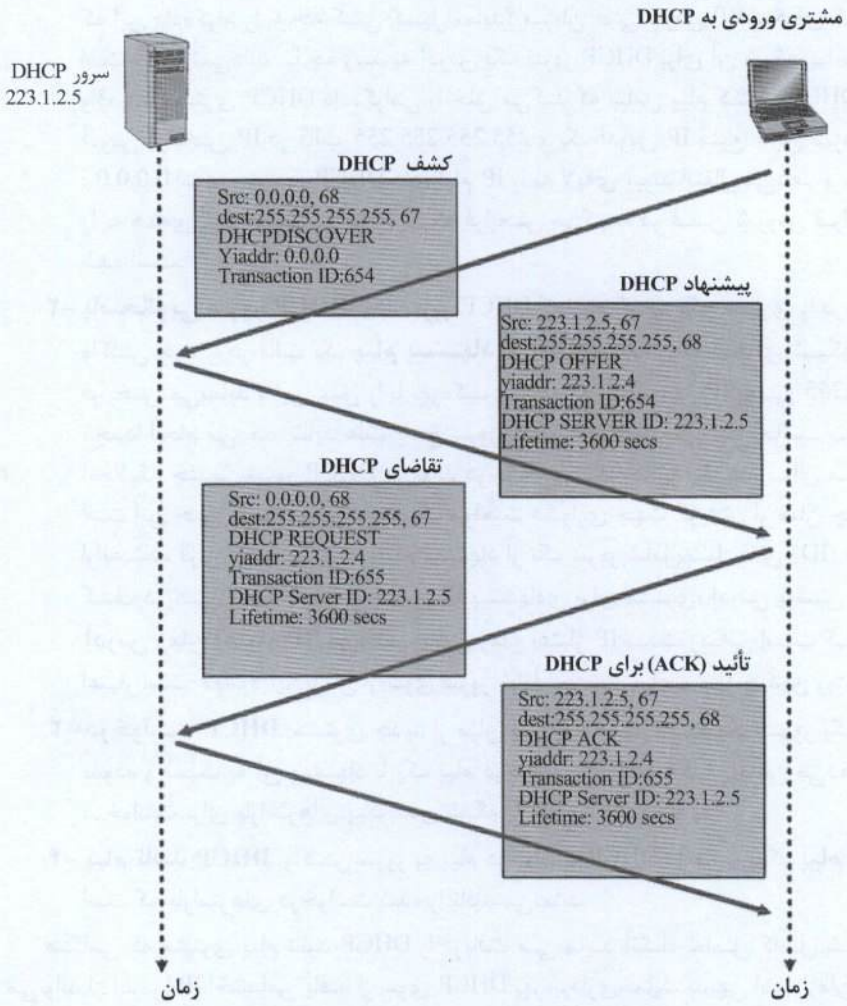
۳- **درخواست DHCP.** مشتری جدید از میان یک یا چندین پیشنهاد سرور یکی را گزینش نموده و نسبت به آن پیشنهاد با یک پیام درخواست DHCP واکنش نشان می‌دهد. با این عمل درخواست برای پارامترهای پیکربندی شبکه را بازگشت می‌دهد.

۴- **پیام تائید DHCP.** واکنش سرور به پیام درخواست DHCP در قالب یک **پیام تائید DHCP** است که پارامترهای درخواست شده را تائید می‌نماید.

هنگامی که مشتری پیام تائید DHCP را دریافت می‌نماید آنگاه تعامل کامل شده و مشتری می‌تواند از آدرس IP اختصاص یافته از سوی DHCP بهره‌برداری نماید. بدیهی است زمان بهره‌برداری برابر زمان اعتبار IP است که قبلاً به آن اشاره شد. از آنجائیکه یک مشتری ممکن است نیازمند استفاده از آدرس فراتر از زمان اعتبار آن باشد آنگاه DHCP عوامل لازم جهت تامین اعتبار مجدد این زمان را برای یک ادرس IP فراهم می‌آورد.

ارزش توان‌مندی بهره‌برداری بی‌درنگ DHCP با توجه به این واقعیت که جایگزین آن پیکربندی دستی آدرس IP میزبان است بسیار آشکار است. در اینجا بازهم دانشجویی را که بین دانشکده، کتابخانه و خوابگاه به همراه کامپیوتر همراه خود در حرکت می‌باشد تصور نمائید. پیوستن به یک زیرشبکه‌ی جدید مستلزم آدرس IP جدید در هر یک از این مکان‌ها می‌باشد. دشواری انجام عملیات برای یک مدیر شبکه به گونه‌ی دستی در این وضعیت که کامپیوتر همراه را هر مرتبه در مکان جدید پیکربندی نماید آشکار است. از نقطه‌نظر تحرک‌پذیری پروتکل DHCP حامل نقاط ضعفی نیز می‌باشد.

1- IP address leave time



شکل ۴-۲۱ ♦ تعامل مشتری – سرور DHCP

. از آنجائیکه اخذ یک آدرس IP جدید از DHCP در هر زمان که کارت شبکه به یک زیرشبکه‌ی جدید متصل می‌شود، انجام می‌پذیرد، پیوستگی یک ارتباط TCP یه یک کاربرد هنگامیکه میزبان بین زیرشبکه‌ها در حرکت است امکان‌پذیر نمی‌باشد. در فصل ۶ موارد مربوط به IP سیار که اجازه‌ی تحرک آزادانه را به یک میزبان با استفاده از یک آدرس دایمی IP بین زیرشبکه‌های گوناگون می‌دهد بررسی خواهد شد.

- **ترجمه‌ی آدرس شبکه (NAT)**

با توجه به بحث آدرس‌دهی اینترنت و قالب‌بندی داده‌گرام‌های IPV4 آشکار است که هر وسیله‌ی IP نیازمند یک آدرس IP نیز می‌باشد. با گسترش زیرشبکه‌های متعلق به دفاتر کار کوچک و خانه و دفتر

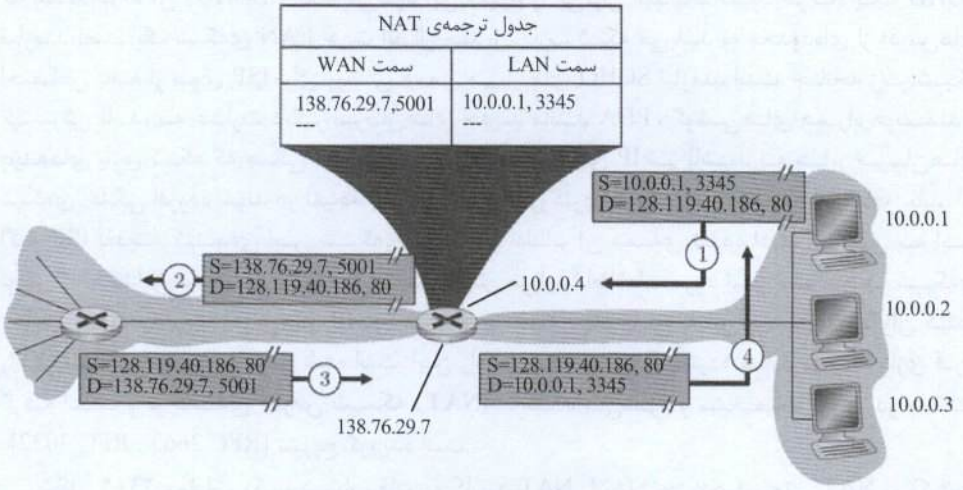
که اصطلاحاً به آن (SOHO)[1] گفته می‌شود این مفهوم را می‌توان استنباط نمود هرگاه یک SOHO نیازمند نصب یک شبکه‌ی LAN جهت اتصال چندین کارت شبکه می‌باشد به محدوده‌ای از آدرس‌های اختصاص یافته از سوی ISP برای پوشش همه‌ی میزبان‌های SOHO نیازمند است. چنانچه زیر شبکه گسترش یابد، به عبارت دیگر میزبان‌های جدید مانند PDA ، گوشی‌های همراه هوشمند و پایانه‌های بازی شبکه که همگی به نوعی نیازمند آدرس‌های IP می‌باشند به سایر میزبان‌های شبکه‌ی خانگی افزوده شوند در نتیجه یک توده آدرس بزرگتری بایستی به این شبکه اختصاص یابد. اما اگر ISP تغذیه کننده‌ی این شبکه‌ی SOHO و نظایر آن همه‌ی توده آدرس متعلق به این بخش را اختصاص داده باشد آنگاه چه وضعیتی پیش می‌آید؟ اطلاعات مورد نیاز مالک یک شبکه‌ی خانگی جهت مدیریت آدرس‌های IP در اختیار او چیست؟ جهت پاسخ به این پرسش‌های نگران کننده راه‌کار نسبتاً ساده‌ای پیش‌بینی شده است. این راه‌کار به صورت گسترده مورد بهره‌برداری قرار گرفته است و **ترجمه‌ی آدرس شبکه (NAT)**[2] نامیده می‌شود و مشخصات آن در دو مستند [RFC 2663 , RFC 3022] تشریح گردیده است.

شکل ۴-۲۲ عملیات یک مسیریاب قادر به کاربرد NAT را نشان می‌دهد. روتر NAT ساکن در شبکه‌ی خانگی از یک واسط به عنوان بخشی از آن شبکه مانند آنچه در سمت راست شکل ۴-۲۲ نشان داده شده است بهره می‌گیرد. آدرس‌دهی در بین شبکه‌ی خانگی مشابه موارد اشاره شده در بالا می‌باشد. همگی چهار واسط شبکه در شکل ۴-۲۲ دارای آدرس‌های زیرشبکه‌ی مشابه 10.0.0.0/24 می‌باشند. فضای آدرس 10.0.0.0/8 یکی از سه بخش فضای آدرس IP ذخیره شده در مستند [RFC 1918] جهت یک شبکه خصوصی و یا یک **قلمرو** با آدرس‌های خصوصی است. یک قلمرو با آدرس‌های خصوصی به یک شبکه اشاره می‌کند که آدرس‌های آن فقط برای تجهیزات درون آن شبکه مفهوم و معتبر است. جهت ملاحظه‌ی اهمیت این موضوع این واقعیت را در نظر بگیرید که صدها هزار شبکه‌ی خانگی وجود دارند که همگی آن‌ها ممکن است از فضای آدرس مشابه 10.0.0.0/24 بهره گیرند. تجهیزات درون یک شبکه‌ی خانگی می‌تواند گسیل بسته‌ها را به یکدیگر با بهره‌گیری از آدرس 10.0.0.0/24 انجام دهد. بسته‌های گسیل شده به خارج از یک شبکه‌ی خانگی مانند اینترنت جهانی آشکارا نمی‌تواند از این آدرس‌ها استفاده نماید چه در قالب آدرس‌های مبدا و یا مقصد زیرا که صدها هزار شبکه وجود دارند که از همین توده آدرس مشابه بهره می‌گیرند. به عبارت دیگر آدرس‌های 10.0.0.0/24 فقط می‌توانند در بین یک شبکه‌ی خانگی مفهوم داشته باشند. اما اگر آدرس‌های خصوصی از طرفی فقط در بین یک شبکه معنی‌دار باشند چگونه پیاده سازی آدرس‌دهی به هنگام ارسال و دریافت بسته‌ها از شبکه‌ی اینترنت جهانی انجام‌پذیر است؟ در کدام بخش، این آدرس‌ها بایستی لزوماً انحصاری باشند؟ پاسخ درک کافی از NAT است.

مسیریاب بهره‌مند از NAT شبیه به یک روتر متعارف برای دنیای خارج از آن نمی‌باشد بلکه مسیریاب NAT به گونه ی یک میزبان با یک آدرس IP رفتار می‌کند. در شکل ۴-۲۲ تمامی ترافیک خارج شده از مسیریاب خانگی به سوی شبکه‌ی اینترنت بزرگتر دارای یک آدرس IP مبدا 138.76.29.7 است و همه‌ی ترافیک وارد شده به روتر خانگی بایستی از یک آدرس مقصد 138.76.29.7 بهره گیرد.

1- Small office home office

2- Network address translation



شکل ۴-۲۲ ♦ترجمه‌ی آدرس شبکه

به عبارت دیگر مسیریاب بهره‌مند از NAT جزئیات مربوط به شبکه‌ی خانگی را از دنیای خارج از آن پنهان نگاه می‌دارد. اکنون این پرسش مطرح می‌شود که کامپیوترهای شبکه‌ی خانگی آدرس‌های خود را از کجا اخذ می‌کنند و روتر این شبکه نیز آدرس IP تکِ خود را از چه مکانی دریافت می‌دارد؟ در اغلب مواقع پاسخ DHCP است! مسیریاب آدرس خود را از سرور DHCP متعلق به ISP می‌گیرد و مسیریابی که پروتکل سرور DHCP را اجرا می‌نماید آدرس‌های IP مورد نیاز کامپیوترهای وابسته به روتر NAT – DHCP کنترل کننده‌ی فضای آدرس شبکه‌ی خانگی را تامین می‌کند.

اگر همه‌ی داده‌گرام‌های وارد شده در مسیریاب NAT از شبکه‌ی WAN با آدرس IP مقصد مشابه باشند به ویژه از واسط سمت WAN که مستقیماً وابسته به مسیریاب NAT است آنگاه چگونه روتر از میزبان داخلی که بایستی یک داده‌گرام را گسیل نماید آگاه می‌شود؟ پاسخ بهره‌گیری از یک **جدول ترجمه‌ی NAT** است. این جدول در مسیریاب NAT ساکن است و شامل شمارگان درگاه‌ها و آدرس‌های IP است.

مثال شکل ۴-۲۲ را مجدداً در نظر بگیرید. اگر فرض شود کاربری در شبکه‌ی خانگی از میزبان با آدرس IP برابر 10.0.0.1 تقاضای یک صفحه‌ی وب بر روی یک وب سرور درگاه 80 را نماید که آدرس IP آن نیز 128.119.40.186 باشد. میزبان 10.0.0.1 شماره‌ی درگاه مبدا دلخواه 3345 را اختیار و داده‌گرام را دریافت و یک شماره درگاه مبدا جدید مانند 5001 را برای آن داده‌گرام تولید می‌کند سپس آدرس IP مبدا را با آدرس سمت WAN خود یعنی 138.76.29.7 جایگزین می‌کند و شماره درگاه 3345 را نیز با شماره پورت 5001 جایگزین می‌کند. هنگام تولید یک شماره درگاه مبدا جدید، مسیریاب NAT می‌تواند هر شماره درگاه مبدا که در حال حاضر در جدول ترجمه‌ی NAT موجود نمی‌باشد را گزینش نماید. لازم به یادآوری است چون میدان درگاه طولی معادل 16 بیت دارد پروتکل NAT می‌تواند بیش از 60000 ارتباط همزمان را با یک آدرس IP طرف WAN برای آن مسیریاب NAT پشتیبانی نماید! علاوه بر آن NAT در مسیریاب یک ورودی را نیز به جدول ترجمه‌ی NAT اضافه می‌کند. وب سرور شبکه بدون آگاهی از این که داده‌گرام دریافتی حامل تقاضای HTTP قبلاً از سوی مسیریاب NAT دست‌کاری شده است واکنش خود را با یک داده‌گرام که آدرس مقصد آن IP

آدرس مسیریاب NAT و درگاه مقصد آن نیز 5001 است، کامل می‌کند. هنگامی که این داده‌گرام وارد مسیریاب NAT می‌شود روتر از طریق شاخص‌گذاری جدول ترجمه‌ی NAT از آدرس IP مقصد و شماره درگاه مقصد بهره‌گیری می‌نماید تا آدرس IP مناسب 10.0.0.1 و شماره درگاه مقصد 3345 را برای مرورگر شبکه‌ی خانگی بدست آورد. مسیریاب آنگاه آدرس مقصد داده گرام و شماره درگاه آن را مجدداً بازنویسی و داده‌گرام را به سوی شبکه‌ی خانگی گسیل می‌کند.

روش ترجمه‌ی آدرس NAT در سال‌های اخیر به گونه‌ی گسترده مورد بهره‌برداری واقع شده است. اما در هر حال هر راه‌کاری منتقدان خود را نیز دارد. در جامعه‌ی IETF افرادی که مخالف NAT می‌باشند، این گونه استدلال می‌کنند که:

- شماره درگاه‌ها به منظور بهره‌گیری فرآیندهای آدرس‌دهی طراحی شده‌اند و به‌منظور آدرس‌دهی میزبان‌ها می‌باشند بنابراین این کاربرد متناقض سبب دشواری‌هایی برای سرورهای مجری شبکه‌های خانگی می‌شود زیرا در فصل دوم ملاحظه شد فرآیندهای سرور به انتظار درخواست ورودی بر روی شماره درگاه‌های شناخته شده می‌ماند.
- مسیریاب‌ها جهت پردازش بسته‌ها حداکثر تا لایه‌ی سوم طراحی شده‌اند در حالیکه در این فرآیند این مورد تا حدودی رعایت نمی‌شود.
- پروتکل NAT در تناقض با موضوع اصطلاحاً انتها به انتها می‌باشد که میزبان‌ها بایستی مستقیماً و بدون واسطه با یکدیگر تعامل برقرار می‌کردند و مداخله‌ای را در کار گره‌هایی که آدرس‌های IP و شماره درگاه‌ها را تغییر می‌دهند، انجام نمی‌دادند.
- بایستی از آدرس نسخه 6 یعنی IPV6 جهت جبران کمبود آدرس‌های IP استفاده شود تا آنکه عجولانه راه‌حل این کمبود از طریق روش‌های میان بُر مانند NAT انجام شود.

شاید یکی از چالش‌های مهم دیگر NAT دشواری ارتباط با کاربردهای P2P باشد. این کاربردها از جمله شامل اشتراک فایل P2P ، مکالمه و صدای P2P از طریق کاربردهای IP می‌شوند. در فصل دوم ملاحظه شد هر کاربرد P2P از طریق یک نظیر بایستی قادر به آغاز یک ارتباط TCP با همانند دیگر خود باشد. مانند دو نظیر A و B. دشواری اصلی در آن است که اگر نظیر B در پس پرده‌ی NAT پنهان باشد آنگاه نمی‌تواند به عنوان یک سرور ارتباط‌های TCP را پذیرش کند. البته یادآوری این نکته لازم است که اگر نظیر A در پس پرده ی NAT نباشد این دشواری را می‌توان برطرف نمود. در این حالت نظیر A می‌تواند نخست از طریق یک مانند بی‌درنگ مثل C با نظیر B ارتباط برقرار نماید. بدیهی است مانند C در پس NAT قرار ندارد و B نیز با آن دارای یک ارتباط TCP جاری است. نظیر A می‌تواند با پرسش از B از طریق نظیر C به انجام یک ارتباط TCP بپردازد که مستقیماً به A بازگشت می‌کند. هنگامی که ارتباط مستقیم TCP به گونه P2P بین دو نظیر A و B برقرار شد آنگاه دو مانند می‌توانند به تبادل پیام و یا فایل اقدام کنند. این حیله دراصطلاح شبکه **ارتباط معکوس**[1] نام گرفته است و معمولاً از طریق بسیاری از کاربردهای P2P جهت پیمایش NAT استفاده می‌شود. اگر هر دو نظیر A و B در پس NAT خویش قرار گیرند آنگاه انجام عملیات ارتباط اندکی دشوارتر می‌شود.

- **پروتکل UPnp**

پیمایش NAT به شکل فزاینده‌ای از طریق **اتصال و کاربرد فراگیر بی‌درنگ (UPnp)**[2] و یا به سخن

1- Connection reversal
2- Universal plug and play

دیگر **افزونه‌ی بی‌درنگ** انجام می‌گیرد که یک پروتکل با امکانات کشف و پیکربندی یک NAT نزدیک برای هر میزبان است. در پروتکل UPnp الزاماً هر دو میزبان و NAT بایستی سازگار با UPnp باشند. یک کاربرد در حال اجرا در یک میزبان با بهره‌گیری از پروتکل UPnp می‌تواند تقاضای یک نگاشت NAT را بین آدرس IP و شماره درگاه خصوصی خود و آدرس IP و شماره درگاه عمومی در شبکه‌ی اینترنت نماید. اگر پروتکل NAT تقاضای صادر شده از سوی یک میزبان را پذیرش نماید. در صورت انجام عملیات نگاشت آنگاه میزبان‌های خارج از شبکه می‌توانند آغازگر ارتباط‌های TCP به آدرس‌های IP و شماره درگاه‌های عمومی باشند. علاوه بر این پروتکل این اجازه را به کاربردها می‌دهد که از ارزش آدرس‌های IP و نیز شناسه‌ی درگاه‌های عمومی آگاه شده و در نتیجه کاربرد قادر به آگهی به خارج از دنیای خویش شود.

به عنوان مثال فرض کنید میزبانی در پس پرده پروتکل PUnp مسیریاب با بهره‌مندی از NAT پنهان شده و دارای آدرس خصوصی 10.0.0.1 است در عین حال این میزبان در حال اجرای بیت تُرنت بر روی درگاه 3345 است. همچنین آدرس IP عمومی NAT نیز 138.76.29.7 می‌باشد. کاربرد بیت تُرنت مایل به پذیرش ارتباط از دیگر میزبان‌ها می‌باشد بنابراین می‌تواند قطعه‌های اطلاعات خود را با آن‌ها مبادله نماید. در این راستا کاربرد بیت تُرنت در میزبان اشاره شده از NAT تقاضا می‌کند که یک حفره که یک آدرس (3345 , 10.0.0.1) را به آدرس (5001 , 138.76.29.7) نگاشت کند، ایجاد نماید. درگاه عمومی 5001 از سوی کاربرد گزینش شده است. کاربرد بیت ترنت در میزبان می‌تواند این آگهی را به برنامه‌ی رهگیرش نیز گسیل نماید تا آن را از در دسترس بودن خود در شناسه‌ی (5001 , 138.76.29.7) آگاه نماید. در این حالت یک میزبان خارجی در حال اجرای بیت ترنت می‌تواند با برنامه‌ی رهگیر تماس گرفته و از اطلاعات شناسه‌ی اشاره شده یعنی (5001 , 138.76.29.7) آگاه گردد. پس از آن میزبان خارجی می‌تواند یک بسته‌ی TCPSYN را به این آدرس گسیل نماید. هنگامی که NAT بسته‌ی SYN را دریافت می‌دارد آدرس IP مقصد و شماره درگاه را در بسته‌ی اشاره شده به اطلاعات جدید یعنی (3345 , 10.0.0.1) تغییر می‌دهد و بسته را از طریق NAT گسیل به جلو می‌نماید.

نتیجه آنکه UPnp اجازه می‌دهد میزبان‌های خارجی آغازگر ارتباط‌ها به میزبان‌های بهره‌مند از NAT شوند و این عمل را از طریق استفاده از UDP و یا TCP انجام دهند. پروتکل NAT برای مدت‌های طولانی است که یک عامل اجتناب‌ناپذیر جهت کاربردهای P2P به شمار می‌آید. با پروتکل UPnp یک پیمایش NAT قدرتمند و مؤثر می‌تواند در نقش منجی آن‌ها ظاهر گردد.

۴-۱۰ پروتکل کنترل پیام اینترنت (ICMP)

با یادآوری این نکته که لایه‌ی شبکه در اینترنت حامل سه عنصر اصلی است که عبارتند از: (۱) پروتکل IP که در بخش پیشین بررسی شد ، (۲) پروتکل‌های مسیریابی اینترنت شامل RIP ، OSPF و BGP که در بخش‌های بعدی همین فصل بررسی خواهند شد و (۳) پروتکل کنترل پیام اینترنت[1] (ICMP) که موضوع این بخش است.

مشخصات پروتکل ICMP در مستند [RFC 792] تشریح شده است. این پروتکل به وسیله‌ی میزبان‌ها و مسیریاب‌ها جهت مبادله‌ی اطلاعات لایه‌ی شبکه به یکدیگر مورد استفاده قرار می‌گیرد.

1- Internet control messege protocol

متداولترین کاربرد ICMP جهت گزارش خطای حاصل در لایه شبکه می‌باشد. به عنوان مثال هنگامی که یک پردازش تِلنت، FTP و یا HTTP در حال اجرا است، ممکن است یک پیام خطا مانند « مقصد شبکه غیرقابل دسترس است» دریافت گردد. این پیام از پروتکل ICMP سرچشمه می‌گیرد. در نقطه‌ای از ارتباط یک مسیریاب IP قادر به یافتن مسیر به میزبان مشخص شده در یکی از کاربردهای اشاره شده (تلنت، FTP و HTTP) نمی‌باشد. در این حالت است که مسیریاب یک پیام ICMP از نوع 3 را تعریف و سپس گسیل کرده است تا میزبان مقصد از خطای ایجاد شده آگاه شود.

پروتکل ICMP معمولاً بخشی از IP به شمار می‌آید اگرچه از لحاظ معماری لایه‌بندی شبکه این پروتکل اندکی بالاتر از IP می‌باشد چرا که پیام‌های ICMP در درون داده‌گرام‌های IP حمل می‌شوند. به عبارت دیگر پیام‌های ICMP به عنوان بخشی از محموله‌ی IP به شمار می‌آیند، درست مانند قطعه‌های TCP و یا UDP که بخشی از محموله IP قلمداد می‌شوند. هنگامی که یک میزبان یک داده‌گرام IP با پیام ICMP مشخص شده‌ای را از پروتکل لایه‌ی بالاتر دریافت می‌کند به بازگشایی محتوای داده‌گرام به ICMP می‌پردازد درست مانند هنگامی که بازگشایی محتوای داده‌گرام‌ها را به TCP یا UDP انجام می‌دهد.

پیام‌های ICMP همگی دارای میدان‌های نوع، کُد و سرپیام می‌باشند. نخستین 8 بایت داده‌گرام IP که سبب پیام ICMP می‌شود نیز بخشی از این اطلاعات است. این 8 بایت از سوی فرستنده جهت شناسایی داده‌گرامی است که سبب خطا شده است. انواع پیام‌های ICMP در شکل ۴-۲۳ نشان داده شده‌اند. بدیهی است بر اساس شکل، پیام‌های ICMP تنها جهت سیگنالینگ وضعیت خطا نمی‌باشند.

برنامه‌ی معروف Ping به ارسال یک پیام ICMP از نوع 8 با کُد صفر را به میزبان مشخص شده گسیل می‌دارد. میزبان مقصد با مشاهده‌ی تقاضای اِکو واکنش خود را در قالب یک پیام نوع صفر و با کد صفر ICMP به فرستنده گسیل می‌کند. اغلب پیاده سازی‌های TCP/IP به پشتیبانی مستقیم سرور Ping در سیستم عامل می‌پردازند. به عبارت دیگر سرور یک پردازش نمی‌باشد بلکه یک برنامه‌ی مشتری است. بدیهی است که برنامه‌ی مشتری بایستی قادر باشد سیستم‌عامل را مجبور به تولید یک پیام ICMP از نوع 8 یا کُد صفر نماید.

یکی دیگر از پیام‌های جالب ICMP پیام فرونشانی مبداء است. این پیام در عمل به ندرت مورد استفاده قرار می‌گیرد. هدف اصلی این پیام کنترل تراکم در شبکه است که مسیریابِ دچار تراکم را قادر به ارسال یک پیام ICMP جهت فرونشاندن مبدا تراکم می‌کند. ارسال این پیام به یک میزبان تولید کننده‌ی تراکم سبب می‌شود که آن میزبان مجبور به کاهش سرعت انتقال خود به شبکه شود. در فصل ۳ ملاحظه شد که پروتکل TCP روش کنترل تراکم خویش را دارد که در لایه‌ی انتقال عمل می‌کند بدون آنکه از بازخورد لایه‌ی شبکه یعنی پیام فرونشانی مبدأ بهره گیرد.

در فصل اول برنامه‌ی رهگیری مسیر معرفی شد که اجازه‌ی رهگیری یک مسیر را از یک میزبان به هر میزبان دیگر شبکه بدون در نظر گرفتن فاصله فراهم می‌آورد. نکته جالب در این است که برنامه‌ی رهگیری مسیر از طریق پیام‌های ICMP قابلیت پیاده‌سازی را دارد. به‌منظور تشخیص اسامی و آدرس‌های مسیریاب‌های بین مبدا و مقصد برنامه رهگیری مسیر در مبدا مجموعه‌ای از داده‌گرام‌های معمولی را به مقصد گسیل می‌کند. هر یک از این داده‌گرام‌ها یک قطعه‌ی UDP را حمل می‌کند که همراه آن یک شماره درگاه غیر متعارف است. نخستین داده‌گرام‌های از این نوع دارای زمان TTL برابر یک می‌باشند. این زمان برای داده‌گرام‌های 2 و 3 به ترتیب برابر 2و 3 و غیره است. بدیهی است مبدا نیز تایمرهایی را برای هر یک از این داده‌گرام‌ها فعال می‌سازد. هنگامی که داده‌گرام n وارد مسیریاب n

می‌شود. این مسیریاب متوجه می‌شود که TTL داده‌گرام به پایان رسیده است. بنابر قواعد پروتکل IP مسیریاب از داده‌گرام صرف‌نظر می‌نماید و پیام ICMP هشدار را به مبدأ که از نوع 11 و کُد صفر است گسیل می‌کند. این پیام هشدار شامل نام و آدرس IP مسیریاب است. هنگامی که این پیام ICMP مجدداً وارد مبدا می‌شود، مبدا زمان TTL را از تایمر فعال شده و نامِ و آدرسِ IP مسیریاب n را از پیام ICMP اخذ می‌کند.

نوع	کد	توضیحات
0	0	پیام انعکاس (اِکو) پاسخ به Ping است
3	0	شبکه‌ی مقصد غیرقابل دسترس است
3	1	میزبان مقصد غیرقابل دسترس است
3	2	پروتکل مقصد غیرقابل دسترس است
3	3	درگاه مقصد غیرقابل دسترس است
3	6	شبکه مقصد ناشناس است
3	7	میزبان مقصد ناشناس است
4	0	فرونشانی مبدا (کنترل تراکم)
8	0	تقاضای پیام انعکاس
9	0	آگهی مسیریاب
10	0	کشف یا شناسایی مسیریاب
11	0	خاتمه‌ی TTL
12	0	سرپیام نامطلوب IP

شکل ۴-۲۳ ♦ انواع پیام ICMP

پرسش مهم دیگر آن است که چگونه برنامه‌ی رهگیری مسیر در مبدا از هنگام توقف ارسال قطعه‌های UDP آگاه می‌شود؟ یادآوری می‌شود که با هر مرتبه ارسال یک داده‌گرام دامنه‌ی TTL از سوی مبدا افزایش می‌یابد بنابراین یکی از داده‌گرام‌ها سرانجام تمام مسیر را تا رسیدن به میزبان مقصد پیمایش می‌کند. از آنجایی که این داده‌گرام شامل یک قطعه‌ی UDP با شماره درگاه غیر متعارف است، میزبان مقصد یک پیام ICMP از نوع 3 با کُد 3 را که به مفهوم غیرقابل دسترس بودن درگاه است گسیل می‌نماید. بدیهی است این پیام به مبدا بازگشت داده می‌شود. هنگامی که میزبان مبدا یک پیام ICMP ویژه‌ای را دریافت نماید، می‌داند که نیازی به ارسال بسته‌های کاوشگر اضافی ندارد. برنامه‌ی رهگیری مسیر معمولاً مجموعه‌ای از سه بسته با TTL مشابه را گسیل می‌کند، بنابراین خروجی برنامه‌ی رهگیرسه نتیجه را برای هر یک از TTL ها فراهم می‌آورد.

در این روش میزبان مبدا به گونه‌ای شماره درگاه را یادگیری می‌کند و به شناسایی مسیریاب‌هایی که بین او و میزبان مقصد قرار گرفته‌اند و نیز محاسبه‌ی مقدار TTL بین دو میزبان می‌پردازد. بدیهی است برنامه‌ی رهگیری مشتری بایستی قادر به مجبور نمودن سیستم‌عامل جهت تولید داده‌گرام‌های UDP با مقادیر TTL مشخص باشد و نیز بتواند از پیام‌های ICMP وارد شده از طریق سیستم‌عامل آگاهی یابد. شاید در این مقطع جالب باشد که روش‌های بازرسی داده‌گرام‌ها از طریق دیواره‌های آتش (فایروال) و آشکارسازی مهاجم بررسی شوند. اگر فرض شود به عنوان یک چالش هدف، مدیریت یک

شبکه‌ی خانگی، شرکت و یا دانشگاهی باشد. مهاجمانی که از محدوده آدرس‌های IP شبکه آگاه باشند به سادگی می‌توانند داده‌گرام‌های IP را به آدرس‌های در محدوده‌ی شبکه ارسال نمایند. این داده‌گرام‌ها همه گونه عملیات و تاثیرهای مخرب را می‌توانند به شبکه اعمال نمایند از جمله‌ی این موارد نگاشت شبکه با پیام‌های مکرر Ping و پویش درگاه‌ها، اخذ و دسترسی به اطلاعات قابل تهدید میزبان‌ها از طریق داده‌گرام‌های غیر متعارف و آلوده می‌باشد. علاوه بر این موارد سیلابی نمودن سرورها با سیلابی از پیام‌های ICMP و آلوده نمودن میزبان‌ها به وسیله‌ی بدافزارهای درون بسته‌ها را نیز می‌توان نام برد. با این عوامل تهدید گسترده یک مدیر شبکه چه تدبیری را بایستی در مورد این تخریب‌گرها که هر یک از آن‌ها قادر به ارسال بسته‌های آلوده به شبکه می‌باشند اتخاذ نماید؟ دو روش مؤثر دفاعی در برابر این تهدیدها دیواره‌های آتش و **سیستم‌های آشکارساز مهاجم (IDS)**[1] می‌باشد.

یک مدیر شبکه به عنوان نخستین راه‌کار در نصب یک دیواره‌ی آتش بین شبکه و اینترنت تلاش می‌کند. بدیهی است اکثریت مسیریاب‌های کنونی از توان‌مندی دیواره‌ی آتش برخوردار می‌باشند. دیواره‌های آتش نقش بازرسی سرپیام داده‌گرام‌ها و قطعه‌ها را بر عهده دارند و بنابراین مانع از ورود داده‌گرام‌های مشکوک به شبکه می‌شوند. به عنوان مثال یک دیواره‌ی آتش را می‌توان جهت مسدود نمودن همه ی بسته‌های درخواست انعکاس ICMP پیکربندی نمود. در این رویکرد یک مهاجم از حمله‌های از نوع پیام‌های مکرر Ping در محدوده‌ی آدرس‌های IP محروم می‌شود. دیواره‌های آتش می‌توانند بسته‌ها را بر مبنای آدرس‌های IP و شماره‌های درگاه مبدا و مقصد مسدود نمایند. علاوه بر موارد اشاره شده دیواره‌های آتش را می‌توان جهت رهگیری ارتباط‌های TCP نیز پیکربندی نمود در نتیجه فقط ورود داده‌گرام‌های متعلق به یک ارتباط مجاز امکان‌پذیر می‌شود.

حفاظت بیشتر از طریق یک IDS نیز امکان‌پذیر می‌باشد. یک آشکارساز تهاجم یا IDS جهت حفاظت از مرزهای شبکه سودمند است. در IDS بازرسی عمیق از بسته انجام می‌شود که مفهوم آن نه تنها بازرسی همه ی محدوده‌های سرپیام داده‌گرام بلکه محموله‌ی آن شامل داده‌ی لایه‌ی کاربرد نیز می‌شود. یک IDS معمولاً دارای یک پایگاه داده از اعتبارسنج‌های بسته است که بخشی از تهاجم شناخته می‌شوند. این پایگاه داده به صورت خودکار با کشف یک تهدید، به روزرسانی می‌شود. هنگامیکه بسته‌ها از IDS عبور می‌کنند، IDS تلاش می‌کند میدان‌های سرپیام و محموله را به اعتبارسنج‌های پایگاه داده‌های خود نگاشت نماید. اگر نگاشتی شناخته شود نتیجه‌ی آن برطرف شدن وضعیت هشدار است. روش دیگر **سیستم‌های ممانعت از تهاجم (IPS)**[2] است که معمولاً مشابه IDS می‌باشد با این تفاوت که علاوه بر تعریف هشدار بسته‌ها را نیز از ورود مسدود می‌کند. در فصل ۸ جزئیات بیشتری از دیواره‌های آتش و IDS تشریح خواهند شد. سرانجام این پرسش که آیا دیواره‌های آتش و IDS می‌توانندکاملاً شبکه را در برابر تهاجم ایمن کنند؟ پاسخ آشکار و قاطع خیر است، زیرا مهاجمان مرتباً روش‌های جدیدی را جهت تهدید شبکه‌ها طراحی می‌کنند که اعتبارسنج‌های آن هنوز طراحی نشده‌اند. بنابراین تنها ایمن‌سازی که دیواره‌های آتش و IDS های مبتنی بر اعتبارسنج‌ها قادرند در اختیار شبکه قرار دهند مربوط به تهدیدهای شناخته شده می‌باشد.

۴-۱۱ پروتکل آدرس‌دهی IP نسخه‌ی 6 (IPV6)

در اوایل دهه‌ی 1990 میلادی کار گروه مهندسی اینترنت (IETF) تلاشی را در جهت یافتن جایگزینی

1- Intrusion detection systems
2- Intrusion prevention systems

برای پروتکل IPV4 آغاز نمود. علت اصلی این کوشش آگاهی از این واقعیت بود که آدرس‌های IP با طول 32 بیت که پروتکل IPV4 تعریف می‌کند عملاً به اتمام رسیده بود و زیرشبکه‌های جدید و گره‌های IP به سرعت در حال افزایش و رقابت با یکدیگر جهت اتصال به شبکه‌ی جهانی اینترنت بودند. برای پاسخگویی به این نیاز فضای آدرس‌دهی IP بزرگتری در قالب یک پروتکل جدید بایستی طراحی می‌شد. پروتکل IPV6 نتیجه‌ی این تلاش شد. طراحان پروتکل جدید از فرصت بهره‌برده جوانب دیگر IPV4 را نیز بر اساس تجربیات بدست آمده از کار با ان تحلیل نمودند.

نقطه‌ای از زمان که پروتکل IPV4 و آدرس‌های آن کاملاً مورد بهره‌برداری قرار می‌گرفتند و بنابراین زیرشبکه‌های جدیدی نمی‌توانست به اینترنت متصل شود. از جمله موضوع پراهمیت از لحاظ مباحث پیرامون پروتکل جدید بود. آنچه IETF پیش‌بینی می‌نمود حکایت از این نگرانی داشت که مابین دو فاصله‌ی زمانی 2008 میلادی و یا 2018 میلادی دیگر آدرس IPV4 انحصاری جهت اختصاص به زیرشبکه‌های جدید وجود نخواهد داشت. تحلیل اخیری که در سال 2008 منتشر شده است حکایت از سال 2010 میلادی دارد. در سال 1996 میلادی **دفتر اسناد رسمی نام و شمارگان اینترنت آمریکا (ARIN)**[1] گزارشی را منتشر نمود مبنی بر آن که کلیه‌ی آدرس‌های مربوط به گروه A کاملاً اختصاص یافته است و این مهم برای آدرس‌های IP گروه B برابر 62 درصد و سرانجام برای آدرس‌های گروه C برابر 37 درصد می‌شد. اگر چه این آمارها حکایت از آن می‌کند که هنوز زمان قابل توجهی در اختیار است تا به کارگیری و پیاده‌سازی یک فناوری جدید در چنین مقیاس گسترده‌ای انجام گیرد، اما در هر حال تلاش جهت طراحی **نسل بعدی IP** که معروف به **(IPng)**[2] می‌باشد با مستند [RFC 1152] آغاز گردید. نتیجه‌ی این تلاش‌ها مشخصات نسخه‌ی 6 از IP را در مستند [RFC 2460] به ارمغان آورد. شاید این پرسش در ذهن ایجاد شود که چه اتفاقی برای IPV5 رخ داده است؟ در ابتدا تصمیم بر آن بود که پروتکل ST-2 به عنوان IPV5 معرفی شود پس از مدتی از ST-2 صرف‌نظر گردید و پروتکل RSVP به عنوان جایگزین آن برگزیده شد. اگر در این مقطع نگرانی در ذهن خواننده در رابطه با این دو پروتکل حاصل شده است پاسخ آن است که در فصل ۷ همه‌ی این موارد تشریح شده است. بنابراین فعلاً جای نگرانی نیست!

- **قالب‌بندی داده‌گرام IPV6**

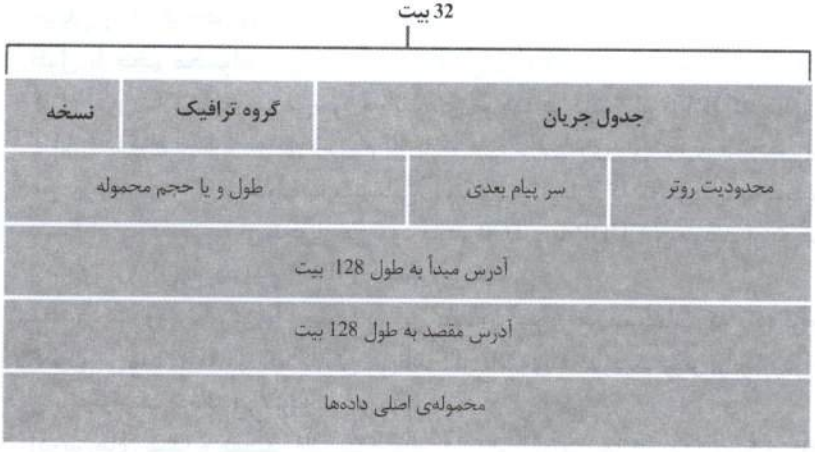
قالب داده‌گرام IPV6 در شکل ۴-۲۴ نشان داده شده است. مهم‌ترین تغییرات انجام شده در IPV6 در چارچوب خود داده‌گرام مشخص می‌باشد. این تغییرات عبارتند از:

۱- **توانمندی آدرس‌دهی توسعه یافته.** پروتکل IPV6 طول و اندازه‌ی آدرس IP را از 32 بیت به 128 بیت افزایش می‌دهد. این موضوع سبب می‌شود که هرگز کمبود آدرس IP دیگر به وجود نیاید. تقریباً با این طول آدرس IP هر متر مربع از سطح کره‌ی زمین می‌تواند شامل 2000 آدرس IP از نوع نسخه 6 شود. به عبارت دیگر هر دانه ماسه روی زمین قابلیت آدرس‌دهی IP را پیدا می‌کند. علاوه بر آدرس‌دهی تک **فراخوانی**[3] و **چند فراخوانی**[4] پروتکل آدرس‌دهی IPV6 یک گونه آدرس‌دهی جدید را نیز امکان‌پذیر نموده است که اصطلاحاً **هر**

1- American registry for Internet numbers
2- Next generation IP
3- Unicast
4- Multicast

فراخوانی[1] نامیده می‌شود. این نوع آدرس اجازه‌ی تحویل یک داده‌گرام به هر یک از گروه میزبان‌ها را می‌دهد. به عبارت دیگر از این ویژگی می‌توان جهت ارسال یک پیام HTTP GET به نزدیک‌ترین سامانه‌های قرینه‌ای که شامل یک سند ویژه است استفاده نمود.

۲- **سرپیام اختصاری به طول 40 بایت.** تعدادی از میدان‌های تعریف شده در سرپیام IPV4 از پروتکل نسخه ی 6 حذف شده و یا حالت اختیاری را یافته‌اند. سرپیام با طول ثابت 40 بایت در این پروتکل جدید اجازه ی پردازش سریعتر داده‌گرام‌های IP را فراهم می‌آورد. بهره‌گیری از امکانات اختیاری رمزگذاری پردازش انعطاف‌پذیر حوزه‌ی اختیارهای سرپیام را نتیجه می‌دهد.



شکل ۴-۲۴ ♦ قالب داده‌گرام IPV6

۳- **اولویت و نشانه‌گذاری جریان‌های داده.** پروتکل IPV6 تفسیر ویژه‌ای را از عبارت **جریان** در شبکه دارد. مستندهای [RFC 1152] و [RFC 2460] این عبارت را این گونه تعریف می‌کنند. « اعمال برچسب به بسته‌هایی که متعلق به جریان‌های خاص می‌باشند از سوی فرستنده‌ای که به درخواست ویژه‌ای پاسخ می‌دهد. از جمله‌ی این درخواست‌ها خدمات زمان واقعی و با کیفیت خدمات غیر پیش‌فرض می‌باشند.»

به عنوان مثال انتقال‌های صوتی و تصویری در قالب یک جریان ممکن است قلمداد شوند. از طرف دیگر کاربردهای قدیمی مانند انتقال فایل و یا پست الکترونیکی جریان به شمار نمی‌اینند. این امکان وجود دارد که ترافیک حمل شده از سوی یک کاربر با اولویت بالا مانند بهره‌وران از خدمات با کیفیت بالا در رده‌ی جریان به شمار آیند. آشکار است که طراحان IPV6 نیاز به ممیزی بین جریان‌ها را ضروری قلمداد کرده‌اند حتی اگر مفهوم دقیق یک جریان هنوز کاملاً مشخص نشده است. سرپیام IPV6 دارای یک محدوده نوع ترافیک به طول 8 بیت است. از این میدان مانند دامنه‌ی TOS در IPV4 می‌توان جهت اختصاص اولویت به داه‌گرام‌های کاربردهای ویژه مانند ICMP در مقابل یک شبکه‌ی خبری استفاده نمود.

از مقایسه‌ی دو شکل ۴-۲۴ و ۴-۱۳ می‌توان به ساختار ساده شده‌ی داده‌گرام در IPV6 پی‌برد. دامنه‌های ذیل از جمله مواردی است که در IPV6 تعریف شده‌اند:

1- Anycast

- **نسخه.** این دامنه‌ی به طول 4 بیت شماره نسخه‌ی IP را مشخص می‌کند. بدیهی است ارزش حمل شده در این دامنه برای IPV6 برابر مقدار 6 می‌باشد اما چنانچه این ارزش به 4 تغییر یابد سبب تعریف یک داده‌گرام مجاز IPV4 نمی‌شود. چنانچه این تغییر امکان‌پذیر می‌شد انتقال داده‌گرام‌ها از IPV4 به IPV6 و عکس نیز بسیار ساده می‌شد.
- **نوع ترافیک.** این محدوده‌ی به طول 8 بیت در سرپیام IPV6 مشابه دامنه‌ی TOS در IPV4 است.
- **برچسب جریان.** قبلاً اشاره گردید این محدوده‌ی به طول 20 بیت جهت شناسایی یک جریان از داده‌گرام‌ها مورد بهره‌برداری قرار می‌گیرد.
- **طول یا حجم محموله.** این دامنه دارای یک ارزش 16 بیت است که به گونه‌ی یک عدد کامل بدون علامت تعریف می‌شود و بایت‌های موجود در داده‌گرام IPV6 که همان محموله‌ی اصلی داده‌گرام است و در پی سرپیام 40 بایت داده‌گرام جاری می‌شود را مشخص می‌کند.
- **سرپیام بعدی.** این محدوده به شناسایی پروتکلی می‌پردازد که محتوی یا محدوده‌ی داده در داده‌گرام مورد نظر به آن تحویل می‌شود مانند TCP و یا UDP. این میدان ارزش‌های مشابه دامنه‌ی پروتکل در سرپیام IPV4 را اختیار می‌کند.
- **محدودیت روتر.** ارزش یا محتوی در این دامنه از سوی هر مسیریاب که داده‌گرام را به جلو گسیل می‌کند به مقدار یک کاهش می‌یابد. اگر دامنه‌ی محدودیت روتر برابر ارزش **صفر** شود از داده‌گرام صرف‌نظر می‌شود.
- **آدرس‌های مبدا و مقصد.** قالب‌بندی‌های گوناگون آدرس 128 بیت پروتکل IP را که در مستند [RFC 4291] تعریف شده است، مشخص می‌کند.
- **داده.** این بخش شامل محموله‌ی داده‌گرام IPV6 است. هنگامی که داده‌گرام به مقصد خود وارد می‌شود محموله از داده‌گرام جدا می‌شود و به سوی پروتکل مشخص شده در دامنه سرپیام بعدی گسیل می‌شود.

آنچه در بالا به آن اشاره شد، اهداف مورد نظر در درون ساختار یک داده‌گرام IPV6 را نشان می‌دهد. مقایسه‌ی قالب داده‌گرام IPV6 در شکل ۴-۲۴ با داده‌گرام IPV4 شکل ۴-۱۳ نشان می‌دهد که از چندین میدان موجود در IPV4 در پروتکل جدید IP یعنی نسخه‌ی IPV6 صرف‌نظر شده است.

۴- **قطعه‌بندی و پیوند مجدد.** پروتکل IPV6 اجازه‌ی قطعه‌بندی و پیوند مجدد داده‌ها را در مسیریاب‌های میانی یک مسیر انتقال نمی‌دهد. این عملیات فقط می‌تواند از سوی مبدا و مقصد انجام پذیرد. اگر یک داده‌گرام IPV6 که از سوی روتر دریافت شده است جهت گسیل به پیوند خروجی بسیار بزرگ باشد آنگاه مسیریاب داده‌گرام را حذف نموده و یک پیام خطای ICMP در قالب « بسته بسیار بزرگ است» را به سوی فرستنده گسیل می‌نماید. فرستنده آنگاه می‌تواند داده را مجدداً با استفاده از داده‌گرام‌های IP با طول کوچکتر گسیل نماید. قطعه‌بندی و پیوند مجدد قطعه‌ها، عملیات توام با مصرف زمانی می‌باشند و حذف این مورد از مسیریاب‌ها و قرار گرفتن آن‌ها به عنوان بخش عملیاتی سیستم‌های انتهایی سبب تسریع عملیات انتقال IP در بین شبکه خواهد شد.

۵- **کُد مجموع مقابله‌ی سرپیام.** از آنجائیکه لایه‌ی انتقال مانند TCP و یا UDP به همراه لایه‌ی پیوند داده مانند پروتکل اترنت و غیره در لایه‌های اینترنت عملیات ارزیابی خطا را با استفاده از

کد مجموع مقابله (Checksum) انجام می‌دهند. طراحان IP معتقد به کفایت این عملیات در لایه‌های دیگر بودند و بنابراین آن را از لایه‌ی شبکه حذف نمودند. علت آن نیز تسریع در پردازش بسته‌های IP بود. با یادآوری بحث IPV4 در بخش‌های قبل از آنجائیکه سرپیام IPV4 حامل دامنه‌ی TTL مشابه به دامنه‌ی محدودیت روتر در سرپیام IPV6 می‌شد بنابراین سرپیام کُد مجموع مقابله در IPV4 نیازمند محاسبه‌ی مجدد آن در هر مسیریاب بود. بنابراین مانند عملیات قطعه‌بندی و پیوند مجدد، این عملیات نیز در IPV4 پر هزینه بود.

۶- **عملیات.** اگرچه دامنه‌ی اختیار به عنوان بخشی از سرپیام استاندارد IP قلمداد نمی‌شود اما به طور کامل کنار گذارده نشده است بلکه دامنه‌ی اختیار می‌تواند یکی از سرپیام‌های بعدی اشاره شده ی ممکن در بین سرپیام IPV6 به شمار آید. مانند سرپیام های پروتکل‌های TCP و یا UDP که می‌توانند به عنوان سرپیام بعدی در بین یک بسته‌ی IP به شمار آیند. محدوده‌ی اختیار نیز چنین نقشی را ایفا می‌کند. حذف دامنه‌ی اختیار سبب ثابت شدن طول سرپیام IP به 40 بایت می‌شود.

با یادآوری این نکته که پروتکل ICMP از سوی گره‌های IP جهت گزارش وضعیت خطا و فراهم‌آوری اطلاعات محدود شده‌ای مانند پاسخ اِکو به یک پیام ping برای یک سیستم انتهایی است. نسخه‌ی جدید ICMP در مستند [RFC 4443] برای IPV6 تعریف شده است. علاوه بر سازمان‌دهی مجدد تعریف‌های نوع و کُد ICMP قبلی در پروتکل ICMPV6 کدها و نوع‌های جدید که مورد نیاز عملیات IPV6 می‌باشد نیز افزوده شده است. این کدها عبارتند از نوع: «پیام بسیار بزرگ است» و «عوامل اختیاری IPV6 شناخته نشده». که به ترتیب برای نوع و کد خطا می‌باشند.

۱۲-۴ گُذر از پروتکل IPV4 به IPV6

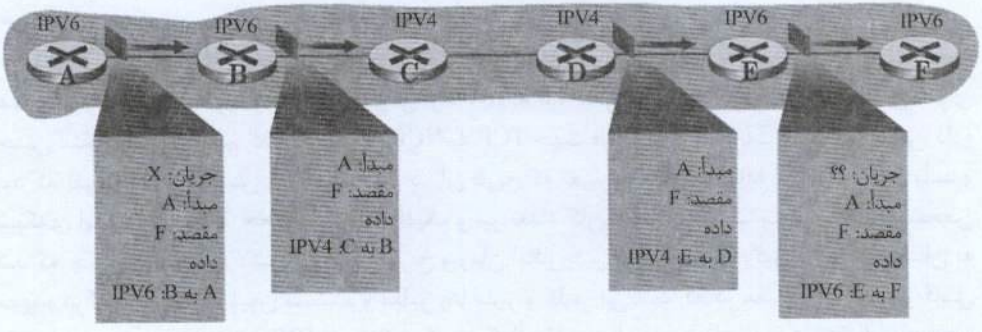
پس از بررسی جزئیات فنی پروتکل‌های IPV6 و IPV4 اکنون جنبه‌های عملی موضوعات مربوط به این دو ، تحلیل می‌شود. پرسش مهم قابل طرح آن است که چگونه شبکه‌ی اینترنت که بر مبنای IPV4 بنیان نهاده شده است به پروتکل IPV6 تبدیل می‌شود؟ به عبارت دیگر چالش‌ها و راه‌کارهای گذر از IPV4 به IPV6 چیست؟ دشواری احتمالی از آنجایی ناشی می‌شود که سیستم‌های مجهز به توان‌مندی‌های IPV6 را می‌توان تبدیل به IPV4 نمود و به عبارت دیگر داده‌گرام‌های IPV6 را می‌توان در قالب IPV4 ارسال، دریافت و یا مسیریابی نمود اما داده‌گرام‌های IPV4 را نمی‌توان در قالب IPV6 آن‌ها ارسال، دریافت ویا مسیریابی نمود. راه‌کارهای گوناگونی پیرامون این دشواری پیشنهاد شده است.

یکی از این راه‌کارها تعریف یک نشان تاریخ و زمان برای تغییر است که در زمان و تاریخ مشخصی همه‌ی سیستم‌های اینترنت خاموش و سپس از IPV4 به IPV6 ارتقاء یابد. آخرین تجربه‌ی موجود برای چنین انتقال فناوری مهم گذر از پروتکل NCP به TCP جهت فراهم‌آوری امکانات انتقال مطمئن داده بود که تقریباً 25 سال پیش روی داد. حتی در آن تاریخ که تغییرات مستند [RFC 801] مطرح شد و شبکه‌ی اینترنت از لحاظ حجم و میزان ترافیک و نیز تعداد کاربران بسیار اندک بود آشکارا مشخص شد که چنین رویکرد بهره‌گیری از نشان تاریخ و زمان امکان‌پذیر نمی‌باشد. بهره‌گیری از چنین نشان به مفهوم درگیری صدها میلیون سیستم و میلیون‌ها مدیر و کاربر می‌باشد که در مقیاس امروز غیر قابل تصور است. مستند [RFC 4213] دو رویکرد که هر کدام را به تنهایی می‌توان جهت انتقال تدریجی میزبان‌ها و مسیریاب‌ها IPV4 به دنیای جدید IPV6 مورد استفاده قرار داد پیشنهاد نموده است.این

رویکردها در حقیقت به گونه‌ی عکس عمل می‌نماید یعنی تجهیزات مجهز به IPV6 تدریجاً جایگزین IPV4 می‌شوند.

یکی از این رویکردها معرفی تجهیزات مجهز به IPV6 با کاربرد دوگانه می‌باشند به این مفهوم که ابزار مورد استفاده مجهز به امکانات هر دو پروتکل IPV4 و IPV6 خواهند شد. چنین گره‌های شبکه‌ی اینترنت اکنون با عنوان گره‌های IPV6/IPV4 در مستند 4213 شناخته شده‌اند و قادر به ارسال و دریافت هر دو نوع داده‌گرام می‌باشند. هنگامیکه این تجهیزات با نوع IPV4 خود تعامل می‌کنند از داده‌گرام‌های نوع IPV4 بهره می‌گیرند و هنگامی که وارد فضای تعامل با گره‌های IPV6 می‌شوند از داده‌گرام‌ها نوع IPV6 استفاده می‌کنند. بدیهی است گره‌های IPV6/IPV4 بایستی هر دو نوع آدرس‌دهی IP را نیز شامل شوند. علاوه بر آن‌ها بایستی قادر به تشخیص اینکه آیا گره‌ی دیگری فقط مجهز به یکی از دو مورد IPV6 و یا IPV4 است نیز باشند. البته این دشواری اخیر را می‌توان از طریق بهره‌گیری از پروتکل DNS که قادر به بازگردانی یک آدرس IPV6 بنابر توان‌مندی گره‌ی مورد نظر در هر یک از دو پروتکل اشاره شده است به یکی از آدرس‌های از نوع IPV6 و یا IPV4 برطرف نمود. بدیهی است اگر گره‌ی صادر کننده‌ی درخواست DNS فقط از نوع IPV4 باشد پروتکل DNS تنها یک آدرس IPV4 را بازگشت می‌دهد.

در رویکردی دوگانه چنانچه فرستنده و یا گیرنده هر کدام فقط از نوع IPV4 باشند بایستی از داده‌گرام IPV4 نیز بهره گرفت. بنابراین این امکان وجود دارد که دو گره‌ی مجهز به IPV6 در نهایت داده‌گرام‌های IPV4 را بین یکدیگر ارسال نمایند. این مورد در شکل ۴-۲۵ نشان داده شده است. در این شکل فرض شده است گره‌ی A مجهز به امکانات IPV6 است و مایل به ارسال یک داده‌گرام به گره‌ی F می‌باشد. گره‌ی F نیز از پروتکل IPV6 استفاده می‌کند. بدیهی است هر دو گره‌ی A و B می‌توانند داده‌گرام‌های IPV6 را بین خود مبادله نمایند اما گره‌ی B بایستی داده‌گرام‌های از نوع IPV4 را به C ارسال نماید. یقیناً بخش داده‌ی IPV6 را می‌توان در درون بخش داده‌ی IPV4 نسخه‌برداری نمود و نیز نگاشت مناسب آدرس مورد نیاز را نیز انجام داد. اما در هنگام انجام عملیات تبدیل از IPV6 به IPV4 به یک دامنه‌ی ویژه در داده‌گرام‌های IPV6 مانند دامنه‌ی شناسه‌ی جریان برخورد می‌کنیم که معادل آن در IPV4 وجود ندارد. نتیجه آنکه اطلاعات درون این محدوده مفقود می‌شود. بنابراین حتی اگر گره‌های E و F قادر به مبادله‌ی داده‌گرام‌های از نوع IPV6 باشند، داده‌گرام‌های IPV4 وارد شده از D به E شامل همه‌ی میدان‌های موجود در داده‌گرام IPV6 ارسالی از سوی A نمی‌باشند.



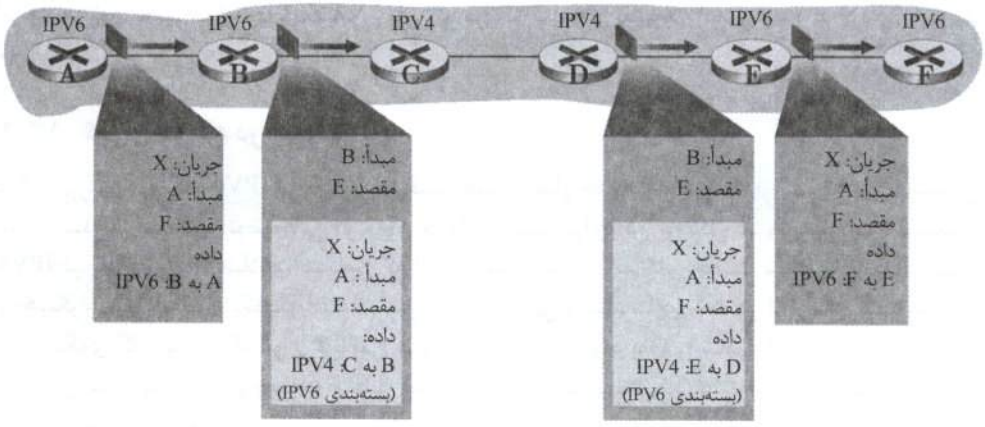
شکل ۴-۲۵ ♦یک رویکرد دوگانه

رویکرد دوگانه‌ی جایگزین در روش نخست که در بالا به آن اشاره رفت و در مستند 4213 نیز تعریف شده است رویکرد اصطلاحاً **تونل زدن**[1] نام گرفته است. در این روش دشواری بیان شده در شکل ۴-۲۵ برطرف می‌شود. زیرا به عنوان مثال به گره‌ی E اجازه می‌دهد که داده‌گرام‌های اصلی IPV6 سرچشمه گرفته از A را دریافت نماید. هدف اصلی روش تونل‌زنی آن است که اگر فرض شود دو گره IPV6 مانند B و E در شکل ۴-۲۵ با بهره‌گیری از داده‌گرام‌های IPV6 مایل به تعامل با یکدیگر باشند و این عمل از طریق گره‌های واسط IPV4 که به یکدیگر متصل شده‌اند انجام گیرد آنگاه مجموعه‌ی مسیریاب‌های IPV4 ما بین دو روتر IPV6 را اصطلاحاً یک **تونل** می‌نامند. شکل ۴-۲۶ این مورد را نشان می‌دهد. در روش تونل‌زنی روتر IPV6 در سمت ارسال تونل مانند B تمامی بخش‌های داده‌گرام IPV6 را در درون بخش محدوده‌ی داده‌ی متعلق به داده‌گرام IPV4 قرار می‌دهد. این داده‌گرام IPV4 سپس به سمت یک مسیریاب IPV6 مانند E آدرس‌دهی می‌شود. آن مسیریاب در سمت دریافت تونل قرار گرفته است. آنگاه داده‌گرام اشاره شده به نخستین مسیریاب در تونل مانند C وارد می‌شود. مسیریاب‌هایIPV4 واسطی در تونل این داده‌گرام از نوع IPV4 را در بین خود مانند داده‌گرام‌های IPV4 متعلق به خود مسیریابی می‌کنند بدون آنکه از محموله‌ی داده‌گرام که در درون خود حامل یک داده‌گرام IPV6 می‌باشد آگاهی یابند. مسیریاب IPV6 در سمت دریافت تونل سرانجام این داده‌گرام IPV4 را که به سمت او آدرس‌دهی شده است دریافت و از درون آن داده‌گرام IPV6 را خارج می‌نماید. پس از خارج نمودن داده‌گرام IPV6 مسیریاب ذینفع این داده‌گرام را به روتر مجاور خود به گونه‌ی یک داده‌گرام IPV6 منتقل می‌کند.

دیدگاه منطقی :



دیدگاه فیزیکی:



شکل ۴-۲۶ ♦ تونل زدن بین پروتکل‌های **IPV4** و **IPV6**

1- Tunnling

اگر چه گزینش IPV6 در ابتدا به کُندی آغاز شد اما در حال حاضر شتاب در به کارگیری آن رو به افزایش گذاشته است. اکثر نهادها و سازمان‌های مهم در آمریکای شمالی با سرعت تمام از اواسط سال 2008 میلادی گذر به IPV6 را غاز نموده‌اند. گسترش تجهیزات مجهز به IPV6 مانند گوشی‌های همراه نسل چهارم به بعد نیاز برای به کارگیری گسترده از IPV6 را بیشتر نموده است. در اروپا **برنامه‌های اشتراک نسل سوم** یا [3GPP 2012][1] روش آدرس‌دهی استاندارد خود را بر اساس IPV6 قرار داده‌اند. این برنامه‌ها جهت کاربردهای سیار چند رسانه‌ای طراحی شده‌اند. اگر چه IPV6 به گونه‌ی گسترده در 10 سال نخست حیات جوان خود مورد بهره‌برداری قرار نگرفت اما آینده به سرعت در حال تغییر به سمت آن می‌باشد. یکی از معماران برجسته‌ی IPV6 به نام برایان کارپنتر عنوان نموده بود که « من همیشه به پیاده‌سازی IPV6 به عنوان یک فرآیند 15 ساله که آغاز ان از 1995 است نگاه کرده‌ام» بنابراین دیدگاه ما اکنون نزدیک به سه چهارم از این زمان را پشت‌سر نهاده‌ایم!

درس مهمی که از تجربه‌ی IPV6 گرفته شده است آن است که اعمال تغییرات در پروتکل‌های لایه‌ی شبکه بسیار دشوار است. از اوایل دهه‌ی 1990 میلادی تعداد بی‌شماری پروتکل لایه‌ی شبکه طراحی و پیشنهاد شده است که هر یک از آن‌ها وعده‌ی دگرگونی شگرف در اینترنت را می‌داد. اما اکثریت آن‌ها از نفوذ محدودی تا به امروز در اینترنت برخوردار شده‌اند. این پروتکل‌ها عبارتند از پروتکل‌های چندپخشی، پروتکل‌های ذخیره‌ی منابع و تا حدودی پروتکل IPV6. بدیهی است معرفی پروتکل‌های جدید به درون لایه‌ی شبکه را می‌توان مانند جایگزین نمودن بنیادهای یک خانه دانست. عمل جایگزینی بنیادهای یک خانه بدون تخریب تمامی و یا بخشی از آن خانه و نیز اسکان موقت در نقطه‌ای دیگر امکان‌پذیر نمی‌باشد. از طرف دیگر شبکه‌ی اینترنت شاهد به کارگیری سریع پروتکل‌های جدید در لایه‌ی کاربرد بوده است از جمله در زمینه‌های وب، پیام بی‌درنگ، P2P، بازی‌های توزیع شده، جریان‌های صوتی و تصویری مانند اسکایپ از جمله مواردی است که می‌توان نام برد. معرفی پروتکل‌های جدید لایه‌ی کاربرد مانند اضافه نمودن یک لایه رنگ به دیوارهای خانه می‌باشد که انجام آن نسبتاً ساده است و اگر موفق به گزینش یک رنگ جذاب شوید همسایگان شما نیز تلاش در نسخه‌برداری از رنگ شما می‌کنند. در آینده این انتظار می‌رود که تغییراتی در لایه‌ی شبکه روی دهد اما فرآیند این تغییرات از لحاظ مقیاس زمانی به مراتب کندتر از تغییرات اعمال شده در لایه‌ی کاربرد خواهد بود.

۱۳-۴ گذری کوتاه در امنیت IP

هنگام بررسی پروتکل IPV4 در بخش‌های قبلی همین فصل خدمات ارایه شده از سوی این پروتکل تحلیل شد اما هیچ‌گونه توضیحاتی در مورد خدمات امنیتی برای این پروتکل عنوان نگردید. پروتکل IPV4 در دهه‌ی 1970 میلادی هنگامی که از اینترنت به عنوان رسانه‌ای جهت تبادل اطلاعات بین پژوهشگران مورد اعتماد یکدیگر استفاده می‌گردید، طراحی و پیاده‌سازی شد. در آن زمان پیاده‌سازی یک شبکه‌ی کامپیوتری که انواع گوناگون فناوری‌های لایه‌ی پیوند داده را شامل شود نگرانی اصلی به شمار می‌آمد و بنابر این توجهی به مسایل امنیتی شبکه معطوف نمی‌شد. علت اصلی نیز اعتماد متقابل بین کاربران در آن زمان بود.

اما امروز امنیت شبکه یکی از نگرانی‌های اصلی کاربران است. پژوهشگران اینترنت تمرکز خود را بر

1-Third generation partnership program

روی آن دسته از پروتکل‌های لایه‌ی شبکه قرار داده‌اند که قادر به تامین انواع خدمان امنیتی می‌باشند. یکی از این پروتکل‌ها **امنیت IP (IP sec)**[1] می‌باشد. این پروتکل یکی از شناخته‌ترین پروتکل‌های امنیت لایه‌ی شبکه است که به گونه‌ی گسترده در **شبکه‌های مجازی خصوصی (VPNs)**[2] بکار گرفته شده است. اگرچه پروتکل IPSec و مکانیزم‌های رمزگذاری آن با همه‌ی جزئیات در فصل ۸ بررسی شده‌اند اما در هر حال معرفی مقدماتی این پروتکل در این مقطع ضروری به نظر می‌رسد.

پروتکل IPSec به گونه‌ای طراحی شده است که سازگار با هر پروتکل IPV4 و IPV6 باشد به ویژه آنکه جهت بهره‌برداری مؤثر از IPSec نیازی به جایگزینی پشته‌ها پروتکلی در همه‌ی مسیریاب‌ها و میزبان‌های اینترنت نمی‌باشد. به عنوان مثال بهره‌گیری از الگوی انتقال که یکی از دو الگوی IPSec می‌باشد. اگر دو میزبان مایل به ارتباط پنهانی با یکدیگر باشند، پروتکل IPSec کافی است که فقط در همین دو میزبان حضور داشته باشد. به عبارت دیگر همه ی مسیریاب‌ها و میزبان‌های دیگر می‌توانند به اجرای IPV4 معمولی ادامه دهند.

در این مقطع از تحلیل IPSec فقط بر روی انتقال این پروتکل تمرکز می‌شود. در این الگو دو میزبان نخست یک تعامل IPSec را بین خود برقرار می‌کنند. بنابراین پروتکل IPSec از نوع اتصال‌گرا به شمار می‌آید. هنگامی که تعامل برقرار شد همگی قطعه‌های TCP و UDP بین این دو میزبان که از خدمات امنیتی IPSec بهره‌مند شده‌اند ارسال و دریافت می‌شوند. در سمت ارسال لایه‌ی انتقال قطعه‌ای را به IPSec تحویل می‌دهد و این پروتکل سپس قطعه را رمزگذاری و دامنه‌های امنیتی اضافه‌تری را به آن قطعه پیوند می‌دهد. پس از انجام این فرآیند قطعه‌ی نهایی نتیجه شده در قالب یک داده‌گرام معمولی IP بسته بندی می‌شود. میزبان فرستنده سپس داده‌گرام را وارد شبکه‌ی اینترنت می‌کند و در نهایت قطعه را رمزگشایی و آن را به لایه‌ی انتقال میزبان مقصد تحویل می‌دهد. خدماتی که از سوی یک تعامل IPSec تامین می‌شوند عبارتند از:

- **توافق‌های رمزگذاری.** روش‌هایی که اجازه‌ی توافق الگوریتم‌های رمزگذاری و کلیدهای وابسته به آن را به دو میزبان در حال تعامل با یکدیگر می‌دهند.
- **رمزگذاری محموله داده‌گرام IP.** هنگامی که میزبان ارسال کننده قطعه‌ای را از لایه‌ی انتقال خود دریافت می‌کند پروتکل IPSec محموله را رمزگذاری می‌نماید و محموله‌ی رمزگذاری شده فقط از سوی میزبان دریافت کننده قابل رمزگشایی می‌باشد.
- **یک‌پارچگی داده.** پروتکل IPSec اجازه می‌دهد میزبان دریافت کننده از عدم تغییر اطلاعات درون محدوده‌های سرپیام و محموله رمزگذاری شده به هنگام عبور از آن مسیر بین مبدا و مقصد اطمینان حاصل نماید.
- **اعتبارسنجی مبدا.** هنگامی که یک میزبان داده‌گرام از نوع IPSec را از یک منبع قابل اعتماد دریافت دارد، این اطیمینان به میزبان داده می‌شود که آدرس IP مبدا در داده‌گرام در حقیقت همان منبع واقعی ارسال داده‌گرام است.

هنگامی که دو میزبان اقدام به تعامل از نوع IPSec بین خود می‌کنند همه‌ی قطعه‌های TCP و UDP مبادله شده بین آن‌ها رمزگذاری و اعتبارسنجی می‌شود. به عبارت دیگر IPSec یک چتر امنیتی را برای همه‌ی ارتباط‌های بین دو میزبان و همه‌ی کاربردهای شبکه فراهم می‌آورد. یک سازمان

1- IP security
2- Virtual private networks

می‌تواند با بهره‌گیری از پروتکل IPSec ارتباط ایمنی را در شبکه‌ی ناامن اینترنت برای کاربردهای خود فراهم آورد. جهت روشن شدن بیشتر موضوع حالتی را در نظر بگیرید که یک شرکت تعداد قابل توجهی بازاریاب سیار دارد. هر یک همراه خود یک کامپیوتر همراه از آن شرکت را حمل می‌کنند. اگر فرض شود این افراد مرتباً جهت دریافت آخرین اطلاعات مربوط به تغییر قیمت‌ها، صورت کالاهای جدید و غیره مجبور به ارتباط با سرور دفتر مرکزی شرکت باشند که این اطلاعات بر روی آن ذخیره شده است و علاوه بر این موارد اگر این فروشندگان نیاز به تبادل اطلاعات و اسناد حیاتی و حساس بین یکدیگر را نیز داشته باشند آنگاه چگونه با بهره‌گیری از IPSec این مهم امکان‌پذیر می‌شود؟ ممکن است حدس زده باشید! از طریق نصب پروتکل IPSec بر روی همه‌ی میزبان‌ها یعنی سرور و کامپیوترهای سیار فروشندگان این امکان پدید می‌آید. با نصب IPSec در این میزبان‌ها هر گاه یک فروشنده نیاز به ارتباط با سرور و یا همکار دیگر خود را داشته باشد انجام تعامل بین آن‌ها به گونه‌ای ایمن صورت می‌پذیرد.

۴-۱۴ الگوریتم‌های مسیریابی

تا کنون در این فصل موضوعات مربوط به لایه‌ی شبکه از لحاظ چگونگی انجام عملیات گسیل به جلو بررسی گردید. آموختیم هنگامی که یک بسته به مسیریاب وارد می‌شود، روتر یک جدول گسیل به جلو را شاخص‌گذاری می‌کند و سپس واسط پیوندی که بسته بایستی به سمت آن هدایت شود را بدست می‌آورد. همچنین آموختیم که الگوریتم‌های مسیریابی که در شبکه‌ی درون مسیریاب‌ها پیاده‌سازی شده‌اند اطلاعات مورد نیاز جهت پیکربندی این جداول را محاسبه و بین یکدیگر مبادله می‌کنند. شکل ۴-۲ رابطه‌ی بین الگوریتم‌های مسیریابی و جداول گسیل به جلو را نشان می‌دهد. پس از بررسی نسبتاً عمیق عملیات گسیل به جلو برای داده‌گرام‌ها اکنون توجه خود را به سمت یکی دیگر از عناوین مهم این فصل قرار می‌دهیم که عملیات حیاتی و حساس مسیریابی لایه‌ی شبکه است. مهم نیست که آیا لایه‌ی شبکه داده‌گرام‌ها را با توجه به روش راهیابی بین مبدا و مقصد از طریق مسیرهای گوناگون انجام می‌دهد و یا آنکه به صورت VC همه‌ی بسته‌ها از طریق یک مسیر، بین مبدا و مقصد مبادله می‌شوند. در هر دو حالت لایه‌ی شبکه بایستی مسیری که بسته‌ها از فرستنده‌ها به طرف گیرنده‌ها پویش می‌کنند را به دست آورد. مسیر محاسبه شده بایستی بهترین مسیر در بین گزینش‌های ممکن باشد.

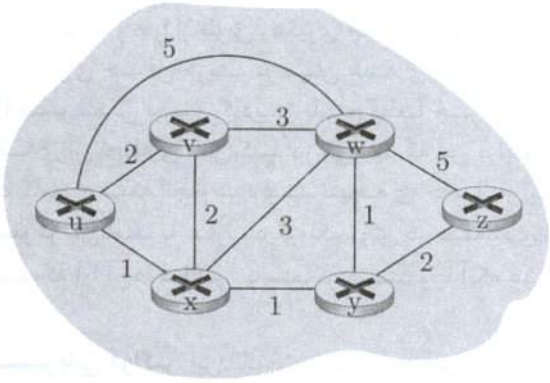
معمولاً در شبکه‌ی اینترنت یک میزبان مستقیماً با یک مسیریاب پیوند دارد که اصطلاحاً به آن **مسیریاب پیش‌فرض** و یا نخستین **هاپ روتر**[1] برای آن میزبان نامیده می‌شود. این عبارت به مفهوم پرش کوتاه است. هرگاه یک میزبان بسته ای را ارسال می‌کند آن بسته به مسیریاب پیش‌فرض آن منتقل می‌شود. معمولاً روترهای پیش‌فرض میزبان مبدا را **مسیریاب مبدا** و میزبان مقصد را **مسیریاب مقصد** می‌خوانند. دشواری مسیریابی یک بسته از مبدا به مقصد آشکارا تبدیل به چالش مسیریابی بسته از روتر مبدا به مسیریاب مقصد است.

هدف الگوریتم‌های مسیریابی بنابراین روشن است. با در اختیار داشتن مجموعه‌ای از مسیریاب‌ها و پیوندهای ارتباطی بین آن‌ها یک الگوریتم مسیریابی وظیفه‌ی یافتن بهترین مسیر ممکن از روتر مبدا و مقصد را بر عهده دارد. معمولاً یک مسیر بهینه به مفهوم کم هزینه‌ترین مسیر است. بدیهی است در

1- Hop router

عمل دستیابی این هدف دشوارتر از آن چیزی است که تصور می‌شود. زیرا به عنوان مثال محدودیت‌هایی ممکن است مانع از تصمیم‌گیری سریع و بهینه شوند. این محدودیت‌ها می‌توانند از جمله شامل قاعده‌ای شوند که به عنوان مثال مسیریاب x از سازمان y مجاز به گسیل هیچ‌گونه بسته‌ای که از سازمان z سرچشمه گرفته است نمی‌شود.

از گراف جهت جمع‌بندی مسایل مسیریابی استفاده می‌شود. لازم به یادآوری است یک گراف G= (N,E) مجموعه‌ای از N گره وتعداد E یال است که در آن هر یال عبارت است از یک جفت گره از N. در چارچوب مسیریابی لایه‌ی شبکه گره‌های گراف مسیریاب‌ها می‌باشند که در درون آن‌ها چگونگی گسیل بسته‌ها به جلو انجام می‌شود و یال‌ها نیز پیوندهای ارتباطی فیزیکی هستند که بین این مسیریاب‌ها قرار دارند و آن‌ها را با یکدیگر ارتباط می‌دهند. تصویر خلاصه‌ای از این گراف برای یک شبکه‌ی کامپیوتری در شکل ۴-۲۷ نشان داده شده است.



شکل ۴-۲۷ ♦ الگوی خلاصه شده گراف از یک شبکه‌ی کامپیوتری

همانگونه که در شکل ۴-۲۷ نشان داده شده است هر یال دارای یک ارزش است که هزینه‌ی آن را مشخص می‌کند. مشخصه‌ی این هزینه برای یال معمولاً طول فیزیکی پیوند است به عنوان مثال یک پیوند بین قاره‌ای زمینی ممکن است هزینه‌ی به مراتب بیشتری در مقابل یک پیوند هوایی اختیار نماید، سرعت پیوند و یا هزینه‌ی مالی آن نیز از جمله عوامل وابسته به هزینه‌ی یال می‌باشند. از نقطه‌نظر بحث در این مقطع فرض بر آن است که هزینه‌ی هر یال مشخص شده است و نیازی نیست نگران چگونگی محاسبه‌ی آن باشیم. برای هر یال (x , y) در مجموعه E از عبارت c(x , y) به عنوان هزینه‌ی یال بین دو گره‌ی x و y استفاده می‌شود. چنانچه جفت (x , y) متعلق به E نباشد آنگاه $c(x, y) = \infty$ می‌شود. در ادامه‌ی بحث گراف‌ها مورد بررسی همگی بدون جهت در نظر گرفته شده‌اند. به عبارت دیگر یال‌های گراف هیچگونه سویی را اختیار نمی‌کنند. بنابراین یال (x , y) مشابه یال (y , x) است و c(x , y) = c(y , x) و نیز گره‌ی y به عنوان **مجاور** گره‌ی x است اگر (x , y) متعلق به مجموعه‌ی E باشد.

با فرض آنکه هزینه‌های اختصاص یافته به یال‌های گراف با هدف اختصار نمودن آن می‌باشد. بنابراین هدف طبیعی یک الگوریتم مسیریابی شناسایی حداقل هزینه‌ی مسیر بین مبدا و مقصد است. یادآوری می‌شود که یک مسیر در گراف G = (N , E) مجموعه‌ای از گره‌های $(x_1, x_2, \ldots, x_p)$ است به گونه‌ای هر جفت (x_1, x_2)، (x_2, x_3)، . . . ، (x_{p-1}, x_p) یال‌هایی در مجموعه‌ی E می‌باشند.

هزینه‌ی یک مسیر $(x_1, x_2, ..., x_p)$ به شکل ساده‌ای مجموع همه‌ی هزینه‌ی یال‌های در طول مسیر است به عبارت دیگر:

$$c(x_1, x_2) + c(x_2, x_3) + ... + c(x_{p-1}, x_p) \qquad (۳-۴)$$

با دراختیار داشتن هر دو گره‌ی x و y تعداد قابل توجهی از مسیرها بین دو گره وجود دارد که هر کدام آن‌ها شامل هزینه‌ای می‌باشند. یک یا چندین مسیر از این نوع دارای یک **مسیر با حداقل هزینه** است. در شکل ۴-۲۷ به عنوان مثال حداقل هزینه‌ی مسیر بین گره‌ی مبدا u و گره مقصد w برابر است با (u , x , y , w) و هزینه‌ی آن 3 می‌باشد. بدیهی است اگر همه‌ی یال‌ها در گراف هزینه‌ی مشابه اختیار کنند آنگاه مسیریاب با حداقل هزینه یک **کوتاه‌ترین مسیر** که به مفهوم کم‌ترین تعداد پیوندها بین مبدا و مقصد است نیز به شمار می‌آید.

به عنوان مثال اگر تلاش کنید مسیر با حداقل هزینه‌ای را از u به z در شکل ۴-۲۷ جستجو نمائید. مانند بسیاری از افراد مسیر بین u و z را از طریق رهگیری چند مسیر و متقاعد نمودن خود که مسیر گزینش شده از سوی شما دارای حداقل هزینه در میان همه ی مسیرهای ممکن است گزینش می‌نمائید. آیا همه‌ی 17 مسیر ممکن بررسی گردید؟ پاسخ قطعاً خیر است! اگر در مورد فرآیند جستجوی مسیر و محاسبه‌ی آن لحظه‌ای تامل کنید در می‌یابید که این فرآیند در یک مکان یعنی مغز شما با توجه به اطلاعات کامل از شبکه انجام شده است. نتیجه ی حاصل از این فرآیند را اصطلاحاً الگوریتم **مسیریابی متمرکز** می‌نامند. به عبارت دیگر یک روش در طبقه‌بندی الگوریتم‌های مسیریابی بر اساس این مبنا می‌باشد که آیا آن‌ها **فراگیر**[1] یا متمرکز هستند و یا آنکه غیر متمرکز و یا محلی می‌باشند.

- **الگوریتم مسیریابی فراگیر.** این الگوریتم مسیر با حداقل هزینه بین مبدا و مقصد را از طریق بهره‌گیری از یک دانش فراگیر و متمرکز از شبکه محاسبه می‌کند. با بهره‌گیری از ارتباط‌های بین همه‌ی گره‌ها و هزینه‌ی پیوندها به عنوان ورودی، این الگوریتم محاسبه‌ی مسیر بهینه را انجام می‌دهد. این به مفهوم آن است که الگوریتم به نوعی بایستی این اطلاعات را قبل از عملیات محاسبه جمع‌آوری نماید. عملیات مربوط به محاسبه می‌تواند در یک سمت یعنی یک الگوریتم مسیریابی فراگیر و متمرکز صورت پذیرد و یا آنکه به شکل تکرار شونده در چندین سمت صورت پذیرد. ویژگی کلیدی در آن است که یک الگوریتم فراگیر اطلاعات کامل از هزینه‌های ارتباطی و پیوندها را در اختیار دارد. در عمل الگوریتم‌های فراگیر را از لحاظ چگونگی جمع‌آوری اطلاعات، **الگوریتم‌های وضعیت پیوند (LS)**[2] می‌نامند زیرا الگوریتم بایستی از هزینه‌ی هر پیوند شبکه آگاه باشد. در بخش ۴-۱۵ این الگوریتم تحلیل می‌شود.
- **الگوریتم مسیریابی غیرمتمرکز.** محاسبه‌ی مسیر با حداقل هزینه در قالب عملیات تکرار شونده و توزیع شده صورت می‌پذیرد. هیچ یک از گره‌ها اطلاعات کامل از هزینه‌ی همه‌ی پیوندهای شبکه در اختیار ندارد. اما، عملیات خود را تنها با دانش از هزینه‌ی پیوندهای ارتباطی مستقیم از خود به دیگران آغاز می‌کند. بنابراین از طریق یک فرآیند تکرار محاسبه و مبادله‌ی اطلاعات با گره‌های مجاورش که در انتهای پیوند به آن‌ها متصل شده است به

1- Global
2- Link state algorithm

تدریج مسیر با حداقل هزینه را تا یک مقصد و یا مجموعه‌ای از مقصدها محاسبه می‌کند. الگوریتم غیرمتمرکز در بخش ۴-۱۶ تحلیل شده است. این الگوریتم را اصطلاحاً الگوریتم **بردار فاصله**[1] می‌نامند زیرا هر گره تخمینی از یک بردار را به دست می‌آورد که هزینه‌ها یا فاصله‌های آن را به دیگر گره‌های شبکه مشخص می‌کند.

روش دوم طبقه‌بندی الگوریتم‌های مسیریابی بر مبنای **ایستایی** و یا **پویایی** الگوریتم‌ها می‌باشد. در الگوریتم‌های **مسیریابی ایستا** تغییرات در مسیرها به کُندی در طول زمان انجام می‌شود و در اغلب موارد این تغییرات بر اساس دخالت انسان است. به عنوان مثال تغییر دستی جداول مسیریابی گسیل به جلو. از طرف دیگر در الگوریتم‌های **مسیریابی پویا** تغییرات مسیرهای شبکه بر مبنای بار ترافیک شبکه و یا تغییر سازه یا توپولوژی آن انجام می‌پذیرد. یک الگوریتم پویا می‌تواند اجرایی متناوب و یا در اثر واکنش مستقیم به تغییر سازه و یا هزینه‌ی پیوند را به همراه داشته باشد. اگر چه الگوریتم‌های پویا کُنش‌گرا نسبت به تغییرات یک شبکه هستند اما در برابر دشواری‌هایی از قبیل در حلقه قرار گرفتن عملیات مسیریابی و یا نوسان‌های حاصل در مسیرها حساس می‌باشند.

روش سوم طبقه‌بندی الگوریتم‌های مسیریابی بر مبنای آن است که آیا آن‌ها در برابر بار ترافیک شبکه حساس و یا غیرحساس می‌باشند؟ در یک **الگوریتم حساس در برابر بار** هزینه‌های پیوند به گونه‌ی پویا تغییر می‌کند تا منعکس کننده‌ی سطح تراکم جاری در پیوند مورد نظر باشد. اگر یک هزینه‌ی بالایی به یک پیوند جاری که متراکم است اختصاص یابد آنگاه الگوریتم مسیریابی تلاش در گزینش مسیرهایی در اطراف چنین پیوندهای متراکم می‌کند. در حال حاضر الگوریتم‌های مسیریابی مانند RIP ، OSPF ، BGP از جمله الگوریتم‌های **حساس در برابر بار** به شمار می‌آیند که شبکه‌ی اینترنت از آن‌ها بهره می‌گیرد. بدیهی است در این الگوریتم‌ها هزینه‌ی یک پیوند انعکاس انحصاری سطح تراکم جاری و یا گذشته‌ی آن نمی‌باشد.

۴-۱۵ الگوریتم مسیریابی وضعیت پیوند (LS)

در الگوریتم مسیریابی وضعیت پیوند توپولوژی شبکه و هزینه‌ی همه‌ی پیوندهای آن شناخته شده می‌باشند. به عبارت دیگر در دسترس الگوریتم LS به عنوان ورودی می‌باشند. در عمل دستیابی به این اطلاعات از طریق آگهی بسته‌های وضعیت پیوند از سوی هر گره به همه‌ی مسیریاب‌های شبکه انجام می‌شود. هر بسته‌ی وضعیت پیوند، حامل شناسه‌ها و هزینه‌ی پیوندهای مربوط به خود می‌باشد. به عنوان مثال پروتکل مسیریابی OSPF در اینترنت این عمل را از طریق الگوریتم **فراپخش وضعیت پیوند** انجام می‌دهد. نتیجه‌ی حاصل از این عمل آن است که همه‌ی گره‌ها دارای یک دیدگاه مشترک و کامل از شبکه می‌شوند. هر گره می‌تواند الگوریتم LS را اجرا و مجموعه‌ای از مسیرهای با حداقل هزینه را مانند هر گره‌ی دیگر محاسبه نماید.

الگوریتم مسیریابی وضعیت پیوند که در ذیل بررسی می‌شود الگوریتم **دایجسترا**[2] نامیده شده است که نام طراح آن است. الگوریتم دیگری که بسیار نزدیک به این الگوریتم است **پرایم** نامیده می‌شود. در هر حال الگوریتم دایجسترا مسیر با حداقل هزینه را از یک گره مبدا مانند U به همه‌ی گره‌های دیگر شبکه محاسبه می‌کند. این الگوریتم از روش تکرار محاسبه استفاده می‌کند و این خاصیت را دارد که پس

1- Distance vector
2- Dijkstra's algorithm

از تعداد k تکرار از الگوریتم، مسیرهای با حداقل هزینه برای تعداد k گره‌ی مقصد شناسایی می‌شوند. در میان مسیرهای با حداقل هزینه به همه‌ی گره‌های مقصد این k مسیر شامل تعداد k هزینه‌ی کمتر می‌باشند. در ذیل عبارت‌های به کار گرفته شده در این الگوریتم معرفی می‌شوند:

- **D (v)** . عبارت است از هزینه‌ی مسیر یا حداقل هزینه از گره‌ی مبدا به مقصد v با تعداد تکرار این نقطه از الگوریتم.
- **P (v)** . گره‌ی قبلی (مجاوری از v) در امتداد مسیر با حداقل هزینه‌ی جاری از مبدا v.
- N' . زیر مجموعه‌ای از گره‌های v در N' اگر مسیر با حداقل هزینه از مبدا به v کاملاً شناخته شده باشد.

الگوریتم مسیریابی فراگیر شامل یک گام مقدماتی است که در پی آن یک حلقه می‌باشد. تعداد مرتبه‌ای که بخش حلقه‌ی الگوریتم تکرار می‌شود برابر با تعداد گره‌های شبکه است. پس از خاتمه‌ی الگوریتم بایستی کوتاه‌ترین مسیرها از مبدا u را برای هر گره‌ی دیگر شبکه محاسبه نماید.

الگوریتم وضعیت پیوند (LS) برای گره‌ی مبدا:

```
1  Initialization:
2    N'= {u}
3    for all nodes v
4      if v is a neighbor of u
5        then D(v) = c (u,v)
       else D(v) = ∞
7
8 Loop
9 find w not in N' such that D(w) is a minimum
10 add w to N'
11 Update D(f) for each neighbor v of w and not in N':
12       D(v) = min D(v), D(w) + C(w,v)
13 /* new cost to v is either old cast to v orknown
14 least path cost to w plus cost from w to v * /
15 until N' = N
```

به عنوان مثال شبکه‌ی شکل ۴-۲۷ را در نظر بگیرید که محاسبه‌ی مسیرهای با حداقل هزینه از U به همه‌ی مقاصد ممکن مورد نظر است. یک خلاصه جدول از محاسبه‌ی انجام شده به وسیله‌ی الگوریتم در جدول ۴-۴ نشان داده شده است. در این جدول هر خط ارزش‌های متغیرهای الگوریتم را در انتهای تکرار نشان می‌دهد. نخستین گام‌های الگوریتم به گونه‌ی ذیل است:

- **مقداردهی اولیه.** مسیرهای جاری با حداقل هزینه از u به گره‌های مستقیم مجاور آن یعنی v ، x و w با ارزش‌های 2، 1 و 5 مقداردهی اولیه می‌شوند. بدیهی است هزینه به w برابر 5 تعریف می‌شود در حالیکه مسیر با هزینه‌ی کمتر نیز وجود دارد. زیرا این هزینه ی 5 متعلق به پیوند مستقیم از u به w است. هزینه‌های به y و g برابر ∞ می‌شود زیرا آن‌ها دارای ارتباط مستقیم با u نیستند.
- **تکرار اول.** در میان گره‌های اضافه نشده‌ی مجموعه‌ی N' جستجویی انجام می‌گیرد و گره‌ی با حداقل هزینه نسبت به آخرین مرتبه‌ی تکرار شناسایی می‌شود. این گره x است با هزینه‌ای برابر 1 و بنابراین x به مجموعه‌ی N' اضافه می‌شود. خط 12 از الگوریتم LS اجرا می‌شود و

تابع D (v) برای همه‌ی گره‌ها v به روزرسانی می‌شود. نتیجه‌ی این عملیات در گام اول جدول ۴-۴ نشان داده شده است. هزینه مسیر به v تغییری نکرده است. هزینه‌ی مسیر به w که قبلاً 5 در نظر گرفته شده بود با عبور از x برابر 4 می‌شود. بنابراین مسیر با حداقل هزینه گزینش می‌شود و گره‌ی پیش از w در امتداد کوتاه‌ترین مسیر از u به x تغییر می‌یابد. به گونه‌ی مشابه هزینه به y از طریق x برابر 2 محاسبه می‌شود و جدول بر اساس آن به روزرسانی می‌شود.

- **در تکرار دوم.** گره‌های v و y با حداقل هزینه ی مسیر برابر 2 شناسایی می شوند و وابستگی تصادفی شکسته می‌شود و گره‌ی y به مجموعه‌ی N' افزوده می‌شود تا N' شامل u ، x و y شود. هزینه‌ی گره‌های باقی‌مانده که در N' حضور ندارند مانند v ، w و g از طریق خط 12 الگوریتم LS به روزرسانی شده و نتیجه‌ی آن در گام دوم جدول ۴-۴ ذخیره می‌شوند.
- **تکرارهای بعدی . . .**

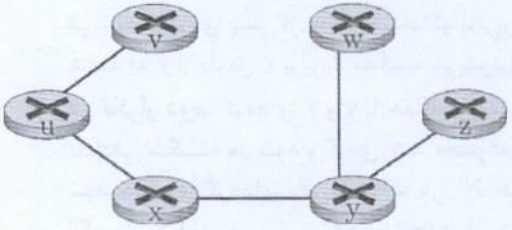
جدول ۴-۴ ♦ اجرای الگوریتم وضعیت پیوند بر روی شبکه‌ی شکل ۴-۲۷

گام	N'	D(v) و P(v)	D(w) و P(w)	D(x) و P(x)	D(y) و P(y)	D(z) و P(z)
۰	u	2 و u	5 و u	1 و u	∞	∞
۱	ux	2 و u	4 و x		2 و x	∞
۲	uxy	2 و u	3 و y			4 و y
۳	uxyv		3 و y			4 و y
۴	uxyvw					4 و y
۵	uxyvwz					

هنگامی که الگوریتم LS خاتمه می‌یابد، برای هر گره مسیر با حداقل هزینه از گره مبدا برای گره‌ی قبل آن بدست می‌آید. برای هر گره پیشین نیز بار دیگر قبل آن بدست می‌آید. این فرآیند ادامه پیدا می‌کند تا تمامی مسیر از گره‌ی مبدا به همه‌ی مقصدها حاصل شود. جدول مسیریابی به جلو در گره‌ای مانند u می‌تواند از طریق جمع‌آوری این اطلاعات و ذخیره‌ی آن برای هر مقصد و مسیریاب بعدی در امتداد کوتاه‌ترین مسیر از u به مقصد تشکیل گردد. شکل ۴-۲۸ مسیرهای با حداقل هزینه و جدول گسیل به جلو در u را ویژه‌ی شبکه‌ی ۴-۲۷ نشان می‌دهد.

اکنون این پرسش مطرح می‌شود که پیچیدگی محاسبه در این الگوریتم چیست؟ به عبارت دیگر با داشتن تعداد n مسیریاب بدون در نظر گرفتن گره‌ی مبدا چه میزان محاسبه بایستی در بدترین حالت ممکن جهت یافتن کوتاه‌ترین مسیر از مبدا به همه‌ی مقصدها انجام گیرد؟ در تکرار نخست عملیات جستجو تعداد n گره بایستی کاوش گردد تا مشخص گردد مسیریاب 5 که در مجموعه‌ی N' نیست دارای کمترین هزینه است. در تکرار دوم عملیات تعداد کاوش‌ها برابر n-1 و در تکرار سوم n-2 مسیریاب و غیره جهت یافتن کم‌ترین هزینه لازم می‌باشد. به طور کلی مجموع تعداد گره‌های لازم جهت جستجوی درون همه‌ی تکرارها برابر است با $\frac{n(n+1)}{2}$ و بنابراین می‌توان این نتیجه‌گیری را نمود که هر پیاده‌سازی از قبل الگوریتم LS دارای بدترین پیچیدگی در مقیاس مجذور n یعنی $O(n^2)$ می‌باشد. بدیهی است پیاده‌سازی این الگوریتم به صورت مجموعه‌ی مکان‌های ذخیره‌ی قابل بازگشت

(heap) در سطر 9 به گونه‌ی لگاریتمی و نه خطی حداقل هزینه را نتیجه می‌دهد که کاهش پیچیدگی الگوریتم را نیز به همراه دارد.



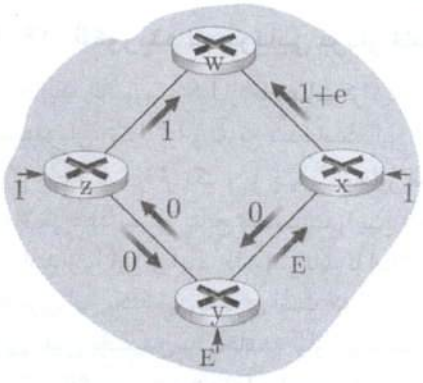
مقصد	پیوند
v	(u,v)
w	(u,x)
x	(u,x)
y	(u,x)
z	(u,x)

شکل ۴-۲۸ ♦ مسیرهای با حداقل هزینه و جدول گسیل به جلو برای گره‌ی u

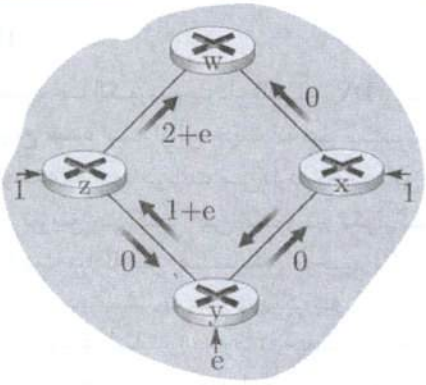
قبل از تکمیل بحث الگوریتم LS آسیب‌شناسی حاصل از این الگوریتم بررسی می‌شود. شکل ۴-۲۹ سازه‌ی یک شبکه‌ی ساده را نشان می‌دهد که هزینه‌های پیوند برابر است با بار حمل شده بر روی پیوند، مانند انعکاس تاخیر تجربه شده در عملیات. در این مثال هزینه‌های پیوند متقارن نمی‌باشند به عبارت دیگر، هزینه‌ی $c(u, v)$ تنها وقتی برابر $c(v, u)$ می‌شود که بار حمل شده در دو جهت پیوند (u, v) مشابه باشد. در این حالت گره‌ای مانند g سرچشمه‌ی یک جزء از ترافیک به مقصد w و گره‌ی x نیز سرچشمه‌ی یک جزء از ترافیک به مقصد w می‌باشد. به همین صورت مسیریاب y مقدار بار ترافیکی برابر e را به مقصد w تزریق می‌کند. شکل ۴-۲۹ (الف) مسیریابی اولیه به همراه هزینه‌های پیوند مربوط به مقدار ترافیک حمل شده را نشان می‌دهد.

هنگامی که الگوریتم LS اجرای بعدی را انجام می‌دهد، مسیریاب y هزینه‌ی مسیر به w را در جهت عقربه‌های ساعت بر مبنای هزینه‌های پیوند شکل ۴-۲۹ (الف) برابر **یک** محاسبه می‌کند در حالیکه مسیر جهت عکس عقربه‌ی ساعت به w که از آن استفاده می‌کرده است هزینه ی برابر 1+e را نتیجه داده است. بنابراین حداقل هزینه‌ی مسیر y به w در حال حاضر در جهت عقربه‌ی ساعت است. به شکل مشابه x نیز هزینه‌ی حداقل مسیر خود به e را به صورت جهت عقربه‌ی ساعت در نظر می‌گیرد و هزینه‌های نتیجه شده از این عمل مطابق شکل ۴-۲۹ (ب) می‌شود. هنگامی که الگوریتم LS اجرای بعدی را انجام می‌دهد، روترهای x و y و g همگی هزینه مسیر **صفر** را به w در جهت عکس عقربه‌های ساعت به دست می‌آورند و همگی ترافیک خود را به مسیرهای عکس عقربه‌ی ساعت جهت‌گیری می‌کنند. سرانجام در اجرای بعدی الگوریتم x و y و g همگی ترافیک خویش را به سوی مسیرهای در جهت عقربه‌ی ساعت هدایت می‌کنند.

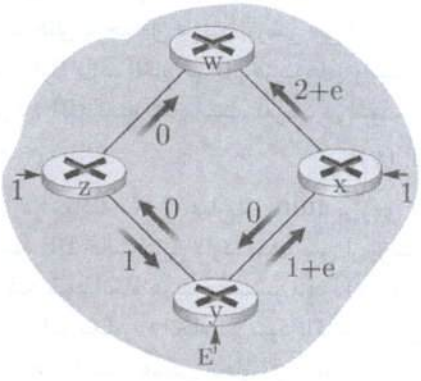
پرسش این است که چگونه می‌توان از این نوسان‌ها و بی‌ثباتی‌ها جلوگیری نمود، که در هر الگوریتمی و نه فقط در LS ممکن است روی دهد، آیا از یک مقیاس متراکم و یا پیوند تاخیردار بهره می‌گیرند؟ یک راه‌حل، اجبار این مورد است که هزینه‌های پیوند، وابستگی به مقدار ترافیک حمل شده نداشته باشند. بدیهی است این راه مطلقاً غیر قابل قبول است زیرا یکی از اهداف مسیریابی خودداری از بهره‌گیری از پیوندهای بسیار متراکم و در نتیجه با تاخیر بسیار طولانی است. راه‌حل دیگر حصول اطمینان از این موضوع است که همه‌ی مسیریاب‌ها همزمان الگوریتم LS را اجرا نکنند.



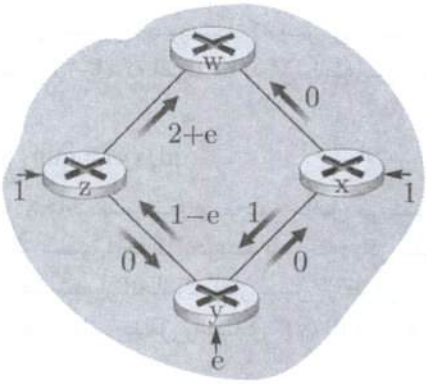
(الف) مسیریابی اولیه



(ب) x و y مسیر بهینه‌ای را به w در مسیر عقربه‌ی ساعت شناسایی می‌کند.



(ج) x، y، z مسیر بهینه‌ای به w در مسیر عکس عقربه‌ی ساعت شناسایی می‌کند.



(د) x، y، z مسیر بهینه‌ای به w در مسیر عقربه‌ی ساعت شناسایی می‌کند

شکل ۴-۲۹ ♦ مسیریابی حساس در برابر تراکم به همراه تغییرات ترافیک

این راه دوم به نظر مقبولیت بیشتری دارد زیرا این امید وجود دارد که حتی اگر مسیریاب‌ها الگوریتم LS را در زمان‌های با فاصله‌ی یکسان از یکدیگر اجرا نمایند آنگاه لحظه‌ی اجرای الگوریتم در هر مسیریاب مشابه نخواهد شد. پژوهش‌های انجام گرفته نشان می‌دهد که مسیریاب‌های اینترنت می‌توانند خود همزمانی در بین یکدیگر را انجام دهند به عبارت دیگر اگرچه آن‌ها در آغاز اجرا الگوریتم را در فواصل یکسان از یکدیگر انجام می‌دهند اما زمان اجرای هر یک متفاوت است. لحظه‌های اجرای الگوریتم می‌تواند سرانجام در مسیریاب‌ها همزمان شده و به همین حالت هم باقی بمانند. یک روش جهت جلوگیری از خود همزمانی آن است که هر روتر زمان ارسال آگهی خود به یک پیوند را تصادفی گزینش کند.

در ادامه مطالب فصل پس از بررسی الگوریتم LS انواع دیگر الگوریتم‌ها مانند مسیریابی بردار فاصله که در حال حاضر بسیار کاربرد دارند تحلیل می‌شوند.

۴-۱۶ الگوریتم مسیریابی بردار فاصله (DV)

در حالیکه الگوریتم LS از اطلاعات فراگیر بهره می‌گیرد اما الگوریتم بردار فاصله DV از نوع غیرهمزمان، تکرارگرا و توزیع شده است. این الگوریتم **توزیع شده** است از آن جهت که هر مسیریاب یا گره اطلاعات را از یک و یا چندین مجاور **مستقیماً پیوسته** به خود دریافت می‌دارد، محاسباتی را انجام می‌دهد سپس نتایج محاسبات را بین مجاوران خود توزیع می‌کند. این الگوریتم **تکرارگرا** می‌باشد زیرا این فرآیند ادامه می‌یابد تا اطلاعات دیگری جهت مبادله بین همسایگان موجود نباشد. یادآوری این نکته نیز ضروری است که این الگوریتم خود پایان‌پذیر است. به عبارت دیگر به سیگنالی برای پایان بخشیدن به محاسبه نیاز نمی‌باشد و الگوریتم خود عملیات را در وقت مناسب متوقف می‌کند. الگوریتم **غیرهمزمان** است از آن جهت که نیاز ندارد همه‌ی مسیریاب‌ها با یکدیگر همگام عمل نمایند. در آینده ملاحظه خواهد شد که یک الگوریتم غیرهمزمان، تکرارگرا، خود پایان‌پذیر و توزیع شونده بسیار جالب‌تر از نوع متمرکز خود می‌باشد!

قبل از تحلیل الگوریتم DV سودمند خواهد بود که رابطه‌ی مهم حاضر بین هزینه‌های وابسته به مسیرهای با حداقل هزینه بررسی می‌گردد. اگر $d_x(y)$ هزینه‌ی مسیری تصور شود که بین روترهای x به y قرار دارد. آنگاه حداقل هزینه‌های وابسته از طریق معادله (۴-۴) که اصطلاحاً به آن رابطه بل مَن فورد (BF) گفته شده است، تعریف می‌شود.

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\} \qquad (۴-۴)$$

در معادله (۴-۴) عبارت $\min_v$ بر روی تمامی مقادیر x ها از مسیریاب‌های مجاور حاصل می‌شود. رابطه BF حالت ذاتی دارد. یقیناً پس از پویش از x به v اگر مسیر با حداقل هزینه از v به y در نظر گرفته شود آنگاه هزینه‌ی مسیر برابر $c(x,v) + d_v(y)$ خواهد شد. از آنجائیکه آغاز پویش از طریق یک مسیریاب مجاور v انجام می‌شود ناگزیر حداقل هزینه از x به y برابر حداقل رابطه‌ی: $c(x,v) + d_v(y)$ با درنظر گرفتن همگی مجاوران v می‌باشد.

اما بر آن گروه از مردمان که در مورد قوت اعتبار معادله چندان خوش‌بین نیستند تحلیل برای گره‌های مبدا u و مقصد z در شکل ۴-۲۷ جهت صحت این رابطه انجام می‌شود. گره‌ی مبدا u دارای 3 مجاور v ، x و w می‌باشد. از طریق پویش در امتداد مسیرهای گوناگون در گراف به سادگی می‌توان ملاحظه نمود که $d_v(z) = 5$ ، $d_x(z) = 3$ و $d_w(z) = 3$ می‌باشد. با جایگذاری این ارزش‌ها در معادله‌ی (۴-۴) به همراه هزینه‌های $c(u,v) = 2$ ، $c(u,x) = 1$ و $c(u,w) = 5$ آنگاه:

$$d_u(z) = \min\{2+5, 5+3, 1+3\} = 4$$

بدیهی است که صحت رابطه اثبات می‌گردد و دقیقاً همان چیزی است که الگوریتم دایجسترا برای همان شبکه به ما عرضه داشت. این اعتبارسنجی بایستی برای نگرانی‌های این گروه از افراد پاسخ لازم را فراهم آورده باشد.

معادله‌ی BF تنها یک رابطه‌ی نظری نیست بلکه در عمل نیز اهمیت ویژه‌ی خود را دارد. به ویژه آنکه راه‌حل معادله‌ی BF ورودی‌هایی را جهت جدول گسیل به جلو برای روتر x فراهم می‌آورد. جهت ملاحظه‌ی این مهم اگر v^* هر گره‌ی مجاوری در نظر گرفته شود که نتیجه‌ی حاصل حداقل را در معادله‌ی (۴-۴) سبب شود آنگاه اگر گره‌ی x مایل به ارسال یک بسته به روتر y در مسیر با حداقل هزینه‌ای باشد. آن گره ابتدا بایستی بسته را به گره‌ی v^* گسیل نماید بنابراین جدول گسیل به جلو

برای x گره‌ی v^* را به عنوان پرش کوتاه بعدی به روتر مجاور جهت رسیدن به مقصد نهایی y تعیین می‌کند. یک کاربرد مهم‌تر دیگر در معادله‌ی BF ارتباط مجاور به مجاور است که در الگوریتم DV رخ می‌دهد.

فلسفه‌ی اصلی این گونه است که هر گره‌ی x آغازی به صورت $D_x(y)$ دارد که همراه آن یک تخمین از هزینه‌ی مسیر با حداقل هزینه از x به گره‌ای مانند y برای همه تعداد گره‌ها می‌باشد. آنگاه بردار فاصله‌ی گره‌ی x در قالب رابطه‌ی (۴-۵) تعریف می‌شود.

$$D_x = [D_x(y) : y \quad in \quad N] \qquad (۵-۴)$$

این بردار هزینه‌ی پیش‌بینی شده از x به همه‌ی گره‌های دیگری است مانند y در N . با بهره‌گیری از الگوریتم DV هر گره‌ی x اطلاعات مسیریابی ذیل را نگهداری می‌کند:

- برای هر مجاور v هزینه‌ی $c(x,v)$ از x به مجاور با ارتباط مستقیم v .
- بردار فاصله‌ی گره‌ی x یعنی $D_x = [D_x(y) : y \text{ in } N]$ که حامل تخمین هزینه‌ی x به همگی مقصدهای y در N می‌باشد.
- بردارهای فاصله هر یک از مجاوران به عبارت دیگر $D_v = [D_v(y) : y \text{ in } N]$ برای هر مجاور v از x .

در الگوریتم غیرهمزمان، توزیع شده در فواصل زمانی، هر گره نسخه‌ای از بردار فاصله‌ی خود را به هر یک از مجاورانش ارسال می‌دارد. هنگامی که یک گره‌ی x بردار فاصله‌ی جدیدی را از هر یک از مجاورانش مانند v دریافت نماید آنگاه به ذخیره‌ی بردار فاصله‌ی v پرداخته و سپس با بهره‌گیری از معادله‌ی BF فاصله بردار خود را به گونه‌ی رابطه‌ی (۴-۶) برای هر گره‌ی y در N به روزرسانی می‌کند.

$$D_x(y) = \min_v \{c(x,v) + D_v(y)\} \qquad (۶-۴)$$

اگر بردار فاصله‌ی گره‌ی x در اثر این عملیات به روزرسانی تغییر نماید آنگاه x بردار فاصله‌ی به روزرسانی شده‌ی خود را به هر یک از مجاورانش گسیل می‌دارد و آن‌ها نیز بردار فاصله‌ی خود را بر این اساس به روزرسانی می‌کنند. در اثر ادامه‌ی این روش غیرهمزمان توزیع بردارهای فاصله بین همه‌ی گره‌ها هر هزینه‌ی تخمینی $D_x(y)$ گرایش به سوی $d_x(y)$ می‌نماید که هزینه‌های واقعی مسیر با حداقل هزینه است از روتر x به مسیریاب y . **الگوریتم بردار فاصله (DV)** در هر گره‌ی x به صورت ذیل قابل محاسبه می‌باشد:

```
1  Initialization
2     for all destinations y in N:
3        Dx(y)= c(x,y)  /* if y is not a neighbor then c(x,y) =∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor
7        send distance vector Dx=[Dx(y): y in N] to w
8
9  loop
10 wait (until I see a link cost change to some neighbor w or
11       until I receive a distance vector from some neighbor w)
12
```

```
13 for each y in N:
14     Dx(y) = minv {c(x,v) + D (y)}
15
16  if Dx(y) changed for any destination y
17     send distance vector Dx =[Dx(y): y in N] to all neighbors
18
19 forever
```

در الگوریتم DV یک گره‌ی x تخمین حاصل از بردار فاصله‌ی خود را هنگامی به روزرسانی می‌کند که یک تغییر هزینه را در یکی از پیوندهای اتصال مستقیم خود مشاهده نماید و یا آنکه به روزرسانی بردار فاصله را از برخی مجاوران خود دریافت نماید. اما برای به روزرسانی جدول گسیل به جلوی خود برای یک مقصد y آنچه x نیاز دارد آگاهی از کوتاه‌ترین مسیر به y نمی‌باشد بلکه نیاز به اگاهی از گره‌ی مجاور $v^*(y)$ دارد که پرش کوتاه به مسیریاب بعدی در امتداد این کوتاه‌ترین مسیر به y می‌باشد. چنانچه انتظار می‌رود $v^*(y)$ روتر مجاور v است که مقدار حداقل را در سطر 14 الگوریتم DV اختیار می‌کند. اگر چندین مجاور v وجود داشته باشند که قادر به تامین حداقل مقدار برای خود باشند آنگاه $v^*(y)$ می‌تواند هر یک از مجاوران حداقل کننده باشد. بنابراین در سطرهای 13 و 14 برای هر مقصد y گره‌ی x مقدار $v^*(y)$ را محاسبه می‌کند و آن را در جدول گسیل به جلو برای مقصد y به روزرسانی می‌کند.

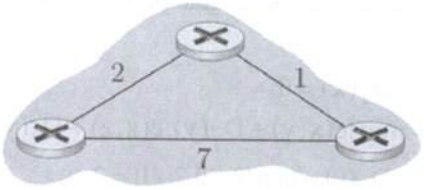
به یادآورید که الگوریتم LS فراگیر است از آن جهت هر مسیریاب را مجبور به جمع‌آوری اطلاعات کامل از شبکه قبل از اجرای الگوریتم دایجسترا می‌کند. الگوریتم DV غیر متمرکز است در برابر فراگیری LS و بنابراین هر یک از روترها اطلاعات جامع از تمامی شبکه را در اختیار ندارند. تنها اطلاعاتی که یک مسیریاب در اختیار دارد هزینه‌ی پیوندهایی است که دارای ارتباط مستقیم با مجاوران آن هستند و نیز اطلاعاتی که از این همسایگان دریافت می‌کند. هر مسیریاب به انتظار اطلاعات به روزرسانی شده از سوی همسایه‌ی خود بر مبنای سطرهای 10 و 11 الگوریتم است. محاسبه‌ی بردار فاصله‌ی جدید بر مبنای اطلاعات به روزرسانی شده در سطر 14 انجام می‌گیرد و توزیع اطلاعات محاسبه شده‌ی جدید به همسایگان یک گره از طریق سطرهای 16 و 17 الگوریتم صورت می‌پذیرد. الگوریتم‌های مشابه DV در بسیاری از پروتکل‌های مسیریابی بهره‌برداری می‌شوند. از جمله پروتکل‌های مسیریابی که از الگوریتم DV بهره می‌گیرند عبارتند از RIP ، BGP ، ISOIDRP ، Novel IPX و چندین پروتکل دیگر.

شکل ۴-۳۰ عملیات الگوریتم DV را برای یک شبکه‌ی ساده که از تعداد 3 مسیریاب تشکیل شده است نشان می‌دهد. عملیات الگوریتم در قالب یک روش همزمان شده عمل می‌کند که همگی گره‌ها بردارهای فاصله را از همسایگان خویش همزمان دریافت می‌دارند و بردارهای فاصله‌ی جدید را بر مبنای آن محاسبه و تغییرات احتمالی را به مجاوران خود آگهی می‌دهند. پس از مطالعه‌ی این مثال با اندکی تامل می‌توانید خود را متقاعد نمائید که الگوریتم به گونه‌ی غیرهمزمان نیز می‌تواند عملکرد درستی را داشته باشد. در چنین حالتی محاسبه و به روزرسانی تولید و دریافت اطلاعات بردارهای فاصله در هر زمان می‌تواند روی دهد.

ستون سمت چپ شکل ۴-۳۰ سه **جدول مسیریابی** اولیه برای هر یک از سه مسیریاب را نشان می‌دهد. به عنوان مثال جدول گوشه‌ی بالای سمت چپ در شکل جدول مسیریابی اولیه‌ی گره‌ی x است. در بین هر جدول مسیریابی ویژه‌ی هر ردیف یک بردار فاصله است و هر جدول مسیریابی گره

شامل بردار فاصله‌ی خود و هر یک از همسایگان خویش است. بنابراین ردیف نخست در جدول اولیه‌ی مسیریابی گره‌ی x به صورت رابطه‌ی (۴-۷) است.

$$D_x = [D_x(x), D_x(y), D_x(z)] = [0, 2, 7] \qquad (۴-۷)$$



جدول روتر x

هزینه به	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

هزینه به	x	y	z
x	0	2	5
y	2	0	7
z	7	1	0

هزینه به	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

جدول روتر y

هزینه به	x	y	z
x	∞	∞	∞
y	0	2	1
z	∞	∞	∞

هزینه به	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

هزینه به	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

جدول روتر z

هزینه به	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

هزینه به	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

هزینه به	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

شکل ۴-۳۰ ♦ الگوریتم بردار فاصله (DV)

ردیف‌های دوم و سوم در این جدول جدیدترین بردارهای فاصله‌ی دریافتی به ترتیب از گره‌های y و z می‌باشند. از آنجائیکه در مقداردهی اولیه مسیریاب x اطلاعاتی را از y و یا z دریافت نکرده است دو ردیف دوم و سوم با ارزش بی‌نهایت (∞) مقداردهی اولیه شده‌اند.

پس از عملیات مقداردهی اولیه هر گره اطلاعات بردار فاصله‌ی خود را به هر یک از دو همسایه‌ی

خویش گسیل می‌دارد. این موضوع در شکل ۴-۳۰ از طریق پیکان‌هایی از ستون اول به ستون دوم جداول مشخص شده‌اند. به عنوان مثال روتر x بردار فاصله‌ی $D_x = [0,2,7]$ را به هر دو مسیریاب y و z ارسال می‌کند. پس از دریافت این اطلاعات به روزرسانی شده هر مسیریاب بردار فاصله‌ی خویش را مجدداً محاسبه می‌کند. روابط (۴-۸) و (۴-۹) عملیات محاسبه‌ی مجدد بردار فاصله را برای مسیریاب x نشان می‌دهند.

$$D_x(x) = 0$$
$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \qquad (۸-۴)$$
$$D_x(y) = \min\{2+0, 7+1\} = 2$$
$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \qquad (۹-۴)$$
$$D_x(z) = \{2+1, 7+0\} = 3$$

ستون دوم بنابراین برای هر گره بردار فاصله‌ی جدید آن را در امتداد بردارهای فاصله‌ی دریافتی در آن لحظه از همسایگان خود به نمایش می‌گذارد. بدیهی است که مسیریاب x حداقل هزینه به z را پیش‌بینی می‌کند. مقدار $D_x(z)$ به عنوان مثال از 7 به 3 تغییر کرده است. همچنین برای گره‌ی x مسیریاب مجاور یعنی y دست آورد حداقل را در سطر 14 الگوریتم DV را به همراه دارد. در این مرحله از الگوریتم در مسیریاب x به ترتیب روابط $v^*(y) = y$ و $v^*(z) = y$ برقرار است.

پس از آنکه گره‌ها بردار فاصله‌ی خویش را محاسبه مجدد نمودند آن‌ها بار دیگر اطلاعات به روزرسانی شده‌ی بردارهای فاصله‌ی خود را به همسایگان در صورت یک تغییر ارسال می‌دارند. این مورد در شکل ۴-۳۰ از طریق پیکان‌هایی از ستون دوم جدول به ستون سوم همان جدول نشان داده شده است. بدیهی است تنها گره‌های x و z اطلاعات به روزرسانی شده‌ی خویش را ارسال می‌دارند. بردار فاصله‌ی گره‌ی y تغییری نکرده است و بنابراین اطلاعات جدیدی نیز گسیل نمی‌کند. پس از دریافت اطلاعات جدید مسیریاب‌ها بردارهای فاصله‌ی خویش را محاسبه‌ی مجدد نموده و جداول مسیریابی خود را به روزرسانی می‌کنند و این تغییرات در ستون سوم شکل ۴-۳۰ نشان داده شده است.

فرآیند دریافت بردارهای فاصله‌ی به روزرسانی شده از همسایگان، محاسبه‌ی مجدد اطلاعات جداول مسیریابی و آگاه نمودن همسایگان از تغییرات حاصل در هزینه‌ها به یک مقصد از طریق مسیر با حداقل هزینه تا آنجا ادامه می‌یابد که دیگر هیچ‌گونه پیام به روزرسانی شده‌ای ارسال نگردد. در این نقطه از آنجائیکه پیام‌های به روزرسانی شده‌ای ارسال نمی‌گردد محاسبات جدول مسیریابی دیگری نیز صورت نمی‌پذیرد و الگوریتم وارد مرحله توقف می‌شود به عبارت دیگر همه ی گره‌ها وارد مرحله‌ی انتظار بر مبنای سطرهای 10 و 11 الگوریتم DV می‌شوند. الگوریتم DV تا زمانی به این مرحله‌ی ایستایی خود ادامه می‌دهد که تغییرات هزینه‌ی یک پیوند گزارش و یا آن پیوند دچار عیب شود.

هنگامی که یک گره الگوریتم DV را اجرا می‌کند تغییر ایجاد شده در هزینه‌ی پیوند از خویش به همسایه‌اش را بر مبنای سطرهای 10 و 11 الگوریتم تشخیص داده و بردار فاصله‌ی خود را بر اساس سطرهای 13 و 14 به روزرسانی می‌کند و اگر تغییری در هزینه‌ی مسیر روی داده باشد همسایگان خود را بر اساس سطرهای 16 و 17 از تغییرات جدید آگاه می‌کند. شکل ۴-۳۱ (الف) حالتی را به تصویر کشیده است که هزینه‌ی پیوند از y به x تغییر از 4 به 1 را داشته است. در این بحث تمرکز فقط بر

روی جداول فاصله‌ی دو مسیریاب y و z به مقصد آن‌ها یعنی x می‌باشد. الگوریتم DV باعث انجام رویدادهای ذیل خواهد شد:

- **در زمان** t_0، گره y تغییر هزینه‌ی پیوند را که از 4 به 1 می‌باشد ملاحظه می‌کند، بردار فاصله‌ی خود را به روزرسانی می‌کند و مجاورانش را از این تغییرات آگاه می‌کند زیرا بردار فاصله‌اش تغییر کرده است.
- **در زمان** t_1، مسیریاب z اطلاعات جدید را از y دریافت و جدول خود را به روزرسانی می‌کند. این گره هزینه‌ی حداقل را به x که کاهش از هزینه ی 5 به 2 می‌باشد محاسبه و بردار فاصه‌ی جدید خود را به مجاورانش گسیل می‌کند.
- **در زمان** t_2، گره‌ی y اطلاعات به روزرسانی شده‌ی z را دریافت و جدول فاصله‌ی خود را بر این اساس تغییر می‌دهد. هزینه‌های حداقل y تغییری نیافته‌اند و بنابراین y هیچگونه پیامی را به z ارسال نمی‌کند. الگوریتم DV در این حالت به وضعیت انتظار می‌رود.

بنابراین فقط دوبار تکرار عملیات برای آنکه الگوریتم DV در وضعیت ایستا قرار گیرد لازم می‌باشد. نکته‌ی مهم آن است که هزینه‌ی کاهش یافته بین x و y به سرعت در سراسر شبکه انتشار می‌یابد.

اکنون حالتی را در نظر بگیرید که هزینه ی پیوند دچار افزایش می‌شود. فرض کنید بر اساس آنچه در شکل ۴-۳۱ (ب) نشان داده شده است هزینه ی پیوند بین x و y از مقدار 4 به 60 افزایش یابد.

۱- قبل از تغییرات هزینه‌ی پیوند، $D_y(x)=4$ ، $D_y(z)=1$ ، $D_z(y)=1$ و $D_z(x)=5$ در زمان t_0 گره ی y تغییر هزینه‌ی پیوند را ملاحظه و افزایش آن را از 4 به 60 شناسایی می‌کند. گره‌ی y مسیر با حداقل هزینه به x بر اساس رابطه‌ی (۴-۱۰) محاسبه می‌کند.

$$D_y(x)=\min\{c(y,x)+D_x(x),c(y,z)+D_z(x)\} \qquad (۴-۱۰)$$

$$D_y(x)=\min\{60+0,1+5\}=6$$

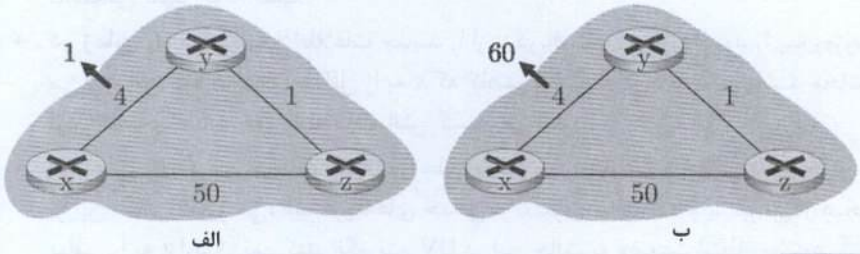
بدیهی است با توجه به نگاه فراگیر از شبکه ملاحظه می‌شود که این هزینه‌ی جدید از طریق z اشتباه است. اما تنها اطلاعاتی که y در اختیار دارد هزینه‌ی مستقیم آن به x است که برابر 60 می‌باشد و اطلاعات z که آخرین مرتبه به y گفته است می‌تواند ارتباط با x را با هزینه‌ی 5 تامین نماید. بنابراین برای رسیدن به x گره‌ی y بایستی از z عبور نماید با این توقع که z قادر است وی را به x با هزینه‌ی 5 ارتباط دهد. اما مانند زمان t_1 یک مسیریابی حلقه مانند حاصل شده و برای رسیدن به x مسیریاب y از z عبور می‌کند و سپس از z بار دیگر به y یک مسیریابی حلقوی مانند یک سیاه‌چاله در فضا عمل می‌کند یعنی بسته ای به مقصد x هنگام ورود به y و یا به z در زمان t_1 بین این دو مسیریاب برای همیشه به عقب و جلو رانده می‌شود و تا زمان اعمال تغییراتی در جداول گسیل به جلو این وضعیت ادامه می‌یابد.

۱- از آنجائیکه گره‌ی y هزینه‌ی حداقل جدید به x را محاسبه کرده است به اگاهی z می‌رساند که بردار فاصله‌ی جدید آن در زمان t_1 چیست.

۲- مدت زمانی که پس از t_1 گره ی z به دریافت بردار فاصله‌ی جدید y می‌پردازد که بیان کننده‌ی این واقعیت است که هزینه‌ی حداقل y برابر 6 می‌باشد. از آنجائیکه z می‌داند با هزینه‌ی 1 می‌تواند به y برسد بنابراین هزینه‌ی حداقل جدید به x را به گونه‌ی: $D_z(x)=\min\{50+0,1+6\}=7$ محاسبه می‌کند. از آنجائیکه هزینه ی حداقل z به x

افزایش یافته است آنگاه گره‌ی y را در زمان t_1 از مقدار جدید بردار فاصله آگاه می‌کند.

۳- در یک روش مشابه گره‌ی y پس از دریافت بردار فاصله‌ی جدید z محاسبه‌ی $x_y(x)=8$ را انجام می‌دهد و بردار فاصله‌اش را به z می‌فرستد. گره z آنگاه $D_z(x)=9$ را محاسبه و بردار فاصله‌ی جدید خود را به y گسیل می‌دارد و این عملیات تکرار می‌شود.



الف ب

شکل ۴-۳۱ ♦ تغییرات در هزینه‌ی پیوند

چه مدت زمان این فرآیند به طول می‌انجامد؟ پاسخ آن است که با اندکی دقت می‌توان دریافت که تبادل پیام بین y و z به صورت حلقه 44 مرتبه تکرار می‌شود تا آنکه z سرانجام هزینه‌ی مسیر خود را از طریق u محاسبه نماید و آگاه شود که مقدار آن از 50 بزرگتر است. در این نقطه z سرانجام در می‌یابد که مسیر با حداقل هزینه‌اش به x از طریق ارتباط مستقیم با x امکان‌پذیر می‌شود. در این حالت است که مسیریاب y راه خود را به x از طریق z می‌گشاید. نتیجه‌ی خبر ناخوشایند افزایش هزینه‌ی پیوند به کُندی منتشر شده است! چه اتفاقی روی می‌دهد اگر هزینه‌ی پیوند $c(y,x)$ از 4 به 10000 تغییر کند و هزینه‌ی $c(z,x)$ برابر 9999 باشد؟ به دلیل وقوع چنین حالت‌هایی، دشواری روی داده اصطلاحاً مسئله‌ی **شمارش به بی‌نهایت**[1] نام گرفته است.

از وقوع مسیریابی حلقوی که در بالا به آن اشاره رفت می‌توان با بهره‌گیری از روشی که به **مسمومیت معکوس**[2] شهرت یافته است بهره گرفت. فلسفه‌ی این روش آن است که اگر z برای رسیدن به x از طریق y اقدام نماید آنگاه z به y آگهی می‌دهد که فاصله‌اش به x بی‌نهایت است. به عبارت دیگر z به y آگاهی می‌دهد که $D_z(x)=\infty$ است اگرچه z در واقع می‌داند که $D_z(x)=5$ می‌باشد. مسیریاب z تکرار این عدم واقعیت را به y تا زمانی ادامه می‌دهد که مسیردهی خود به x را از طریق y انجام می‌دهد. از آنجائیکه y معتقد است z مسیری به x ندارد آنگاه y هرگز تلاش در مسیردهی به x از طریق z را نمی‌کند. بدیهی است تا مادامی که مسیردهی به x از طریق y انجام می‌شود.

اکنون جهت آگاهی از این موضوع که روش مسمومیت معکوس چگونه این شکل مسیریابی حلقوی را برطرف می‌سازد شکل ۴-۳۱ (ب) را مجدداً در نظر می‌گیریم. در نتیجه‌ی روش مسمومیت معکوس جدول فاصله‌های y به صورت $D_z(x)=\infty$ در می‌آید. هنگامی که هزینه‌ی تغییر پیوند $c(x,y)$ از 4 به 60 در زمان t_0 روی دهد، مسیریاب y جدول خود را به روزرسانی نموده و ادامه‌ی مسیردهی را مستقیماً به x انجام می‌دهد حتی در یک هزینه‌ی بالاتر از 60 و z را از هزینه‌ی جدید خود به x آگاه می‌کند. به عبارت دیگر $D_y(x)=60$ می‌باشد. پس از دریافت اطلاعات به روزرسانی شده در t_1 گره‌ی z بی‌درنگ مسیر خود را به x از طریق پیوند مستقیم (z,x) با هزینه‌ی 50 تغییر می‌دهد. از آنجائیکه

1- Count to infinity
2- Poisoned reverse

این مسیر با حداقل هزینه به x است و از آنجائیکه مسیر دیگری از درون y عبور نمی‌کند آنگاه گره‌ی z در زمان t_2 به y اطلاع می‌دهد که $D_z(x)=50$ است. پس از دریافت اطلاعات جدید از z مسیریاب y جدول فاصله‌ی خود را با $D_y(x)=51$ به روزرسانی می‌کند و نیز گره‌ی z بر روی مسیر حداقل هزینه‌ی y به گره x قرار می‌گیرد. در این حالت است که y مسیر معکوس از z به x را با آگاهی دادن z در زمان t_3 با $D_y(x)=\infty$ مسموم می‌کند با آن که y می‌داند که $D_y(x)=51$ می‌باشند.

پرسش این است که آیا راه‌حل مسمومیت معکوس شکل شمارش به بی‌نهایت را برطرف می‌سازد؟ پاسخ خیر است زیرا حلقه‌های با تعداد گره‌ی مجاور 3 و یا بیشتر از سوی روش مسمومیت معکوس قابل تشخیص نمی‌باشد.

۱۷-۴ مقایسه‌ی دو الگوریتم مسیریابی DV و LS

از لحاظ محاسبه‌ی مسیریابی الگوریتم‌های LS و DV رویکردهای مکمل یکدیگر را دارند. در الگوریتم DV هر گره فقط با همسایه‌ی مستقیم خود تعامل می‌کند اما در عین حال پیش‌بینی‌های حداقل هزینه از خود به همگی گره‌های شناخته شده‌ی خود را در شبکه در اختیار همسایگان قرار می‌دهد. در الگوریتم LS هر گره با همگی گره‌های دیگر شبکه از طریق راه‌کار فراپخش تعامل می‌کند اما فقط هزینه‌های پیوندهای ارتباطی مستقیم خود را به آن‌ها اطلاع می‌دهد. با توجه به اینکه N و E به ترتیب مجموعه‌ی گره‌ها و یال‌ها می‌باشند مقایسه‌ی ذیل را می‌توان برای دو الگوریتم برشمرد.

۱- **پیچیدگی پیام.** ملاحظه شد در الگوریتم LS هر مسیریاب از هزینه‌ی هر پیوند شبکه آگاه است. این به مفهوم آن است که تعداد $O(|N|\ |E|)$ پیام بایستی گسیل شود. همچنین هرگاه هزینه‌ی یک پیوند تغییر نماید، هزینه جدید پیوند بایستی به همه‌ی گره‌ها گسیل گردد. الگوریتم DV تبادل پیام همسایگان مستقیماً مرتبط با یکدیگر را برای هر تکرار عملیات خواهان است. قبلاً مشاهده شد که زمان لازم برای آنکه الگوریتم همگرا شود بستگی به عوامل زیادی دارد. هنگامی که هزینه‌های پیوند تغییر می‌کند، الگوریتم DV نتایج حاصل از تغییرات را تنها اگر هزینه‌های جدید سبب تغییری در مسیر حداقل هزینه‌ی یکی از گره‌های مرتبط با آن پیوند گردد انتشار می‌دهد.

۲- **سرعت همگرایی.** ملاحظه شد که پیاده‌سازی الگوریتم LS یک رویکرد $O(|N^2|)$ است و به تعداد $O(|N|\ |E|)$ پیام نیازمند است. الگوریتم DV به کندی همگرا می‌شود و می‌تواند سبب مسیریابی حلقوی به هنگام همگرا شدن گردد. الگوریتم DV از طرف دیگر از مشکل شمارش به سمت بی‌نهایت رنج می‌برد.

۳- **قدرتمندی.** چه حادثه‌ای روی می‌دهد اگر مسیریابی دچار عیب، سوء رفتار و یا تخریب گردد؟ در شرایط الگوریتم LS مسیریاب می‌تواند یک هزینه‌ی نادرست برای یکی از پیوندهای متصل به خود را فراپخش نماید. بدیهی است در LS ارسال پیام نادرست برای دیگر پیوندها روی نمی‌دهد. در نتیجه یک الگوریتم می‌تواند هر بسته ی دریافتی را به عنوان بخشی از یک فراپخش LS حذف و یا آلوده نماید. اما یک گره‌ی LS فقط به محاسبه‌ی جدول مسریابی خود می‌پردازد و دیگر مسیریاب‌ها محاسبات مشابه را برای خود انجام می‌دهند. این به مفهوم آن است که محاسبه‌ی مسیر به نوعی در شرایط LS مجزا انجام می‌شود و در نتیجه سبب اندازه‌ای از قدرتمندی است. در شرایط الگوریتم DV یک گره می‌تواند مسیرهای با حداقل هزینه‌ی نادرست را به هر یک و یا تمامی مقصدها آگهی نماید. در سال 1997 میلادی یک مسیریاب

معیوب در یک ISP کوچک اطلاعات مسیریابی نادرست را برای روترهای زیرساخت کشوری آمریکا فراهم نمود. این عیب سبب شد دیگر مسیریاب‌ها این روتر معیوب را با حجم ترافیک خود سیلابی نمایند و در نتیجه بخش وسیعی از شبکه‌ی اینترنت برای مدت چند ساعت از کار افتاد. به طور کلی در هر تکرار عملیات، یک محاسبه در الگوریتم DV از سوی یک گره به مجاور آن منتقل می‌شود و سپس غیرمستقیم به همسایه مجاورش در تکرار بعدی و این حالت ممکن است گسترش یابد. به عبارت دیگر در الگوریتم DV یک محاسبه‌ی نادرست در یک گره می‌تواند تمامی شبکه را تحت تاثیر خود قرار دهد.

در انتها می‌توان این نتیجه‌گیری را نمود که هیچ یک از الگوریتم‌ها بر دیگری برتری ندارد و به همین دلیل از هر دو الگوریتم در شبکه‌ی اینترنت استفاده می‌شود. دو الگوریتم LS و DV نه تنها به شکل گسترده‌ای در عمل مورد استفاده قرار می‌گیرند بلکه تنها دو الگوریتمی هستند که در حال حاضر اینترنت از آن‌ها بهره می‌گیرد. الگوریتم‌های بی‌شمار دیگری نیز در طول 30 سال گذشته از سوی پژوهشگران پیشنهاد شده‌اند. از جمله‌ی این الگوریتم‌ها که سرچشمه‌ی آن از دنیای ارتباطات تلفنی است **الگوریتم‌های مسیریابی راهیابی شده**[1]! می‌باشند که در شبکه‌های راهیابی بسته مورد بهره‌برداری قرار می‌گیرند. بدیهی است این رویکرد هنگامی مورد استفاده قرار می‌گیرد که منابع هر پیوند مانند بافرها و یا بخشی از پهنای باند برای هر ارتباط مسیریابی شده‌ی پیوند بایستی حالت ذخیره را داشته باشند.

۴-۱۸ مسیریابی سلسله مراتبی

در مطالعه‌ی الگوریتم‌های LS و DV شبکه در قالب مجموعه‌ای از مسیریاب‌های متصل به یکدیگر در نظر گرفته شد. ممیزی یک مسیریاب از دیگری از لحاظ اینکه همه‌ی روترها الگوریتم مسیریابی مشابهی را جهت محاسبه‌ی مسیرهای مطلوب از درون تمامیت شبکه به کار می‌گرفتند امکان‌پذیر نمی‌نمود. در عمل این الگو و نگاه آن به مجموعه‌ای از مسیریاب‌های همگن که همگی الگوریتم مسیریابی مشابهی را به اجرا در می‌آورند به دو دلیل مهم ذیل ساده به نظر می‌رسد:

۱- **مقیاس.** هر چه بر تعداد مسیریاب‌ها افزوده شود، سرآیند حاصل از محاسبه، پردازش، ذخیره‌سازی و اطلاعات مسیریابی مانند به روزرسانی‌های LS و یا تغییرات مسیر با هزینه‌ی حداقل نیز در مقیاس غیر قابل قبولی افزایش می‌یابد. امروزه شبکه‌ی اینترنت شامل صدها میلیون میزبان است. ذخیره‌سازی اطلاعات مسیریابی در هر یک از این میزبان‌ها نیازمند حجم وسیع حافظه است. سرآیند لازم جهت آگهی به روزرسانی‌های LS در بین همه‌ی مسیریاب‌ها در شبکه‌ی اینترنت عمومی پهنای باند کافی را جهت ارسال بسته‌های داده بر جای نمی‌گذارد! الگوریتم بردار فاصله که در بین چنین تعداد انبوه از مسیریاب‌ها تکرار گردد نیز قطعاً هرگز به تقارب نخواهد رسید. بدیهی است جهت کاهش پیچیدگی محاسبه‌ی مسیر در شبکه‌هایی به وسعت اینترنت جهانی راه‌کارهایی بایستی اندیشه نمود.

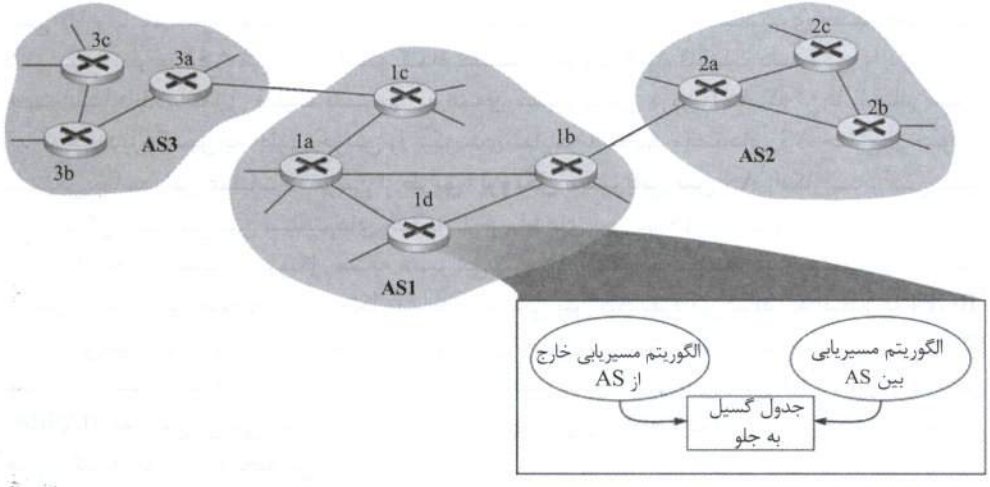
۲- **خود مختاری مدیریتی.** اگر چه پژوهشگران تلاش می‌کنند موضوعاتی از قبیل نیاز شرکت‌ها در به کارگیری مسیریاب‌های خویش را به گونه‌ی دلخواه در نظر نگیرند که یکی از این موارد دلخواه می‌تواند گزینش هر الگوریتم مسیریابی مورد نظر باشد و یا پنهان نگاه داشتن سازماندهی داخلی شبکه‌های شرکت از چشم بیگانگان، اما لحاظ نمودن این موارد بسیار مهم است. مطلوب آن است که یک نهاد بایستی قادر به پیاده‌سازی، اجرا و مدیریت شبکه‌ی خود به

1- Switched routing algorithms

گونه‌ی دلخواه باشد، در عین حالیکه قادر به اتصال شبکه‌اش به دیگر شبکه‌های بیگانه نیز باشد.

هر دو چالش می‌تواند از طریق سازماندهی مسیریاب‌ها در قالب **سیستم‌های خود مختار (AS)**[1] پیاده‌سازی گردد. که در آن هر AS شامل گروهی از مسیریاب‌ها می‌باشد که در معرض کنترل مدیریتی مشابه مانند ISP مشابه می‌باشند و یا تعلق به شبکه‌ی مشابهی دارند. مسیریاب‌های درون AS همگی الگوریتم مسیریابی یکسان مانند LS و DV را به اجرا درمی‌آورند و آگاهی از یکدیگر دارند، عیناً مانند الگوی تصوری که در بخش قبل تحلیل شد. الگوریتم مسیریابی اجرا شده در یک سیستم خودمختار اصطلاحاً **پروتکل مسیریابی درون سیستم خود مختار (IASRP)**[2] نامیده می‌شود. در این حالت لازم است هر یک از AS ها به یکدیگر متصل شده و بنابراین یک یا چندین مسیریاب در یک AS بایستی وظایف افزون‌تری را بر عهده گیرند که گسیل بسته‌ها به مقصد خارج از AS را انجام دهند. این مسیریاب‌ها **مسیریاب دریچه**[3] نامیده شده‌اند.

شکل ۴-۳۲ مثالی ساده از تعداد 3 سیستم خود مختار AS1 ، AS2 ، AS3 را نشان می‌دهد. در این شکل خطوط ضخیم پیوندهای ارتباطی مستقیم بین جفت مسیریاب‌ها می‌باشد و خطوط نازکتر زیر شبکه‌هایی هستند که مستقیماً به این روترها متصل شده‌اند. سیستم خودمختار AS1 شامل چهار مسیریاب است با شناسه‌های 1a ، 1b ، 1c ، 1d که پروتکل مسیریابی درونی AS را که در درون AS استفاده می‌شود به اجرا در می‌آورند. بنابراین هر یک از این چهار مسیریاب آگاهی از چگونگی گسیل بسته‌ها در امتداد مسیر بهینه‌ای به هر مقصدی در بین AS1 را دارند. به صورت مشابه سیستم‌های AS2 ، AS3 هر یک تعداد سه مسیریاب دارند. بدیهی است پروتکل‌های (IASRP) که AS1 ، AS2 و AS3 در حال اجرای آن هستند لازم نیست مشابه یکدیگر باشند. همچنین روترهای 1b ، 1c ، 2a و 3a همگی مسیریاب‌های از نوع مسیریاب دریچه یا دروازه می‌باشند.



شکل ۴-۳۲ ♦ مثالی از ارتباط مابین سیستم‌های خودمختار

1- Autonomous systems
2- Intra Autonomous System Routing Protocol
3- Gateways

اکنون بایستی آشکار شده باشد که چگونه مسیریاب‌ها در یک AS مسیرهای راهیابی برای جفت مبدا و مقصد را به دست می‌آورند. اما هنوز هم یک حلقه‌ی گم شده در معمای مسیریابی انتها به انتها وجود دارد! چگونه یک مسیریاب در بین AS می‌داند مسیردهی بسته‌ای را به یک مقصد خارج از خود انجام دهد؟ پاسخ بسیار ساده است اگر AS فقط دارای یک مسیریاب دریچه باشد که فقط به یکی دیگر از سیستم‌های خود مختار (AS) متصل است. در این حالت به دلیل آنکه پروتکل الگوریتم درونی (IASRP) مسیر با حداقل هزینه از هر مسیریاب داخلی به مسیر دریچه را محاسبه نموده است، هر مسیریاب از چگونگی گسیل بسته‌ها به جلو آگاهی دارد. مسیریاب دریچه هنگام دریافت بسته آن را به پیوندی گسیل می‌کند که هدایت بسته را به خارج از AS انجام می‌دهد. از طرف دیگر AS طرف مقابل مسئولیت مسیریابی بسته را به مقصد نهایی آن بر عهده می‌گیرد. به عنوان مثال اگر فرض شود مسیریاب 2b در شکل ۴-۳۲ بسته‌ای را دریافت دارد که مقصد آن خارج از AS2 باشد. مسیریاب 2b آنگاه بسته را به یکی از دو روتر 2a و یا 2c که از سوی جدول مسریابی 2b تعیین شده است تحویل می‌دهد. بدیهی است پیکربندی این جدول از سوی پروتکل مسیریابی درونی AS2 انجام شده است. بسته سرانجام به روتر دریچه یعنی 2a می‌رسد و آن بسته را به سوی مسیریاب 1b هدایت می‌کند. هنگامی که بسته 2a را ترک نماید وظیفه‌ی AS2 در برابر این بسته به پایان می‌رسد.

ملاحظه می‌شود هنگامی که AS مبدا فقط یک پیوند به سیستم‌های خودمختار خارج از خود دارد مشکل ساده است. اما اگر تعداد این پیوندها دو یا چندین پیوند باشد آنگاه وضعیت ASمبدا چگونه است؟ به عبارت دیگر تعداد مسیریاب‌های دریچه اگر بیش از یک در AS باشند چه اتفاقی روی می‌دهد؟ در این حالت اگاهی از اینکه بسته به کجا بایستی گسیل گردد دشوارتر می‌شود. به عنوان مثال یک مسیریاب در AS1 را در نظر بگیرید و فرض کنید بسته‌ای را دریافت می‌دارد که مقصد آن خارج از AS است. مسیریاب، آشکارا بایستی بسته را به یکی از دو مسیریاب دریچه‌اش هدایت نماید. یعنی 1b و یا 1c اما به کدام یک؟ جهت حل این مشکل AS1 نیازمند (۱) یادگیری مقصدهای قابل دسترسی از طریق AS2 می‌باشد و اینکه کدام مقصد از طریق AS3 قابلیت دسترسی را دارد و (۲) جهت انتشار اطلاعات این قابلیت دسترسی به همه‌ی مسیریاب‌های درون AS1 که آن‌ها را قادر سازد هر کدام جدول گسیل به جلوی خویش را پیکربندی نمایند تا بتوانند مقصدهای AS خارج از خود را دستیابی نمایند. این عملیات دسترسی از **طریق پروتکل مسیریابی بین AS**[1] امکان‌پذیر می‌شود. زیرا پروتکل مسیریابی بین سیستم‌های خودمختار ارتباط‌های بین دو AS را تحلیل می‌کند و بنابراین دو AS ارتباطی بایستی از پروتکل مشابه مسیریابی بین AS استفاده نمایند. در حقیقت در اینترنت همه‌ی سیستم‌های خودمختار از پروتکل مشابه مسیریابی بین AS بهره می‌گیرند که اصطلاحاً **BGP4**[2] نامیده می‌شود. این پروتکل در بخش بعدی این فصل تحلیل می‌شود. همانگونه که در شکل ۴-۳۲ نشان داده شده است هر مسیریاب اطلاعاتی را از پروتکل‌های مسیریابی بین AS و نیز از پروتکل (IASRP) که درون آن می‌باشد دریافت می‌کند و با بهره‌گیری از اطلاعات این دو پروتکل پیکربندی جدول گسیل به جلو را انجام می‌دهد.

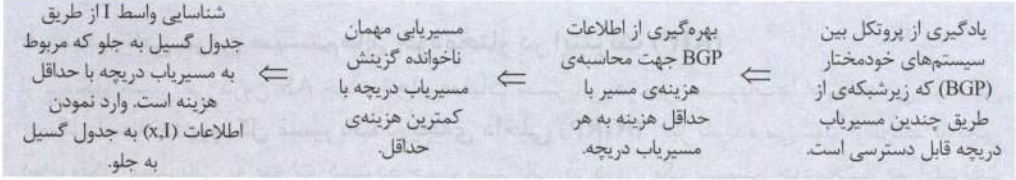
به عنوان مثال زیرشبکه‌ی x را در نظر بگیرید که شناسه‌ی آن یک آدرس (CIDR) است و فرض کنید سیستم خودمختار AS1 از طریق پروتکل مسیریابی بین AS خودآگاه می‌شود که زیرشبکه‌ی x از طریق AS3 قابل دسترسی است در صورتیکه از AS2 دسترسی به آن غیر ممکن است. شبکه‌ی AS1

1- Inter AS routing protocol
2- Border gateway protocol 4

آنگاه این اطلاعات را به همه‌ی مسیریاب‌های خود آگهی می‌کند. هنگامی که مسیریاب 1d آگاه می‌شود دسترسی زیر شبکه‌ی x از AS3 امکان‌پذیر است و بنابراین از طریق مسیریاب دریچه 1c نیز قابل دستیابی است آنگاه با بهره‌گیری از این اطلاعات که پروتکل (BGP) فراهم آورده است واسط روتر مسیر حداقل هزینه از 1d به مسیریاب دریچه‌ی 1c که I نام دارد، را شناسایی می‌کند. مسیریاب 1d جدول گسیل به جلوی خود را با اطلاعات (x,I) به روزرسانی می‌کند.

اکنون اگر AS2 و AS3 به دیگر سیستم‌های خودمختاری متصل شده باشند که در شکل نشان داده شده است و AS1 از طریق پروتکل (BGP) خودآگاه شود که زیرشبکه‌ی x از طریق هر دو AS2 و AS3 و به ترتیب از طریق دروازه‌های 1d و 1c قابل دسترسی است. سیستم خودمختار AS1 این اطلاعات را به همه‌ی مسیریاب‌های خود از جمله 1d منتقل می‌کند. اکنون مسیریاب 1d جهت به روزرسانی جدول گسیل به جلو بایستی گزینشی را بین 1d یا 1c انجام دهد که مایل است بسته‌ها را از طریق آن به زیرشبکه‌ی x و سرانجام مقصد نهایی هدایت نماید. یک رویکرد که عملاً مورد استفاده قرار می‌گیرد بهره‌گیری از روشی به نام **مسیریابی مهمان ناخوانده**[1] است. در این روش AS تلاش می‌کند هرچه سریعتر از مهمان ناخوانده (بسته‌ی داده) خلاصی یابد. به عبارت دیگر با صرف هر هزینه‌ی حداقل ممکن. این مهم از طریق مجبور نمودن یک مسیریاب به ارسال بسته‌ها به مسیریاب دریچه‌ای که دارای کمترین هزینه در میان همه ی مسیریاب‌های روتر به دریچه در بین دروازه‌ها به مقصد است، انجام می‌گیرد. در چارچوب مثال جاری مسیریابی مهمان ناخوانده در 1d در حال اجرا می‌باشد و با بهره‌گیری از اطلاعات پروتکل مسیریابی بین AS هزینه‌های مسیر به 1d و 1c را محاسبه می‌کند و سپس به گزینش مسیری می‌پردازد که از همه هزینه‌ی کمتری را دارد. هنگامی که این مسیر گزینش شود، مسیریاب 1d جدول خود را برای زیرشبکه‌ی x بر این مبنا به روزرسانی می‌کند. شکل ۴-۳۳ خلاصه‌ی عملیات انجام شده در مسیریاب 1d و افزودن اطلاعات برای x را در جدول گسیل به جلو نشان می‌دهد.

هنگامی که AS از مسیری به یک AS مجاور آگاه می‌شود، آنگاه آن AS می‌تواند اطلاعات مسیریابی را به برخی از مجاوران خود آگهی نماید. به عنوان مثال اگر AS1 از طریق AS2 آگاه شود که زیرشبکه‌ی x از طریق AS2 قابل دسترسی است. آنگاه AS1 سیستم خودمختار AS3 را می‌تواند از دستیابی به زیرشبکه‌ی x از طریق خویش آگاه نماید. در این حالت اگر AS3 نیازمند مسیردهی یک بسته به مقصد x باشد آنگاه AS3 بسته را به AS1 گسیل می‌کند و AS1 نیز بسته را به AS2 ارسال می‌کند. در بحث BGP در آینده‌ی نزدیک ملاحظه خواهد شد که یک AS از انعطاف‌پذیری بالایی جهت تصمیم‌گیری در مورد این که از کدام مقصد جهت آگهی اطلاعات به مجاورانش بهره گیرد، برخوردار است. این تصمیم‌گیری استراتژیک می‌باشد و بیش از آنکه وابسته به موارد منفی باشد مربوط به موضوعات اقتصادی قضیه است.



شکل ۴-۳۳ ♦ مراحل افزودن مقصد یک AS خارجی در جدول گسیل به جلوی مسیریاب

1- Hot potato routing

در بخش ۱-۱۵ ملاحظه شد اینترنت شامل سلسله مراتبی از ارتباط‌های بین ISP ها می‌باشد. اکنون این پرسش مطرح می‌شود که چه ارتباطی بین ISP ها وسیستم‌های خودمختار AS می‌تواند وجود داشته باشد؟ ممکن است این تصور وجود داشته باشد که مسیریاب‌ها در یک ISP و پیوندهایی که آن‌ها را به یکدیگر متصل نموده است یک AS را تشکیل می‌دهد. اگرچه در اغلب موارد این حالت درست است اما بسیاری از ISP ها شبکه‌های خود را در قالب چندین AS سازماندهی می‌کنند. به عنوان مثال بعضی از ISP های واقع در گروه اینترنت از یک AS برای تمامی شبکه‌های خود بهره‌ی‌گیرند، دیگران ISP خود را به ده‌ها AS متصل به یکدیگر تقسیم می‌کنند.

به شکل خلاصه چالش مقیاس و اعمال مدیریت از طریق سیستم‌های خود مختار قابل حل است. در درون یک AS همه ی مسیریاب‌ها به اجرای یک پروتکل مشابه مسیریابی(BGP) بین خود می‌پردازند. دشواری مربوط به مقیاس نیز این گونه برطرف می‌شود که یک مسیریاب درون AS فقط نیازمند آگاهی از مسیریاب‌های درون آن AS می‌باشد. چالش اعمال مدیریت نیز به دلیلی آنکه یک سازمان می‌تواند هر پروتکل مسیریابی بین AS یعنی (BGP) دلخواه خود را گزینش و پیاده‌سازی نماید برطرف می‌شود. بدیهی است هر جفت AS متصل به یکدیگر نیازمند اجرای پروتکل (BGP) مشابه بین خود می‌باشند که از طریق آن به مبادله‌ی اطلاعات مربوط به دسترسی به یکدیگر بپردازد.

در بخش بعدی این فصل پروتکل مسیریابی درون AS که عبارتند از (RIP) و (OSPF) و پروتکل مسیریابی بین سیستم‌های خودمختار یعنی (BGP) که در حال حاضر شبکه‌ی اینترنت از آن‌ها بهره می‌گیرد تحلیل شده‌اند. این موارد مطالعه‌ی موضوع مسیریابی سلسله‌مراتبی را کامل می‌کنند.

۱۹-۴ مسیریابی در اینترنت

پس از مطالعه‌ی آدرس‌دهی اینترنت و پروتکل IP اکنون وقت آن رسیده است که پروتکل های مسیریابی اینترنت مورد مطالعه قرار گیرد. وظیفه‌ی این پروتکل‌ها شناسایی مسیر پویش شده از سوی داده‌گرام‌ها بین مبدا و مقصد می‌باشد. در این مطالعه ملاحظه خواهد شد که پروتکل‌های مسیریابی اینترنت بسیاری از اصول مطالعه شده در اوایل این فصل را در درون خود جایگذاری نموده‌اند. رویکردهای وضعیت پیوند (LS) و بردار فاصله (DV) و نیز مفهوم سیستم‌های خودمختار همگی از لحاظ چگونگی عملیات مسیریابی در اینترنت امروزی مرکزیت ویژه‌ی خود را دارند.

لازم به یادآوری است که یک سیستم خودمختار (AS) مجموعه‌ای از مسیریاب‌های تحت مدیریت و کنترل فنی مشابه می‌باشند و همگی پروتکل مسیریابی مشترکی را بین یکدیگر به اجرا می‌گذارند. هر سیستم خودمختار شامل چندین زیرشبکه است. عبارت زیرشبکه به مفهوم دقیق خود عبارت است از آدرس‌دهی که قبلاً توضیح داده شد.

• مسیریابی درون سیستم‌های خودمختار در اینترنت (RIP)

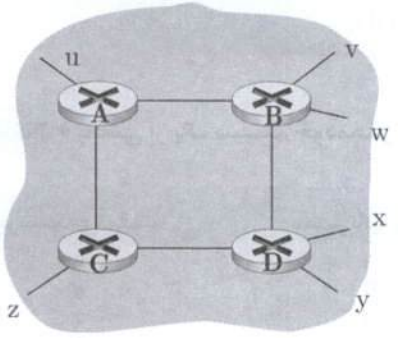
از پروتکل مسیریابی درون AS جهت انجام عملیات مسیریابی در بین مسیریاب‌ها استفاده می‌شود. این پروتکل اصطلاحاً **پروتکل مسیریاب دریچه‌ی داخلی (IGRP)**[1] نیز نامیده می‌شود. از لحاظ تاریخی دو پروتکل مسیریابی به صورت گسترده جهت مسیریابی در درون یک سیستم خودمختار در اینترنت

1- Interior gateway routing protocol

مورد بهره‌برداری قرار گرفته‌اند. یکی از این پروتکل‌ها اصطلاحاً **پروتکل اطلاعات مسیریابی (RIP)**[1] و دیگری پروتکل گزینش **آزاد نخستِ کوتاه‌ترین مسیر (OSPF)**[2] نامیده می‌شوند.پروتکل دیگری که شباهت بسیاری به (OSPF) دارد پروتکل **IS-IS** در مستند [RFC 1142] می‌باشد.

پروتکل RIP از جمله نخستین از نوع خود جهت انجام عملیات مسیریابی در درون AS به شمار می‌آید و هنوز هم از کاربرد گسترده‌ای برخوردار است.نام این پروتکل برگرفته از معماری سیستم شبکه‌ی زیراکس یا (XNS) می‌باشد. کاربرد گسترده‌ی RIP از آنجا ناشی می‌شود که در سال 1982 میلادی در نسخه‌ی **نرم‌افزار توزیعی (BSD)**[3] از سیستم‌عامل UNIX که مجموعه‌ی پروتکلی TCP/IP را پشتیبانی می‌کرد مورد استفاده قرار گرفت. نسخه‌ی اول RIP در مستند [RFC 1058] و نسخه‌ی دوم آن در [RFC 2453] توصیف شده است. پروتکل RIP یک الگوریتم بردار فاصله است. نسخه‌ی RIP در مستند [RFC 1058] از شمارش پرش کوتاه به عنوان هزینه استفاده می‌کند. به عبارت دیگر هر پیوند دارای هزینه‌ی **یک** می‌باشد. اگر چه در بخش‌های قبلی به دلیل ساده شدن موضوع الگوریتم DV هزینه‌ی بین جفت مسیریاب‌ها را محاسبه می‌نمود. اما در دو پروتکل RIP و OSPF هزینه در حقیقت از مسیریاب مبدا تا یک زیرشبکه‌ی مقصد می‌باشد. پروتکل RIP از عبارت hop به مفهوم پرش کوتاه استفاده می‌کند که تعداد زیرشبکه‌های پویش شده در کوتاه‌ترین مسیر بین روتر مبدا و زیرشبکه‌ی مقصد می‌باشد. شکل ۴-۳۴ یک سیستم خود مختار را با تعداد 6 زیرشبکه نشان می‌دهد. جدول نشان داده شده در شکل تعداد پرش‌های کوتاه از مبدا A به هر یک از زیرشبکه‌ها را نشان می‌دهد.

حداکثر هزینه‌ی یک مسیر به ارزش 15 محدود می‌شود. بنابراین بهره‌گیری از RIP به سیستم‌های خودمختار از لحاظ سطح پوشش یافته کمتر از 15 می‌باشد. در پروتکل DV مسیریاب‌های مجاور هم به تبادل بردارهای فاصله با یکدیگر می‌پرداختند. بردار فاصله برای هر روتر تخمین جاری فواصل کوتاه‌ترین مسیر از آن مسیریاب به زیرشبکه‌ها در AS است. در RIP اطلاعات به روزرسانی شده ی مسیرها بین مجاوران تقریباً هر 30 ثانیه یک‌بار با بهره‌گیری از **پیام واکنش RIP** صورت می‌پذیرد. پیام واکنش از سوی یک مسیریاب یا میزبان شامل فهرستی از 25 زیرشبکه‌ی مقصد در بین یک AS و نیز فاصله‌ی فرستنده‌ی پیام به هر یک از زیرشبکه‌ها می‌باشد. پیام‌های واکنش RIP همچنین به عنوان **آگهی RIP** نیز شناخته می‌شوند.



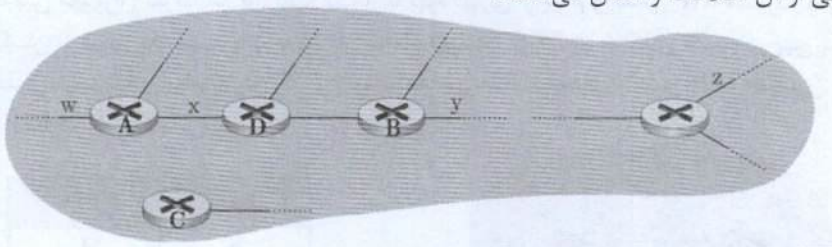
مقصد	hops
u	1
v	2
w	2
x	3
y	3
z	2

شکل ۴-۳۴ ♦ تعداد hop ها از مبدا A به زیرشبکه‌های گوناگون

1- Routing information protocol
2- Open shortest path first protocol
3- Brekly software disribution

اکنون به بررسی یک مثال ساده از چگونگی عملکرد آگهی‌های RIP می‌پردازیم. بخشی از یک AS را بر مبنای شکل ۴-۳۵ در نظر بگیرید. در این شکل خطوط ارتباطی که مسیریاب‌ها را به یکدیگر متصل نموده‌اند زیر شبکه قلمداد شده‌اند. در این شکل مسیریاب‌های A ، B ، C و D و زیرشبکه‌های w ، x ، y و z گزینش شده‌اند و بنابراین به آن‌ها برچسب‌های مربوط نیز اختصاص یافته است. خطوط نقطه‌چین به مفهوم ادامه‌ی AS می‌باشد و بنابراین این سیستم خودمختار دارای تعداد قابل توجهی مسیریاب و پیوند می‌باشد.

هر مسیریاب یک جدول RIP را به نام **جدول مسیریابی** نگهداری می‌کند. جدول مسیریابی هر روتر شامل اطلاعات بردار فاصله‌ی مسیریاب و جدول گسیل به جلو برای آن می‌باشد. شکل ۴-۳۶ جدول مسیریابی متعلق به روتر D می‌باشد. ملاحظه می‌شود جدول مسیریابی شکل ۴-۳۶ شامل سه ستون است. نخستین ستون زیرشبکه‌ی مقصد، ستون دوم شناسه‌ی مسیریاب بعدی در امتداد کوتاه‌ترین مسیر به مقصد زیرشبکه و ستون سوم تعداد زیرشبکه‌هایی که تا مقصد به همراه زیرشبکه‌ی مقصد پویش می‌شوند. به عبارت دیگر تعداد پرش‌های کوتاه (hops) در امتداد کوتاه‌ترین مسیر به زیرشبکه‌ی مقصد است. در این مثال آنچه از جدول می‌توان استنباط نمود آن است که اگر داده‌گرام‌ها از روتر D به زیرشبکه‌ی مقصد w گسیل شوند، آنگاه آن‌ها بایستی به مسیریاب مجاور A گسیل به جلو گردند. جدول همچنین مشخص می‌کند که زیرشبکه‌ی مقصد یعنی w تعداد دو پرش کوتاه یا هاپ در امتداد کوتاه‌ترین مسیر تا هدف فاصله دارد. علاوه بر موارد اشاره شده جدول ۴-۳۶ به این نکته نیز اشاره می‌کند که زیرشبکه‌ی z تعداد 7 پرش کوتاه تا مسیریاب B فاصله دارد. اصولاً یک جدول مسیریابی شامل یک ردیف برای هر زیرشبکه در AS می‌باشد. نسخه دوم پروتکل RIP اجازه‌ی انباشت اطلاعات ورودی زیرشبکه‌ها را به بهره‌گیری از فناوری انباشت مسیر می‌دهد، مانند آنچه در بخش ۴-۸ مطالعه گردید. جدول شکل ۴-۳۶ و جداول بعدی شامل اطلاعات کامل شده‌ی یک جدول نمی‌باشند بلکه بخشی از آن اطلاعات را نشان می‌دهند.



شکل ۴-۳۵ ♦ بخشی از یک سیستم خودمختار (AS)

زیرشبکه‌ی مقصد	مسیریاب بعدی	تعداد پرش‌های کوتاه به مقصد (hops)
w	A	2
y	B	2
z	B	7
X	-	1
- - -	- - -	- - -

شکل ۴-۳۶ ♦ جدول مسیریابی در روتر D قبل از دریافت اطلاعات مسیر از A

اکنون اگر فرض شود پس از گذشت 30 ثانیه مسیریاب D اطلاعات آگهی شده در جدول ۴-۳۷ را از مسیریاب A دریافت نموده باشد. بدیهی است این اطلاعات جدول مسیریابی ارسالی از سوی A

می‌باشد و نشان دهنده‌ی آن است که زیرشبکه z فقط تعداد 4 گام تا مسیریاب A فاصله دارد. هنگامی که مسیریاب D این آگهی اطلاعات را دریافت می‌کند آن‌ها را با اطلاعات قدیمی در جدول شکل ۴-۳۷ ترکیب می‌کند. به ویژه مسیریاب D یادگیری می‌کند که اکنون مسیری از درون A به زیرشبکه‌ی z وجود دارد که این مسیر کوتاه‌تر از مسیر پیشنهادی از طریق مسیریاب B می‌باشد. بنابراین مسیریاب D جدول خویش را به روزرسانی نموده تا اطلاعات مربوط به این کوتاه‌ترین مسیر را منظور نموده باشد. این موضوع در شکل ۴-۳۸ نشان داده شده است. ممکن است این پرسش برای شما مطرح شود که چگونه کوتاه‌ترین مسیر به زیرشبکه‌ی z در حقیقت کوتاه‌تر شده است؟ پاسخ آن است که این احتمال وجود دارد که الگوریتم بردار فاصله هنوز در فرآیند تقارب بر اساس بخش ۴-۱۶ این فصل قرار دارد و یا آنکه پیوندها و یا مسیریاب‌های جدیدی به AS افزوده شده‌اند که نتیجه‌ی آن تغییر در کوتاه‌تر شدن مسیرها در AS شده است.

زیرشبکه‌ی مقصد	مسیریاب بعدی	تعداد پرش‌های کوتاه به مقصد (hops)
z	C	4
w	-	1
X	-	1
- - -	- - -	- - -

شکل ۴-۳۷ ♦ آگهی از مسیریاب A

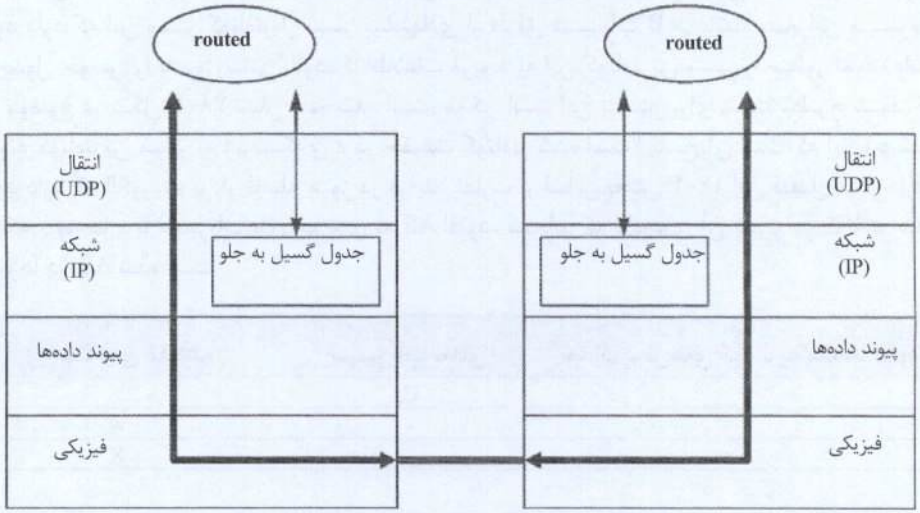
در ادامه‌ی مطالب چند حالت گوناگون از پیاده‌سازی پروتکل RIP بررسی می‌شود. این نکته یادآوری می‌شود که مسیریاب‌های RIP تقریباً هر 30 ثانیه یک بار به تبادل اطلاعات جداول مسیریابی می‌پردازند. اگر یک مسیریاب از مجاور خود حداقل بعد از گذشت 180 ثانیه یک بار اطلاعاتی را دریافت نکند آن مجاور از دید مسیریاب غیر قابل دسترس قلمداد می‌شود به این مفهوم که روتر مجاور یا از کار افتاده است و یا آنکه پیوند ارتباطی آن دچار اشکال می‌باشد. هنگامی که چنین حالتی پدیدار شود پروتکل RIP جدول مسیریابی محلی خود را بر این اساس تغییر می‌دهد و سپس این اطلاعات را از طریق آگهی به مسیریاب‌های مجاور که هنوز قابل دسترسی می‌باشند اطلاع می‌دهد.

زیرشبکه‌ی مقصد	مسیریاب بعدی	تعداد پرش‌های کوتاه به مقصد (hops)
w	A	2
Y	B	2
z	A	5
- - -	- - -	- - -

شکل ۴-۳۸ ♦ جدول مسیریابی D پس از دریافت آگهی از مسیریاب A

یک مسیریاب می‌تواند درخواست اطلاعات پیرامون مجاور خویش را از نقطه‌نظر هزینه به یک مقصد مورد نظر از طریق بهره‌گیری از پیام تقاضای RIP نیز داشته باشد. مسیریاب‌ها درخواست‌های RIP خود و واکنش آن‌ها نسبت به یکدیگر را از طریق پروتکل UDP با بهره‌گیری از پورت شماره‌ی 520 ارسال و یا دریافت می‌کنند. قطعه‌ی UDP حمل شده بین مسیریاب‌ها به صورت یک داده‌گرام استاندارد سازماندهی می‌شود. این واقعیت که پروتکل RIP از یک پروتکل لایه‌ی انتقال یعنی (UDP) برفراز یک پروتکل لایه‌ی شبکه یعنی (IP) بهره می‌گیرد تا یک عملیات لایه‌ی شبکه که همان الگوریتم

مسیریابی است را پیاده‌سازی نماید به نظر پیچیده می‌آید. با تامل بر چگونگی پیاده‌سازی RIP این موضوع بیشتر آشکار می‌شود.



شکل ۴-۳۹ ♦ پیاده‌سازی RIP به عنوان یک قلمرو مسیردهی شده (routed)

شکل ۴-۳۹ نشان می‌دهد چگونه RIP در یک سیستم UNIX پیاده‌سازی می‌شود. به عنوان مثال یک پایانه‌ی UNIX که به صورت یک مسیریاب عمل می‌کند در این شکل به تصویر درآمده است. یک فرآیند با نام (routed) به اجرای پروتکل RIP می‌پردازد. به عبارت دیگر اطلاعات مسیریابی و پیام‌های مبادله شده را به همراه فرآیندهای (routed) اجرا شده در مسیریاب‌های مجاور نگهداری می‌کند. از آنجائیکه RIP به عنوان یک فرآیند لایه‌ی کاربرد پیاده‌سازی شده است بنابراین می‌تواند به ارسال و دریافت پیام‌ها از طریق یک سوکت استاندارد با بهره‌گیری از لایه‌ی انتقال اقدام نماید.

۴-۲۰ مسیریابی درونی سیستم خودمختار در اینترنت (OSPF)

پروتکل مسیریابی OSPF همانند پروتکل RIP به صورت گسترده‌ای برای مسیریابی درونی AS در اینترنت به کار گرفته می‌شود. این پروتکل نسبت بسیار نزدیکی با پروتکل دیگری به نام IS -IS دارد که در ISP های لایه‌های بالاتر اینترنت به کار گرفته می‌شود. این در حالی است که RIP در ISP های لایه‌های پائین‌تر و در شبکه‌های سازمانی مورد بهره‌برداری قرار می‌گیرد. واژه‌ی **آزاد** (Open) در پروتکل OSPF به مفهوم آن است که مشخصات پروتکل مسیریابی به گونه‌ی عمومی (فراگیر) در دسترس می‌باشد. به عبارت دیگر مانند پروتکل مشابه (EIGRP) انحصاری نمی‌باشد. جدیدترین نسخه OSPF در مستند [RFC 2328] تعریف شده است.

پروتکل OSPF به عنوان جایگزین RIP مورد پذیرش واقع شده است. در مرکز ثقل این پروتکل یک پروتکل وضعیت پیوند (LS) پیاده سازی شده است که با بهره‌گیری از اطلاعات وضعیت پیوند و یک الگوریتم مسیر با حداقل هزینه‌ی دایجسترا عملیات و وظایف خود را دنبال می‌نماید. با بهره‌گیری از پروتکل OSPF یک مسیریاب نقشه‌ی کاملی از ساختار شبکه یعنی یک گراف را که در برگیرنده‌ی همه ی سیستم‌های خودمختار است تشکیل می‌دهد. مسیریاب آنگاه الگوریتم کوتاه‌ترین مسیر دایجسترا را

به گونه‌ی محلی به اجرا درمی‌آورد تا یک درخت کوتاه‌ترین مسیر به همه‌ی زیرشبکه‌ها را ایجاد نماید. ریشه ی درخت تشکیل شده مسیریاب محلی اجرا کننده‌ی الگوریتم است. هزینه‌های هر یک از پیوندها از سوی مدیر شبکه پیکربندی می‌شود. مدیر شبکه می‌تواند همه‌ی پیوندها را از لحاظ هزینه برابر یک نماید. نتیجه‌ی این رویکرد حداقل مسیریابی به صورت پرش کوتاه است و یا ممکن است توزین پیوند را به گونه‌ی نسبت معکوس ظرفیت هر پیوند گزینش نماید تا از جاری شدن ترافیک بر روی پیوندهای کم سرعت جلوگیری نماید. پروتکل OSPF هیچ قاعده‌ای را برای چگونگی توزین پیوند دنبال نمی‌کند. این مهم بر عهده‌ی مدیر شبکه است. به جای آن پروتکل لازم برای محاسبه‌ی مسیر با حداقل هزینه را برای یک مجموعه از سنجش‌های پیوند فراهم می‌آورد.

در پروتکل OSPF هر مسیریاب اطلاعات مسیریابی خود را به روترهای دیگر در یک AS فراپخش می‌کند. به عبارت دیگر عملیات محدود به مجاوران آن نمی‌شود. اطلاعات وضعیت پیوند هرگاه که تغییری در وضعیت یک پیوند ملاحظه شود از سوی مسیریاب فراپخش می‌شود. تغییرات ممکن است شامل هزینه‌ها و یا وضعیت فعال و غیرفعال پیوندها باشد. یک مسیریاب همچنین به صورت متناوب و حداقل هر 30 دقیقه یک بار وضعیت هر پیوند را آگهی می نماید حتی اگر وضعیت آن پیوند دچار تغییر نشده باشد. در مستند [RFC 2338] به این نکته اشاره شده است که «این به روزرسانی متناوب از آگهی وضعیت پیوند سبب افزایش قدرت‌مندی الگوریتم LS می‌شود». آگهی‌های پروتکل ISPF در پیام‌های آن خلاصه شده‌اند که مستقیماً به وسیله‌ی IP با یک پروتکل لایه‌ی بالاتر 89 برای OSPF حمل می‌شوند. بنابراین پروتکل OSPF بایستی پیاده‌سازی عملیاتی مانند انتقال مطمئن پیام و فراپخش LS را انجام دهد. علاوه بر این موارد سنجش و آزمون عملیات و فعال بودن پیوندها از طریق پیام (HELLO) به مجاوران مستقیماً پیوسته به آن از جمله وظایف OSPF است. امکان دستیابی به بانک اطلاعات مسیریاب مجاور از نقطه‌نظر وضعیت پیوند از جمله موارد دیگر این پروتکل است. در ذیل برخی از برجستگی‌های منظور شده در پروتکل OSPF تشریح می‌گردد:

۱- **امنیت.** مبادلات بین مسیریاب‌های OSPF مانند به روزرسانی‌های LS می‌تواند اعتبارسنجی گردد. اعمال اعتبارسنجی فقط به مسیریاب‌های امین امکان شرکت در عملیات پروتکل OSPF در بین یک AS را می‌دهد و از نفوذ مهاجمان جلوگیری می‌شود. به این ترتیب از ورود و یا تزریق اطلاعات نادرست به جداول مسریابی روترها از سوی افراد و یا شبکه‌های مشکوک ممانعت به عمل می‌آید. به گونه‌ی پیش‌فرض بسته‌های OSPF بین مسیریاب‌ها اعتبارسنجی نمی‌شوند و بنابراین می‌توانند جعل شوند. دو نوع روش اعتبارسنجی در OSPF قابل تعریف است. روش **ساده** و نوع دوم که **MD5** نام گرفته است. در فصل ۸ این موارد تحلیل بیشتری شده‌اند. در حالت اعتبارسنجی ساده از یک رمز عبور مشابه در هر مسیریاب استفاده می‌شود. هنگامی که یک روتر بسته‌ی OSPF را ارسال می‌کند یک رمز عبور در قالب متن ساده را به آن اضافه می‌کند. بدیهی است اعتبارسنجی ساده دارای امنیت چندانی نمی‌باشد. در روش اعتبارسنجی MD5 از کلید رمز مشترک که در همه‌ی مسیریاب‌ها پیکربندی شده است بهره‌گیری می‌شود. برای هر بسته‌ی OSPF ارسالی مسیریاب یک کد MD5 ویژه را با توجه به اطلاعات درون بسته‌ی OSPF به گونه درهم سازی محاسبه می‌کند و بسته‌ی OSPF را با کلید رمز درهم می‌آمیزد. آنگاه مسیریاب ارزش hash نتیجه شده را در بسته‌ی OSPF قرار می‌دهد. مسیریاب دریافت کننده با بهره‌گیری از یک کلید رمز از قبل تعریف شده‌ای کُد MD5 جدید را برای بسته‌ی OSPF دریافتی محاسبه و آن را با ارزش حمل شده از سوی بسته مقایسه

می‌کندو در نتیجه عملیات اعتبارسنجی MD5 انجام می‌گردد. از اعداد به صورت پی در پی نیز به همراه اعتبارسنجی MD5 جهت حفاظت در برابر تهاجم‌های تکراری استفاده می‌شود.

۲- **مسیرهای چندگانه با هزینه‌ی مشابه.** هنگامی که چندین مسیر به یک مقصد هزینه‌های مشابه دربردارند. پروتکل OSPF امکان می‌دهد از مسیرهای چندگانه بهره‌گیری شود. به عبارت دیگر لازم نیست که از یک مسیر فقط جهت حمل همه‌ی ترافیک بهره‌گیری شود در حالیکه چندین مسیر با هزینه مشابه وجود دارد.

۳- **پشتیبانی یک‌پارچه از مسیریابی یک یا چندپخشی.** پروتکل OSPF چندپخشی که اصطلاحاً (MOSPF) خوانده می‌شود و در مستند [RFC 1584] تعریف شده است جهت مسیریابی چندپخشی مورد بهره‌برداری قرار می‌گیرد. این پروتکل به بانک اطلاعات پیوندهای OSPF نوع جدید آگهی LS را به روش فراپخش وضعیت پیوند OSPF می‌افزاید.

۴- **پشتیبانی رعایت سلسله‌مراتب در درون یک قلمرو و مسیریابی.** شاید یکی از چشمگیرترین پیشرفت‌های حاصل در OSPF توان‌مندی ساختار یک سلسله مراتب OSPF است. در بخش ۴-۱۸ بسیاری از ویژگی‌های ساختار مسیریابی سلسله‌مراتبی بررسی گردید. در ادامه‌ی مطالب پیاده‌سازی مسیریابی سلسله‌مراتبی OSPF بررسی می‌شود.

یک سیستم خودمختار OSPF را می‌توان در قلمروهای سلسله‌مراتبی پیکربندی نمود. هر قلمرو الگوریتم مسیریابی وضعیت پیوند OSPF خود را اجرا می‌کند. هر یک از مسیریاب‌ها در هر ناحیه وضعیت پیوند خویش را به بقیه‌ی مسیریاب‌های آن ناحیه آگهی می‌کند. در درون هر محدوده یک یا چندین **مسیریاب مرزی قلمرو** مسئول مسیریابی بسته‌های خارج آن ناحیه می‌باشند. یک قلمرو OSPF در سیستم خودمختار به عنوان **زیرساخت** پیکربندی می‌شود. نقش اصلی قلمرو زیرساخت مسیردهی ترافیک بین دیگر قلمروهای AS می‌باشد. زیرساخت همیشه شامل همه‌ی مسیریاب‌های مرزی قلمرو در AS است و ممکن است روترهای مجاور را نیز شامل شود. مسیریابی بین قلمرو در درون یک AS نیازمند آن است که بسته ابتدا به یک روتر ناحیه‌ی مرزی هدایت شده به عبارت دیگر مسیریابی درون قلمرو انجام گیرد و سپس بسته به زیرساخت و از طریق آن به مسیریاب مرزی قلمرو که در محدوده‌ی مقصد قرار گرفته است هدایت شود و پس از آن به مقصد نهایی گسیل گردد.

مطالعه‌ی موارد مربوط به مسیریابی وضعیت پیوند با بهره‌گیری از این فرض صورت گرفت که هر پیوند توزین ویژه‌ای را شامل می‌شود که در الگوریتم مسیریابی مانند OSPF اجرا می‌شود و جریان ترافیک بر مبنای اطلاعات جدول مسیریابی محاسبه شده از سوی الگوریتم LS جاری می‌گردد. در تفسیر کُنش و واکنش، وزن پیوندها در اولویت قرار می‌گیرند و نتیجه‌ی حاصل از الگوریتم دایجسترا یعنی مسیریابی‌هایی انجام گرفته با حداقل هزینه‌ی کلی حاصل می‌شود. با این نگرش توزین پیوندها منعکس کننده‌ی هزینه‌ی بهره‌گیری از یک پیوند است به عبارت دیگر اگر توزین پیوند نسبت معکوس با ظرفیت آن پیوند داشته باشد آنگاه پیوندهای با ظرفیت بالا وزن کمتری را به خود اختصاص می‌دهند و احتمال گزینش آن‌ها جهت عملیات مسیریابی افزایش می‌یابد. الگوریتم دایجسترا نیز در پی به حداقل رساندن هزینه‌ها می‌باشد.

در عمل رابطه‌ی کنش و واکنش بین توزین پیوندها و مسیرهای گزینش شده ممکن است عکس یکدیگر باشند. به این مفهوم که پیکربندی شبکه جهت دستیابی مسیرهایی با اهداف مهندسی شده‌ی ترافیک ویژه صورت پذیرفته باشد. نتیجه‌ی آن پیکربندی توزین پیوندها به گونه‌ی ویژه‌ای می‌باشد. به

عنوان مثال اگر فرض شود پیش‌بینی برای ترافیک شبکه در هر یک از نقاط ورودی به خروجی انجام گرفته باشد و نیازمند یک مسیریابی ویژه از جریان ورودی به خروجی باشد که حداکثر بهره‌وری همه‌ی پیوندهای شبکه را حداقل می‌نماید. آنگاه در یک الگوریتم مسیریابی مانند OSPF تنها امکان مسیریابی جریان ترافیک از درون شبکه با کمک وزن اختصاص یافته به پیوندها می‌باشد. بنابراین جهت دستیابی به حداقل بهره‌وری از یک مقدار حداکثر بایستی مجموعه‌ی توزین پیوندهایی محاسبه شود که دسترسی به این هدف را امکان‌پذیر نماید. این حقیقت معکوس رابطه‌ی کنش و واکنش مسیریابی مطلوب جریان ترافیک به شمار می‌آید و وزن پیوند OSPF بایستی به گونه‌ای محاسبه شود که الگوریتم مسیریابی OSPF جریان مطلوبی از مسیریابی مورد نظر نتیجه دهد.

۴-۲۱ مسیریابی بین سیستم‌های خود مختار (BGP)

در بخش قبل ملاحظه شد چگونه ISP با بهره‌گیری از RIP و OSPF مسیرهای بهینه بین جفت گره‌ی مبدا و مقصد را که در درون یک AS قرار دارند محاسبه می‌کند. در ادامه محاسبه‌ی مسیرها بین جفت گره‌ی مبدا و مقصد مابین سیستم‌های خودمختار بررسی می‌شود. نسخه 4 پروتکل **مسیریاب دریچه‌ی مرزی**[1] (BGP) که در مستند [RFC 4271] تعریف شده است به عنوان پروتکل استاندارد مسیریابی بین AS در حال حاضر در شبکه‌ی اینترنت به شمار می‌آید. این پروتکل اصطلاحاً BGP4 و به صورت اختصاری آن BGP نامیده می‌شود. یک پروتکل مسیریابی بین AS امکانات ذیل را فراهم می‌آورد:

- اطلاعات دسترسی به زیرشبکه را از جانب سیستم‌های خودمختار مجاور بدست می‌آورد.
- اطلاعات دسترسی را به همه‌ی مسیریاب‌های درونی AS انتشار می‌دهد.
- مسیر بهینه به یک زیر شبکه را بر مبنای اطلاعات دسترسی و نیز رویکرد هر AS محاسبه می‌کند.

مهم‌تر از همه‌ی موارد بالا پروتکل BGP امکان می‌دهد هر زیر شبکه به اگهی موجودیت خود به بقیه‌ی اعضا در شبکه‌ی اینترنت اقدام نماید. به عبارت دیگر یک زیرشبکه فریاد می‌زند « من وجود دارم. من اینجا هستم» و BGP این اطمینان را حاصل می‌کند که همه‌ی سیستم‌های خود مختار در اینترنت آگاهی از زیرشبکه و چگونگی رسیدن به آن را در اختیار دارند. اگر BGP وجود نداشت هر زیر شبکه با دیگر شبکه‌ها در انزوا قرار می‌گرفت.

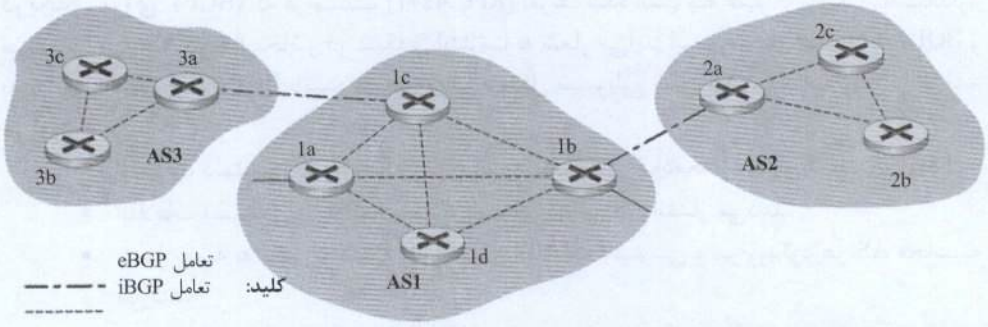
۱- اصول BGP

پروتکل BGP برخوردار از پیچیدگی فراوان است. مطالعات گسترده‌ای پیرامون آن در جریان است. مهارت در بکارگیری BGP به مطالعه‌ی زیاد و تجربه‌ی عملی گسترده‌ای نیاز دارد. از آنجائیکه این پروتکل بسیار حیاتی برای شبکه‌ی اینترنت به شمار می‌آید، بنابراین درک مقدماتی از آن و چگونگی عملکرد آن اهمیت قابل ملاحظه‌ای دارد. ابتدا در چارچوب یک شبکه‌ی ساده مانند شکل ۴-۳۲ عملکرد BGP تحلیل می‌شود، به دنبال آن موضوع مسیریابی سلسله مراتبی دنبال می‌شود.

در پروتکل BGP تبادل اطلاعات مسیریابی بین جفت روترها با یک ارتباط TCP نیمه دایم انجام می‌شود که از درگاه 179 بهره می‌گیرد. ارتباط‌های نیمه دایمی TCP برای شبکه‌ی شکل ۴-۳۲ در

1- Border gateway protocol

تصویر ۴-۴۰ نشان داده شده‌اند. در این شکل یک ارتباط ویژه‌ای از نوع BGPTCP برای هر پیوندی که مستقیماً دو مسیریاب را در دو AS مختلف به یکدیگر متصل می‌کند وجود دارد. بنابراین در شکل ۴-۴۰ یک ارتباط TCP بین مسیریاب‌های دریچه‌ی 3a و 1c و ارتباط TCP دیگری بین مسیریاب‌های 1b و 2a وجود دارد. علاوه بر آن ارتباط‌های BGPTCP غیر دایمی بین مسیریاب‌های درون یک AS نیز وجود دارند. به ویژه شکل ۴-۴۰ یک پیکربندی مشترک از یک ارتباط TCP برای هر جفت مسیریاب‌های داخلی به یک AS را نشان می‌دهد که شبکه‌ای از ارتباط‌های TCP در درون هر AS را خلق می‌کنند. برای هر ارتباط TCP دو مسیریاب حاضر در انتهای ارتباط اصطلاحاً **نظیرهای (BGP)**[1] نامیده می‌شود و ارتباط TCP در امتداد همه‌ی پیام‌های BGP از طریق ارتباطی گسیل می‌شوند که آن را **تعامل (BGP)**[2] نامیده‌اند. علاوه بر همه‌ی موارد اشاره یک تعامل BGP پوششی از دو AS است که آن را **تعامل خارجی (EBGP)**[3] می‌نامند و یک تعامل BGP بین مسیریاب‌های درون AS که **تعامل درونی (IBGP)**[4] نامیده شده است. در شکل ۴-۴۰ تعامل‌های (eBGP) از طریق خطوط شکسته‌ی طولانی ضخیم و تعامل‌های (iBGP) با خطوط شکسته‌ی کوتاه نازک نشان داده شده‌اند. بدیهی است خطوط تعامل (BGP) در شکل ۴-۴۰ همیشه به مفهوم پیوندهای فیزیکی در شکل ۴-۳۲ نمی‌باشند.



شکل ۴-۴۰ ♦ تعامل‌های (iBGP) و (eBGP)

پروتکل BGP اجازه می‌دهد هر AS به یادگیری مقصدهای قابل دسترس از طریق همسایگان دیگرش بپردازد. در BGP مقصدها میزبان‌ها نمی‌باشند بلکه ارزش‌های سایدر است (CIDR) که هر پیش‌وند ان یک زیرشبکه یا مجموعه‌ای از زیرشبکه‌ها را نمایندگی می‌کند. به عنوان مثال اگر تعداد 4 زیرشبکه‌ی متصل به AS2 به گونه‌ی : 138.16.64/24 ، 138.16.65/24 ، 138.16.66/24 ، 138.16.67/24 باشند آنگاه AS2 می‌تواند پیش‌وندهای این 4 زیرشبکه را انباشت نماید و از (BGP) جهت آگهی فقط یک پیش‌وند با آدرس 138.16.64/22 به AS1 استفاده نماید. در مثال دیگر اگر فقط 3 زیرشبکه از این 4 زیرشبکه در AS2 و زیرشبکه‌ی چهارم یعنی 138.16.67/24 در AS3 باشد آنگاه همان‌گونه که در بخش ۴-۹ تشریح شد به علت آن که مسیریاب‌ها از نگاشت طولانی‌ترین پیش‌وند جهت گسیل به جلو برای داده‌گرام‌ها بهره می‌گیرند بنابراین AS3 پیش‌وندهای مشخص 138.16.67/24 را به AS1 آگهی نماید و AS2 هنوز می‌تواند پیش‌وند انباشتی 138.16.64/22 را به AS1 آگهی نماید.

1- BGP peers
2- BGP sessions
3- External BGP sessions
4- Internal BGP sessions

اکنون چگونگی توزیع اطلاعات دسترسی پیش‌وند از طریق تعامل (BGP) در شکل ۴-۴۰ بررسی می‌شود. همانگونه که انتظار می‌رود با بهره‌گیری از تعامل (eBGP) بین مسیریاب‌های دریچه ی 3a و 1c سیستم خودمختار AS3 فهرستی از پیش‌وندهای قابل دسترس از خود را به AS1 گسیل می‌کند. همچنین AS1 فهرستی از پیش‌وندهای قابل دسترس خود و یا AS2 را از طریق مسیریاب‌های 1b و 2a با یکدیگر مبادله می‌کنند. بدیهی است هنگامیکه یک مسیریاب دریچه در هر سیستم خودمختار پیش‌وندهای یادگیری شده (eBGP) را دریافت دارد آن مسیریاب با بهره‌گیری از تعامل (iBGP) خود به توزیع پیش‌وندها به دیگر مسیریاب‌ها در AS می‌پردازد. بنابراین همه‌ی مسیریاب‌ها در AS1 به یادگیری در مورد پیش‌وندهای AS3 می‌پردازند که شامل روتر 1b نیز می‌شود. مسیریاب دریچه‌ی 1b در AS1 بنابراین می‌تواند پیش‌وندهای AS3 را به AS2 آگهی کند. هنگامی‌که یک مسیریاب که از نقطه‌نظر نوع مهم نمی‌باشد (دریچه یا غیر از آن) پیش‌وند جدید را یادگیری می‌کند، یک ورودی را برای پیش‌وند موجود در جدول گسیل به جلوی خود بر مبنای بخش ۴-۱۸ خلق می‌کند.

۲- ویژگی مسیر و مسیریاب‌های BGP

با درک مقدماتی که از BGP در بخش قبل حاصل شد در ادامه مسیرها و ویژگی آن‌ها در BGP بررسی می‌شود. یک سیستم خودمختار به گونه‌ی فراگیر از طریق انحصار یک **شماره سیستم خود مختار (ASN)** که در مستند [RFC 1930] تشریح شده است، شناسایی می‌شود. در عمل هر AS شامل یک ASN نیست به ویژه یک سیستم خودمختار ریشه فقط ترافیکی را حمل می‌کند که فقط برای آن مبدا و مقصد است و بنابراین شامل یک ASN نمی‌باشد.

شماره‌های AS مانند آدرس‌های IP از سوی ICANN بر مبنای مستند [ICANN 2009] اختصاص می‌یابند.

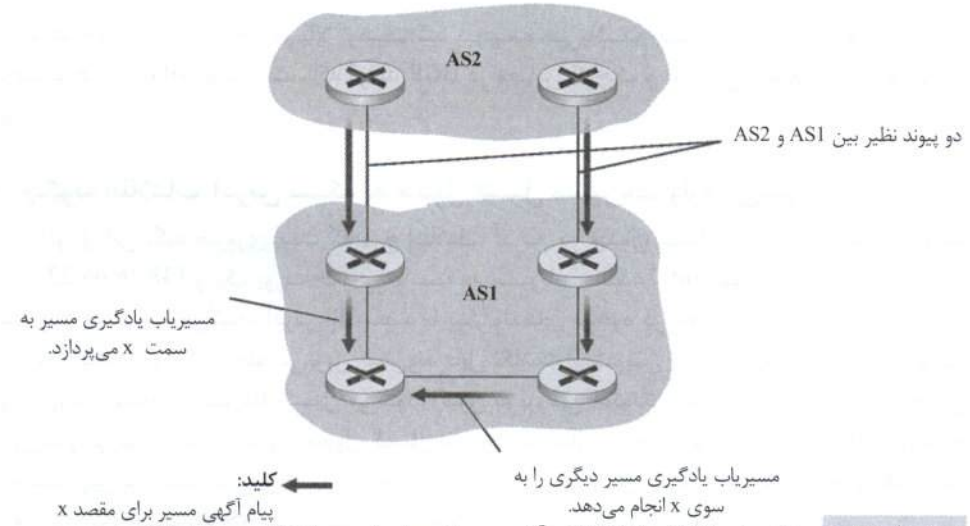
هنگامی که یک مسیریاب پیش‌وندی را در یک تعامل BGP آگهی می‌دهد همراه آن یک شماره از ویژگی‌های BGP را نیز اختصاص می‌دهد. در واژگان BGP یک پیش‌وند به همراه ویژگی آن را یک مسیر (route) می‌نامند. بنابراین نظیرهای BGP مسیرها را بین یکدیگر آگهی می‌کنند. دو فقره‌ی مهم از این ویژگی‌ها عبارتند از AS – PATH و NEXT – HOP.

- **AS – PATH مسیر سیستم خودمختار**. این ویژگی شامل یک AS است که آگهی پیش‌وند از آن عبور نموده است. هنگامی که یک پیش‌وند از یک AS عبور می‌کند، سیستم خودمختار ASN خود را به ویژگی AS – PATH اضافه می‌کند. به عنوان مثال مجدداً شکل ۴-۴۰ را در نظر بگیرید و فرض کنید که پیش‌وند 138.16.64/24 نخست از AS2 به AS1 آگهی شده است. اگر AS1 سپس پیش‌وند را به AS3 آگهی نماید آنگاه AS – PATH به صورت AS2AS1 می‌باشد. مسیریاب‌ها از AS – PATH جهت شناسایی و ممانعت از تکرار آگهی‌ها بهره می‌گیرند. به ویژه اگر یک مسیریاب ملاحظه کند که AS آن در فهرست مسیر موجود می‌باشد آنگاه از پذیرش آگهی خودداری می‌کند. مسیریاب‌ها همچنین از ویژگی AS – PATH هنگام عملیات گزینش در بین چندین مسیر با پیش‌وندهای مشابه بهره می‌گیرند.
- **NEXT – HOP پرش کوتاه بعدی**. با توجه به رابطه‌ی حیاتی بین پروتکل‌های مسیریابی درون و بین AS ها ویژگی پرش کوتاه بعد یا NEXT – HOP کاربرد بسیار مهمی را شامل می‌شود. NEXT – HOP در حقیقت واسط مسیریابی است که آغازگر AS – PATH به

شمار می‌آید. جهت درک بهتر این ویژگی مجدداً شکل ۴-۴۰ را بررسی می‌کنیم. حالتی را در نظر بگیرید که مسیریاب دریچه‌ی 3a در AS3 مسیری را به گره‌ی مانند خود 1c در AS1 با بهره‌گیری از (eBGP) آگهی می‌کند. مسیر اشاره شده شامل پیش‌وند آگهی شده است به نام x به همراه یک AS – PATH در آن پیش‌وند. این اطلاعات آگهی شده همچنین شامل NEXT – HOP نیز می‌باشد که آدرس IP واسط روتر 3a می‌باشد و به سوی 1c هدایت شده است. لازم به یادآوری است که یک مسیریاب شامل چندین آدرس IP است که یکی از آن‌ها متعلق به واسط آن می‌باشد. اکنون حالتی را مجسم کنید که مسیریاب 1d از طریق پروتکل (iBGP) از وجود این مسیر آگاه می‌شود. پس از آگاهی از این مسیر به x، روتر 1d ممکن است مایل به ارسال بسته‌ها به x در امتداد آن مسیر باشد، به عبارت دیگر 1d مایل به افزودن ورودی (x,l) به جدول گسیل به جلوی خویش باشد که l واسط آن به شمار می‌آید که آغاز آن مسیری با حداقل هزینه از 1d به طرف مسیریاب دریچه ی 1d می‌باشد. به منظور محاسبه ی l مسیریاب آدرس IP در ویژگی NEXT - HOP را برای مدول مسیریابی درونی AS خویش تعریف می‌کند. الگوریتم مسیریابی درونی AS مسیر با حداقل هزینه را برای همه‌ی زیرشبکه‌های متصل به مسیریاب‌ها در AS1 که شامل زیرشبکه با پیوندی بین 1c و روتر 3a می‌باشد فراهم می‌نماید . از این مسیر با حداقل هزینه از 1d به زیرشبکه‌ی 3a – 1c مسیریاب 1d واسط مسیریاب l را محاسبه می‌کند که آغاز کننده‌ی این مسیر است و سپس ورودی (x,l) را به جدول مسیریابی به جلوی خویش اضافه می‌کند. اندکی گیج کننده است! سرانجام ویژگی NEXT-HOP از سوی مسیریاب جهت پیکربندی مناسب جدول گسیل به جلو مورد استفاده قرار می‌گیرد.

- بار دیگر شکل ۴-۴۱ حالت دیگری را نشان می‌دهد که NEXT-HOP مورد نیاز است. در این شکل AS1 و AS2 از طریق دو پیوند نظیر به یکدیگر متصل شده‌اند. یک مسیریاب در AS1 می‌تواند از دو مسیر گوناگون آگاهی پیدا کند که پیش‌وند مشابه x را دارند. این دو مسیر می‌توانند دارای ویژگی مشابه AS – PATH به x باشند، اما ارزش‌های متفاوتی از NEXT – HOP را اختیار نماید که مربوط به پیوندهای نظیر مختلف می‌باشند. با بهره‌گیری از ارزش‌های NEXT-HOP و الگوریتم مسیریابی درونی AS یک مسیریاب می‌تواند هزینه‌ی مسیر به هر پیوند نظیر را محاسبه نماید و سپس مسیریابی مهمان ناخوانده در بخش ۴-۱۸ را به آن اعمال نماید تا واسط مناسب را بدست آورد.

پروتکل BGP همچنین شامل ویژگی‌های دیگری است که به مسیریاب‌ها اجازه می‌دهد ارزش‌های ترجیحی را به مسیر اختصاص دهند. یک ویژگی که نشان می‌دهد چگونه پیش‌وند BGP به AS اصلی وارد شده است. هنگامی که یک مسیریاب دریچه آگهی یک روتر را دریافت می‌کند، از الگوی سیاست‌گذاری خود جهت پذیرش و یا حذف مسیر و نیز اختصاص ویژگی‌هایی مانند ارزش‌های ترجیحی بهره می‌گیرد. الگوی سیاست‌گذاری ممکن است یک مسیر را به دلیل آنکه AS مایل به ارسال ترافیک از طریق یکی از مسیرهای AS – PATH نمی‌باشد، حذف نماید. مسیریاب دریچه نیز ممکن است مسیری را به دلیل آنکه یکی مسیر ترجیحی با پیش‌وند مشابه را در اختیار دارد حذف نماید.



شکل ۴-۴۱ ♦ ویژگی‌های NEXT – HOP در آگهی جهت محاسبه‌ی پیوند نظیر مورد استفاده.

۳- گزینش مسیر BGP

قبلاً اشاره شد پروتکل BGP با بهره‌گیری از (eBGP) و (iBGP) به توزیع مسیرها به همه‌ی مسیریاب‌های درون سیستم‌های خود مختار می‌پردازد. از طریق این روش توزیع، مسیریاب ممکن است از چندین مسیر به هر پیش‌وند آگاهی حاصل نماید. در این حالت مسیریاب بایستی یکی از مسیرهای ممکن را گزینش نماید. اطلاعات تاثیرگذار در این فرآیند گزینش، مجموعه‌ای از همه‌ی مسیرهای یادگیری و پذیرش شده از سوی مسیریاب می‌باشد. اگر دو و یا چندین مسیر با پیش‌وند مشابهی وجود داشته باشد آنگاه (BGP) به گونه‌ی ترتیبی تا ماندگاری فقط یک مسیر قواعد حذف ذیل را فعال می‌سازد:

- به مسیرها یک ارزش ارجاع محلی به عنوان یکی از ویژگی‌ها اختصاص می‌یابد. ارجاع محلی یک مسیر از سوی یک مسیریاب ممکن است تعریف شده نباشد و یا آن که از سوی یک روتر در همان AS یادگیری شده باشد. این اتخاذ تصمیم برعهده‌ی مدیر شبکه‌ی AS است. معمولاً مسیرهای با بیشترین ارزش ارجاع محلی گزینش می‌شوند.
- از باقیمانده‌ی مسیرها که همگی ارزش ارجاع محلی مشابه دارند، مسیری با کوتاه‌ترین AS – PATH گزینش می‌شود. اگر این شرایط تنها قانون گزینش مسیر باشد آنگاه (BGP) با بهره‌گیری از یک الگوریتم DV شناسایی مسیری را انجام می‌دهد که فاصله‌ی فیزیکی آن مسیر تعداد پرش‌های کوتاه AS در برابر تعداد مسیریاب‌های پرش کوتاه می‌باشد.
- از باقیمانده‌ی مسیرها که همگی ارزش ارجاع محلی و طول AS – PATH مشابه دارند، مسیری در نزدیک‌ترین مسیریاب NEXT – HOP گزینش می‌شود. عبارت نزدیکترین به مفهوم آن است که مسیریاب با کمترین هزینه‌ی مسیر که از سوی الگوریتم درونی AS مشخص می‌شود و آن هزینه از همه کوچکتر است گزینش می‌شود. این همان فرآیند مسیریابی مهمان ناخوانده است که در بخش ۴-۱۸ بررسی شد.
- اگر بیش از یک مسیر هنوز وجود دارد، مسیریاب با بهره‌گیری از شناسه‌های (BGP) مسیر را گزینش می‌کند.

قواعد حذف بیش از آنچه در بالا توصیف شد پیچیده می‌باشد. جهت خودداری از ورود به دام پیچیدگی‌های (BGP) بهینه است که قواعد BGP در مقیاس کوچک و تدریجی مطالعه و تجربه آموزی شود.

- **چگونه اطلاعات آدرس شبکه به جدول گسیل مسیریاب وارد می‌شود؟**

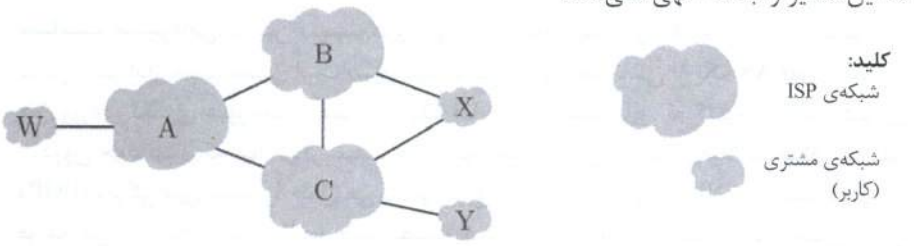
یادآوری این نکته ضروری است که درج اطلاعات آدرس در جدول گسیل روتر شامل یک پیش‌وند مانند: 138.16.64/22 و یک پورت خروجی وابسته به مسیریاب مانند: Port 7 می‌شود. هنگامی که یک بسته وارد مسیریاب می‌شود، آدرس IP مقصد با پیش‌وندهای موجود در جدول گسیل مقایسه می‌شود تا آدرسی یافت شود که با طولانی‌ترین پیش‌وند قابل نگاشت است. پس از آن بسته در درون روتر و به سوی پورت وابسته به پیش‌وند گسیل می‌شود. اکنون به بررسی چگونگی اعمال اطلاعات ورودی: یعنی پیش‌وند و پورت وابسته به درون جدول گسیل می‌پردازیم. مثال ساده‌ی ذیل بسیاری از مطالب توضیح داده شده پیرامون مسیریابی و گسیل را به یکدیگر گره می‌زند. فرض کنید پیش‌وند از نوع **بیگانه** است به عبارت دیگر متعلق به AS روتر نیست بلکه به AS دیگری وابسته است.

برای آنکه یک پیش‌وند وارد جدول گسیل مسیریاب گردد آن روتر بایستی ابتدا از پیش‌وندی آگاه گردد که وابسته به یک زیرشبکه یا انباشتی از زیرشبکه‌ها است. همانگونه که قبلاً توضیح داده شده است یک مسیریاب از طریق آگهی مسیر یک BGP از پیش‌وند یک روتر آگاه می‌شود. چنین آگهی ممکن است از طریق یک تعامل eBGP از یک مسیریاب به AS دیگر و یا از طریق یک تعامل iBGP یعنی از یک روتر در AS مشابه انجام گیرد. پس از آنکه مسیریاب از پیش‌وند آگاه گردید نیازمند تعیین پورت خروجی مناسبی است که داده‌گرام‌ها به سوی یک پیش‌وند از طریق آن گسیل می‌شوند. اطلاعات مربوط به این پورت قبل از پیش‌وند وارد جدول گسیل می‌شود. اگر مسیریاب بیش از یک مسیر آگهی شده را برای این پیش‌وند دریافت دارد آنگاه از فرآیند گزینش مسیر BGP بهره می‌گیرد تا **بهترین** مسیر برای پیش‌وند را شناسایی نماید. در این حالت فرض کنید بهترین مسیر گزینش شده است. همانگونه که قبلاً توضیح داده شده است مسیر انتخاب شده شامل یک ویژگی NEXT – HOP می شود که آدرس IP نخستین روتر خارج از مسیریاب AS در امتداد بهترین مسیر قرار می‌گیرد. بر مبنای مطالب اشاره شده در بالا پس از آن مسیریاب از پروتکل مسیریابی بین AS یعنی OSPF استفاده می‌کند تا کوتاه‌ترین مسیر به روتر NEXT – HOP را شناسایی نماید. مسیریاب سرانجام شماره پورت وابسته به پیش‌وند را از طریق شناسایی نخستین پیوند در امتداد کوتاه‌ترین مسیر شناسایی می‌کند. پس از آن روتر می‌تواند جفت اطلاعات پیش‌وند و شماره پورت را وارد جدول گسیل خود نماید. محاسبه‌ی جدول گسیل از طریق پردازشگر مسیریابی در شکل ۴-۶ انجام می‌شود و سپس به درون کارت‌های پورت ورودی روتر اعمال می‌گردد.

۴- سیاست مسیریابی

در این بخش برخی از مبانی بنیادی سیاست مسیریابی (BGP) بررسی می‌شود. شکل ۴-۴۲ تعداد 6 سیستم خود مختار متصل به یکدیگر را نشان می‌دهد. این سیستم‌ها عبارتند از A ، B ، C و W ، X و Y . این نکته حائز اهمیت است که A ، B ، C و W ، X و Yهمگی سیستم خودمختار هستند به عبارت دیگر مسیریاب نمی‌باشند. اگر فرض شود سیستم‌های خود مختار W ، X و Y شبکه‌های ریشه باشند و A ، B و C شبکه‌های زیرساخت و نیز A ، B و Cهمگی حالت نظیر را با یکدیگر اختیار کنند و

اطلاعات کامل (BGP) را در اختیار شبکه‌های مشتری خود قرار دهند. آنگاه تمامی ترافیک ورودی به **شبکه‌ی ریشه**[1] بایستی فقط به مقصد آن شبکه باشد و همه‌ی ترافیک خروجی از یک شبکه‌ی ریشه بایستی از همان شبکه سرچشمه گرفته باشد. سیستم‌های خودمختار W و Y بنابراین شبکه‌های ریشه می‌باشند. شبکه X اصطلاحاً **شبکه‌ی ریشه‌ی چندخانه**[2] خوانده می‌شود زیرا که به بقیه‌ی شبکه از طریق دو تامین کننده‌ی متفاوت متصل شده است. این وضعیتی است که در عمل هر روز در اینترنت رو به گسترش است. مانند شبکه‌های W و Y سیستم خودمختار X بایستی مبدا و مقصد همه‌ی ترافیک ورودی و یا خروجی به X باشد. پرسش آن است که چگونه رفتار این شبکه‌ی ریشه پیاده‌سازی و تقویت می‌شود؟ چگونه از گسیل ترافیک بین B و C از جانب X ممانعت به عمل می‌آید؟ پاسخ این موارد به سادگی از طریق کنترل روشی که مسیرهای (BGP) آگهی می‌شوند قابل دستیابی است. به ویژه آن که X به عنوان یک شبکه‌ی ریشه عمل می‌کند بنابراین اگر به مجاوران خود یعنی B و C آگهی نماید که دارای مسیری به هیچ یک از مقصدهای دیگر به غیر از خودش نمی‌باشد. به عبارت دیگر اگرچه X ممکن است از وجود مسیری مانند XCY آگاه باشد که سرانجام آن شبکه‌ی Y می‌باشد اما این مسیر را به B آگهی نمی‌دهد.



شکل ۴-۴۲ ♦ یک حالت ساده از عملکرد BGP

از آنجائیکه B از وجود مسیری به Y از سوی X آگاه نمی‌باشد، بنابراین شبکه‌ی B هرگز ترافیک به مقصد Y و یا حتی C را از طریق X گسیل نمی‌کند. این مثال ساده بیانگر چگونگی گزینش آگهی مسیر می‌باشد که می‌تواند روابط مسیریابی مشتری و یا تامین کننده را پیاده‌سازی نماید.

اکنون تمرکز را بر روی تامین کنندگان شکل ۴-۴۲ یعنی سیستم خودمختار B قرار می‌دهیم. اگر فرض شود B از شبکه‌ی A یادگیری نماید که A دارای مسیری مانند AW به شبکه‌ی W است. آنگاه سیستم خودمختار B می‌تواند مسیر BAW را در جدول اطلاعات مسیریابی خود وارد نماید. بدیهی است B همچنین مایل به آگهی مسیر BAW به مشتری خود یعنی X می‌باشد. بنابراین X نیز اگاه می‌شود که می‌تواند مسیردهی اطلاعات را به B از طریق W انجام دهد. اما نکته آن است که آیا B بایستی مسیر BAW را به C آگهی نماید؟ اگر چنین کند آنگاه C قادر خواهد شد که ترافیک را از طریق CBAW به W انتقال دهد. اگر A ، B و C همگی زیرساخت‌های تامین کننده باشند آنگاه B ممکن است این احساس را داشته باشد که می‌تواند از زیر بار مسئولیت و نیز هزینه‌ی حمل ترافیک ترانزیت بین A و C شانه خالی کند. به عبارت دیگر B به درستی تعبیر نماید که این وظیفه‌ی A و C می‌باشد که از مسیردهی C به و یا از طرف مشتریان شبکه‌ی A با بهره‌گیری از یک ارتباط مستقیم بین A و C اطمینان حاصل نماید. در حال حاضر استاندارد تدوین شده‌ای که حاکم بر چگونگی

1- Stub network
2- Multi homed stub network

مسیردهی در بین ISPها باشد وجود ندارد. اما در ISP های تجاری این قانون کلی پذیرفته شده است که هر ترافیک جاری در ISP متعلق به شبکه‌ی زیرساخت بایستی دارای مبدا و مقصد و یا هر دو باشد که مشتری آن ISP به شمار می‌آید در غیر این صورت ترافیک آزادانه بر روی شبکه‌ی ISP جریان می‌یابد. توافق‌های نظیرسازی فردی که پاسخ پرسش‌هایی از قبیل آنچه در بالا عنوان شد را فراهم می‌اورند بین جفت ISP ها و اغلب به صورت محرمانه انجام می‌گیرد. در صورت علاقه به مطالعه بیشتر پیرامون (BGP) و پیش‌وندهای آن خواننده به سامانه‌ی http://www.routeviews.org هدایت می‌شود.

۱- دلایل پروتکل‌های متفاوت مسیریابی درونی و بیرونی (AS)

با توجه به بررسی جزئیات پروتکل‌های مسیریابی درون و بین سیستم‌های خودمختار در اینترنت ممکن است این پرسش بنیادی مطرح شود، چرا دو پروتکل مختلف مسیریابی درون AS و بین آن‌ها مورد نیاز می‌باشد؟ پاسخ به این پرسش در اهمیت تفاوت بین اهداف مسیریابی در درون و ما بین سیستم‌های خودمختار است. این اهداف در قالب مطالب ذیل خلاصه می‌شوند:

- **سیاست مسیریابی.** ما بین سیستم‌های خودمختار موضوعات مربوط به سیاست مسیریابی بسیار تاثیرگذار است. ممکن است بسیار مهم باشد که ترافیک ناشی از یک AS قادر نباشد از AS ویژه‌ی دیگری عبور نماید. همچنین یک AS ممکن است مایل به کنترل ترافیک عبوری از درون خود باشد که بین دیگر سیستم‌های خودمختار حمل می‌نماید. قبلاً ملاحظه شد که (BGP) ویژگی‌های مسیر را حمل می‌کند و توزیع کنترل شده‌ای از اطلاعات مسیریابی را فراهم می‌آورد. انجام این وظایف جهت تصمیم‌گیری مسیریابی برمبنای این اصول کلی پایه‌گذاری شده است. در درون یک AS هر فرآیندی تحت تاثیر مدیریت کنترل مشابه است و بنابراین سیاست‌گذاری در مسیریابی نقش کم‌رنگی در گزینش مسیرهای بین سیستم‌های خودمختار بازی می‌کند.
- **مقیاس.** توان‌مندی الگوریتم مسیریابی و ساختمان داده‌ی آن در مقیاس پذیری جهت اداره‌ی عملیات مسیریابی در درون و مابین تعداد انبوهی از شبکه‌هاموضوع بسیار حیاتی در مسیریابی بین AS می‌باشد. در درون یک AS موضوع مقیاس‌پذیری نگرانی اندکی را به همراه دارد. اگر یک قلمرو مدیریت به تنهایی بسیار بزرگ شود همواره این امکان وجود دارد که آن را به دو AS تقسیم نمود و عملیات مسیریابی بین AS را به این دو سیستم خودمختار حاصل شده گسترش و انجام داد، با توجه به این نکته که پروتکل OSPF اجازه می‌دهد چنین سلسله مراتبی را که تقسیم AS به محدوده‌های مجزا است، پیاده‌سازی نمود.
- **کارایی.** از آنجائیکه مسیریابی خارج از AS گرایش به چگونگی سیاست گزینش مسیر را به دنبال دارد. کارایی و به عبارتی کیفیت مسیرهای بهره‌گیری شده معمولاً نقش دست دوم را بازی می‌کنند. به عبارت دیگر مسیرهای طولانی‌تر و پر هزینه‌تر مطلوب برخی سیاست‌های مسیریابی است و این واقعیت در مقابل هزینه‌ی کمتر و مسیر کوتاه‌تر قرار می‌گیرد. ملاحظه شد که در بین AS ها وابستگی بین هزینه و مسیرها وجود ندارد. در درون یک AS نیز اصول گزینش مسیر از لحاظ سیاست‌گذاری چندان نگران کننده نمی‌باشد. بنابراین می‌توان به این نتیجه رسید که میزان کارایی حاصل از یک مسیر به مراتب اهمیت بیشتری دارد.

- **حضور در اینترنت: اتصال قطعه‌های یک معما**

فرض کنید به‌تازگی یک شبکه‌ی کوچک کامپیوتری را راه‌اندازی نموده‌اید که دارای تعداد اندکی از سرورها شامل یک وب سرور عمومی که محصولات و خدمات شرکت شما را معرفی می‌کند، یک سرور پستی که از طریق آن کارکنان شما پیام‌های ایمیل دریافت می‌دارند و یک سرور DNS است. آشکار است که مایل هستید از سراسر گیتی سامانه‌ی وب شما مورد بازدید قرار گیرد تا همگان با محصولات و خدمات شما آشنا شوند. علاوه بر آن شما تمایل دارید کارکنان را قادر به ارسال و دریافت ایمیل از طریق مشتریان بالقوه در سراسر گیتی نمائید. برای رسیدن به این اهداف ابتدا نیاز به دریافت یک ارتباط اینترنتی می‌باشد که از طریق اشتراک و ارتباط با یک ISP محلی امکان‌پذیر است. شرکت شما بایستی بهره‌مند از یک روتر دریچه باشدکه متصل به یک مسیریاب در ISP محلی می‌شود. یک ارتباط ممکن است یک اتصال DSL از طریق زیرساخت شبکه‌ی تلفنی موجود، یک خط استیجاری به مسیریاب ISP و یا یکی از ده‌ها راهکار دستیابی توصیف شده در فصل ۱ باشد. بدیهی است علاوه بر این مهم ISP محلی شما مجموعه‌ای از آدرس‌های IP را نیز در اختیار شما قرار می‌دهد. فرض کنید این مجموعه آدرس‌های IP در قالب a/24 و شامل 256 آدرس باشند. بی‌درنگ پس از آنکه شما از ارتباط فیزیکی و محدوده آدرس‌های IP بهره‌مند گردید به اختصاص یک آدرس از آدرس‌های مجاز خود به وب سرور، سرور پستی، سرور DNS، مسیریاب دریچه و نیز مابقی آدرس‌ها به دیگر تجهیزات و امکانات شبکه‌ی شرکت می‌پردازید.

علاوه بر عقد قرارداد با ISP محلی، نیاز به امضای قرارداد با یک ثبت‌کننده‌ی اینترنتی به‌منظور دریافت یک نام قلمرو براساس موضوعات توصیف شده در فصل ۲ دارید. به‌عنوان مثال اگر نام شرکت NKE باشد طبیعی است که تلاش می‌کنید نام قلمرو NKE.com را برای آن اخذ نمایید. شرکت شما بایستی در سیستم DNS نیز حضور داشته باشد. از سوی دیگر به دلیل آنکه افراد خارج از شرکت شما مایل به تماس با سرور DNS شما و دریافت آدرس IP سرورهای شبکه‌ی شما هستند بایستی به ثبت‌کننده‌ی قلمرو نام، آدرس IP متعلق به سرور DNS خود را نیز ارایه نمایید. در این هنگام است که ثبت‌کننده‌ی نام قلمرو اقدام به ثبت نام قلمرو و آدرس IP مربوط به سرور DNS در سرورهای قلمرو سطح بالای مربوط به (.com) براساس مطالب فصل ۲ می‌کند. پس از انجام این مراحل هر کاربر که با نام NKE.com آشنا باشد می‌تواند آدرس IP سرور DNS شما را از طریق سیستم DNS اخذ نماید.

برای آنکه مردمان قادر به شناسایی آدرس IP وب سرور باشند بایستی سرور DNS حامل آدرس‌های IP قابل نگاشت با نام میزبان وب سرور یعنی WWW.NKE.com باشد. مشابه همین موارد را برای دیگر سرورهای عمومی مانند سرور پستی و غیره نیاز دارید. درنتیجه اگر فردی مانند A مایل به جست‌وجوی وب سرور شما باشد سیستم DNS در تماس با سرور DNS شرکت شما آدرس IP وب سرور را شناسایی و آن را در اختیار A می‌گذارد. پس از آن A از طریق یک ارتباط TCP مستقیماً به وب سرور شما متصل می‌شود.

اما با همه‌ی این موارد هنوز یک گام لازم و حیاتی دیگر وجود دارد تا افراد خارج از شبکه‌ی شما قادر به دسترسی و تماس با وب سرور شما از سراسر گیتی باشند. تصور کنید اگر A به آدرس IP وب سرور شما آگاهی داشته باشد و اقدام به ارسال یک داده گرام IP مانند یک قطعه‌ی TCP SYN نماید چه اتفاقی روی می‌دهد؟ این داده گرام از درون اینترنت مسیریابی و پس از ملاقات تعدادی مسیریاب در ASهای متفاوت سرانجام به وب سرور شما وارد می‌شود. هنگامی که یکی از این مسیریاب‌ها داده گرام را دریافت می‌دارد به جست‌وجوی یک آدرس IP در جدول گسیل خود می‌پردازد تا پورت خروجی

ارسال داده‌گرام را شناسایی نماید. بنابراین هر یک از مسیریاب‌ها نیاز به آگاهی از وجود پیش‌وند 24/ شرکت شما دارند. چگونه یک روتر از این پیش‌وند آگاه می‌شود؟ همانگونه که در مطالب این فصل ملاحظه شد از طریق BGP پی به این موضوع می‌برد. هنگامی که شرکت شما با یک ISP مشترک می‌شود و از طریق آن محدوده‌ی آدرس IP را به همراه پیش‌وند دریافت می‌دارد، ISP محلی از پروتکل BGP بهره می‌گیرد تا این اطلاعات را به ISPهای دیگر طرف قرارداد خود آگهی نماید. این ISPها در لحظه‌ی دریافت این اطلاعات، از BGP مجدداً بهره می‌گیرند تا اطلاعات اشاره شده را به دیگران انتشار دهند. سرانجام همه‌ی مسیریاب‌های اینترنت از پیش‌وند 24/ شما آگاه می‌شوند و بنابراین امکان گسیل داده‌گرام‌ها به مقصد وب سرور یا سرور پستی فراهم می‌شود.

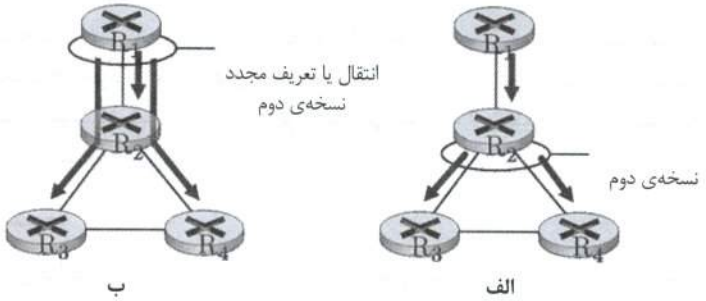
۴-۲۲ مسیریابی فراپخش و چند پخش

در مطالب گذشته پروتکل‌های مسیریابی که ارتباط تک‌پخش یا به عبارت دیگر نقطه به نقطه را پشتیبانی می‌کنند مورد بررسی قرار گرفت. حالتی که در آن یک گره‌ی مبدا بسته‌های اطلاعاتی را به یک گره‌ی مقصد گسیل می‌نمود. در این بخش پروتکل‌های فراپخش و چندپخش تحلیل می‌شوند. در **مسیریابی فراپخش (BR)**[1] لایه‌ی شبکه خدمات تحویل بسته‌های ارسالی از یک گره‌ی مبدا را به همه ی گره‌های دیگر در شبکه بر عهده می‌گیرد.در **مسیریابی چندپخش (MR)**[2] یک گره‌ی مبدا قادر است نسخه ای از بسته‌ها را به زیر مجموعه‌ای از دیگر شبکه گسیل نماید.

• الگوریتم‌های مسیریابی فراپخش

شاید ساده ترین روش انجام ارتباط فراپخش ارسال نسخه‌های مجزا از بسته‌های اطلاعات از سوی گره‌ی فرستنده به هرمقصد بر مبنای شکل ۴-۴۳ (الف) باشد. اگر تعداد گره‌های مقصد N فرض شود آنگاه مسیریاب مبدا کافی است تعداد N نسخه از یک بسته را به مقصدهای گوناگون آدرس‌دهی نموده و سپس N نسخه را به N مقصد با بهره‌گیری از مسیریابی فراپخش گسیل نماید. اصطلاحاً این رویکرد N مسیر تک‌پخش در یک عملیات فراپخش ساده‌ترین از نوع خود می‌باشد. زیرا به پروتکل مسیریابی جدیدی در لایه شبکه ، تکرار بسته و نیز عملیات گسیل بسته نیاز نمی‌باشد. اما این رویکرد در عین حال سادگی نقاط ضعف چشمگیری دارد. نخستین ضعف آن عدم کارآمدی است. اگر گره‌ی مبدا از طریق یک پیوند به بقیه‌ی بخش‌های شبکه متصل شده باشد آنگاه N نسخه‌ی مجزا از یک بسته فقط از این پیوند عبور می‌نماید. بدیهی است چنانچه فقط یک نسخه از بسته بر روی نخستین مسیریاب طرف پیوند ارسال گردد و گره‌ی طرف دیگر پیوند هر نسخه‌ی اضافی دیگر را در صورت نیاز تشکیل و سپس گسیل نماید کارآمدی حاصل چشمگیر خواهد شد. این به مفهوم آن است که اگر گره‌های شبکه به صورت مستقل و بدون نیاز به گره‌ی مبدا نسخه‌های تکراری از یک بسته را خلق نمایند کارآیی به مراتب بیشتر خواهد شد. به عنوان مثال در شکل ۴-۴۳ (ب) فقط یک نسخه از بسته است که پیوند R1–R2 را پویش می‌نماید. آن بسته سپس در R2 نسخه‌برداری مجدد می‌شود و یک نسخه از طریق پیوند R2 – R3 و R2 – R4 گسیل می‌شود.

1- Broad cast routing
2- Multicast routing



شکل ۴-۴۳ ♦ نسخه‌برداری مبدا در برابر نسخه برداری شبکه

ضعف دوم روش تک‌پخش N طرفه بر مبنای این فرض ذاتی است که دریافت کننده‌های فراپخش و آدرس‌های آن‌ها از سوی فرستنده شناسایی شده‌اند. اما چگونه این اطلاعات حاصل شده است؟ احتمالاً پروتکل‌های دیگری مانند **عضویت در فراپخش** یا پروتکل **ثبت مقصد** برای انجام این مهم مورد نیاز می‌باشد. این نیاز خود سرآیند بیشتر را سبب می‌شود و پیچیدگی بیشتری را به پروتکل تحمیل می‌نماید. سرانجام ضعف سوم روش تک‌پخش Nطرفه مربوط به فلسفه‌ی بهره‌گیری از فراپخش است. در بخش ۴-۱۴ ملاحظه شد که پروتکل‌های مسیریابی وضعیت مسیر با بهره‌گیری از فراپخش، اطلاعات وضعیت مسیر را انتشار می‌دهند تا جهت محاسبه‌ی مسیر تک‌پخش یا نقطه به نقطه به کار گرفته شود. بدیهی است در حالت‌هایی که از فراپخش جهت خلق و به روزرسانی مسیرهای تک‌پخش استفاده می‌شود معقول نیست که از زیرساخت مسیریابی تک‌پخش جهت دستیابی به اهداف فراپخش استفاده شود.

با توجه به این نقاط ضعف رویکردهایی که در آن‌ها گره‌های شبکه نقش فعال در نسخه برداری از بسته‌ها، گسیل آن‌ها و محاسبه‌ی مسیرهای فراپخش را شامل شوند از جذابیت بیشتری برخوردار خواهد بود. در ادامه‌ی مطلب چندین رویکرد برمبنای بخش ۴-۵ بررسی می‌شود. الگوی شبکه بر مبنای یک گراف G=(N,E) می‌باشد که در آن هر یال E یک جفت گره از مجموعه‌ی N گره است. از N جهت اشاره به مجموعه‌ای از گره‌ها و نیز اندازه‌ی مجموعه (|N|) استفاده می‌شود.

۱- سیلاب کنترل نشده

یک روش بدیهی برای دستیابی به فراپخش سیلاب می‌باشد. در این روش گره‌ی مبدا نسخه‌ای از بسته را به همه‌ی مجاوران خود گسیل می‌کند. هنگامی که یک گره بسته‌ی فراپخش را دریافت نماید آن را نسخه‌برداری نموده و سپس به همه‌ی مجاوران خود گسیل می‌کند. بدیهی است نسخه‌ی بسته به گره مجاوری که از آن دریافت بسته صورت گرفته است مجدداً گسیل نمی‌شود. آشکار است در صورت کامل بودن گراف در این روش سرانجام نسخه‌ای از بسته‌ی فراپخش به همه‌ی گره‌های گراف تحویل می‌شود. اگر چه این روش ساده و جذاب است اما از این ایراد مهم رنج می‌برد که اگر گراف دارای دوره یا حلقه باشد آنگاه یک و یا چندین نسخه از هر بسته‌ی فراپخش به گونه‌ای نامحدود و بی‌نهایت حلقه گونه تکرار می‌شوند. به عنوان مثال در شکل ۴-۴۳ گره‌ی R2 سیلاب R3 و گره‌ی R3 سیلاب R4 و R4 سیلاب R2 و R2 مجدداً سیلاب R3 و غیره را سبب می‌شوند. این مثال ساده نشان می‌دهد که چگونه فراپخش دو بسته سبب یک دوره‌ی بی‌پایان می‌شود که یکی در جهت عقربه‌ی ساعت و دیگری بر خلاف آن است. علاوه بر نمونه ی اشاره شده موضوع دیگر مربوط به هنگامی است که یک گره با بیش از دو گره‌ی دیگر در ارتباط است. در

این حالت آن گره نسخه‌های چندگانه از بسته‌ی فراپخش را تولید و آن‌ها را گسیل می‌کند هر یک از این نسخه‌ها تعدادی از مانندهای خود را در گره‌های دیگر که بیش از دو مجاور دارند تکثیر می‌نمایند و این عمل ادامه می‌یابد. این رخداد سبب تشکیل یک طوفان فراپخش می‌شود که از تکثیر بی‌پایان بسته‌های فراپخش به وجود می‌آید و سرانجام شبکه را بی‌حاصل می‌سازد.

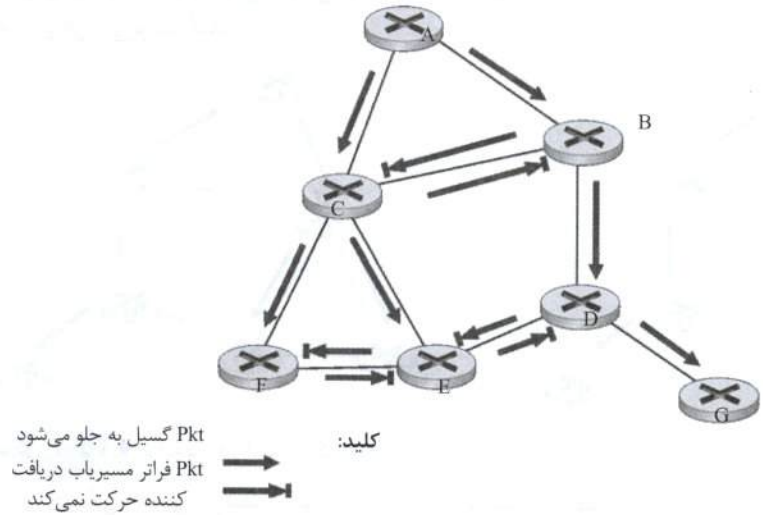
۲- سیلاب کنترل شده

کلید ممانعت از طوفان فراپخش آن است که یک گره با آگاهی زمان سیلابی کردن یک بسته و یا عدم انجام آن را تشخیص دهد. به عبارت دیگر تشخیص دهد آیا نسخه ی قبلی از یک بسته را تا به حال دریافت داشته است و آیا ان را سیلابی نموده است یا خیر؟ در عمل دستیابی به این مهم از طریق روش‌های مختلف امکان‌پذیر است. در ادامه این رویکردها تشریح می‌شوند. یکی از این رویکردها **شماره ردیف سیلاب کنترل شده**[1] (SNCF) می‌باشد. در این حالت یک گره‌ی مبدا آدرس خویش و یا شناسه‌ی انحصاری دیگری را به همراه یک **شماره ردیف فراپخش** در بسته‌ی فراپخش قرار می‌دهد. سپس بسته را به همه‌ی مجاورانش گسیل می‌کند. هر گره صورتی از آدرس مبدا و شماره ردیف هر بسته‌ی فراپخش دریافتی را نسخه‌برداری و به جلو گسیل می‌کند. هنگامی که یک گره بسته‌ی فراپخش را دریافت نماید ابتدا بسته را از لحاظ حضور آن در فهرست خود سنجش می‌کند. اگر بسته در فهرست موجود باشد ان را حذف می‌کند در غیر این صورت بسته نسخه برداری شده و به همه‌ی گره‌های مجاور به جز گره‌ای که بسته را قبلاً ارسال نموده است گسیل می‌شود. پروتکل **گونتلا**[2] که در فصل ۲ به آن اشاره شد از شماره ردیف (SNCF) جهت فراپخش درخواست‌ها بر روی شبکه‌ی خود بهره می‌گیرد. در نیوتلا نسخه‌برداری پیام و گسیل به جلو در لایه‌ی کاربرد در عوض لایه‌ی شبکه صورت می‌پذیرد.

رویکرد دوم سیلاب کنترل شده **مسیر معکوس به جلو**[3] (RPF) می‌باشد که به آن فراپخش مسیر معکوس (RPB) نیز گفته می‌شود. آنچه در پس پرده‌ی (RPF) است ساده و در عین حال دارای ظرافت است. هنگامی که یک مسیریاب بسته‌ی فراپخش را دریافت می‌کند که همراه آن آدرس مبدا نیز می‌باشدآن بسته را به همه‌ی پیوندهای خروجی خود به جز پیوندی که آن بسته را از طریق آن دریافت نموده است در صورتی گسیل می‌کند که بسته وارد پیوندی شده باشد که بر روی کوتاه‌ترین مسیر نقطه به نقطه‌ای است که حالت بازگشت به مبدا را دارد. در غیر این صورت مسیریاب از بسته‌ی ورودی صرف‌نظر می‌نماید. بدون آنکه آن را به هر یک از پیوندهای خروجی خود گسیل نماید. صرف‌نظر نمودن از چنین بسته‌ای به علت آن است که مسیریاب می‌داند نسخه‌ای از بسته را بر روی کوتاه‌ترین مسیر خود که بازگشت به فرستنده می‌کند یا دریافت خواهد نمود و یا انکه در حال حاضر دریافت نموده است. بدیهی است RPF از روش مسیریابی تک پخش یا نقطه به نقطه جهت تحویل یک بسته به مقصد استفاده نمی‌کند و نیازمند آگاهی یک مسیریاب از کوتاه‌ترین مسیر از خود به مبدا نیز نمی‌باشد. تنها موضوعی که RPF نیاز به اگاهی آن دارد مجاور بعدی بر روی کوتاه‌ترین مسیر نقطه به نقطه به فرستنده می‌باشد. چنین جستجویی از طریق شناسه‌ی مجاور خود فقط برای تعیین این نکته است که آیا یک بسته‌ی فراپخش دریافتی را بایستی سیلابی نماید یا خیر؟

1- Sequence number controlled flooding
2- Guntella
3- reverse path forwarding

شکل ۴-۴۴ پیاده سازی RPF را نشان می‌دهد. اگر فرض شود پیوندهای با خطوط ضخیم بیان کننده‌ی مسیرهای با حداقل هزینه از گیرنده‌ها به مبدا (A) باشند. گره‌ی (A) در آغاز بسته‌ی مبدا (A) را به گره‌های (B) و (C) فراپخش می‌کند. گره‌ی B بسته‌ی مبدا A را که از A به دلیل آنکه کوتاه‌ترین مسیر به A بوده است دریافت نموده است به هر دو گره‌ی C و D گسیل می‌کند. در این حالت گره‌ی B بدون انجام عملیات گسیل به جلو هر بسته‌ی با مبدا A را که از دیگر گره‌ها مانند C یا D دریافت نموده است حذف و یا از آن‌ها صرف‌نظر می‌کند. اکنون اگر گره‌ی C در نظر گرفته شود که بسته‌ای را از مبدا A مستقیماً از طریق A و B دریافت نماید. از آنجائیکه B در کوتاه‌ترین مسیر C حالت بازگشت به A را ندارد، گره‌ی C هر بسته‌ی مبدا A را که از B دریافت نماید حذف می‌کند. از طرف دیگر هنگامی‌که C بسته ی از مبدا A را مستقیماً از گره‌ی A دریافت نماید ان را به سوی گره‌های B ، E و F گسیل می‌کند.



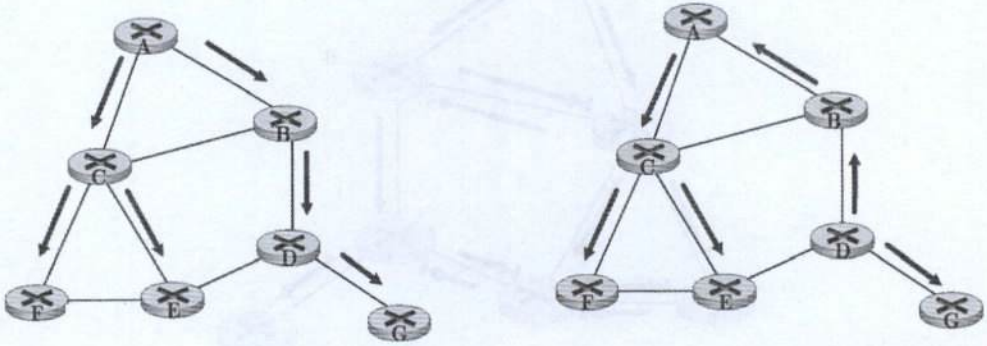
شکل ۴-۴۴ ♦ گسیل مسیر معکوس

۳- فراپخش درخت پوشا

در حالیکه دو روش (SNCF) و (RPF) مانع از طوفان فراپخش می‌شوند اما به طور کامل از انتقال بسته‌های فراپخش افزونه جلوگیری نمی‌کنند. به عنوان مثال در شکل ۴-۴۵ گره‌های B ، C ، D ، E و F یک یا دو بسته‌ی اضافه را دریافت می‌دارند. مطلوب آن است که هر گره فقط نسخه‌ای از بسته‌ی فراپخش را دریافت نماید. با بررسی درخت شکل ۴-۴۵ (الف) که گره‌های آن از طریق خطوط ضخیم به یکدیگر متصل شده‌اند ملاحظه می‌شود که اگر بسته‌های فراپخش فقط در امتداد پیوندهای درون همین درخت گسیل به جلو شوند. هر یک و تمامی گره‌های شبکه فقط یک نسخه از بسته‌ی فراپخش را دریافت می‌دارند. این همان راه حل مطلوب است. این درخت نمونه‌ای از یک درخت پوشا می‌باشد. درختی که شامل هر یک و تمامی گره‌های یک گراف است. یک درخت پوشا از یک گراف G=(N,E) گراف دیگری است در قالب $G' = (N', E')$ که در آن E' زیر مجموعه‌ای از E است و G' شامل

دوره و یا حلقه نمی‌باشد و همه‌ی گره‌های G را در درون خود جای داده است. اگر هر پیوند هزینه‌ای را وابسته به خود داشته باشد و هزینه‌ی درخت مجموع هزینه‌ی پیوندها باشد آنگاه یک درخت پوشا که هزینه‌ی آن حداقل همه‌ی هزینه‌های گراف درخت پوشا باشد را اصطلاحاً **درخت پوشای حداقل**[1] (MST) نامیده می‌شود.

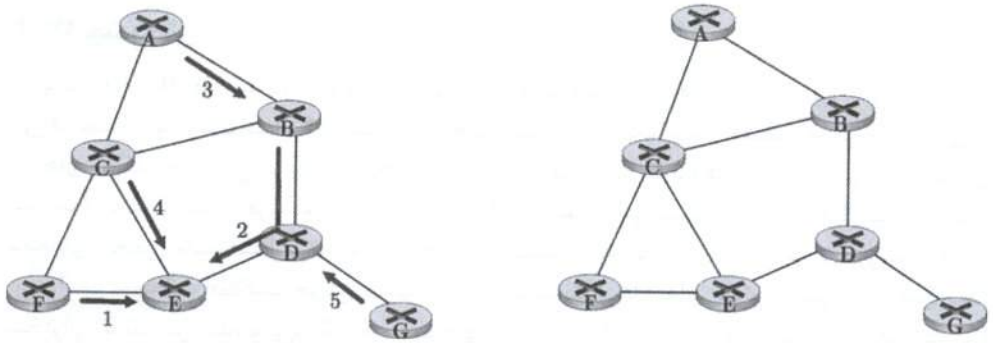
رویکرد دیگر تامین فراپخش آن است که گره‌های شبکه نخست یک درخت پوشا را تشکیل دهند. هنگامی که یک گره‌ی مبدا مایل به ارسال یک بسته‌ی فراپخش است آن گره بسته را به همه‌ی پیوندهای متعلق به درخت پوشا گسیل می‌کند. گره‌ای که یک بسته‌ی فراپخش را دریافت می‌دارد آن را به همه‌ی مجاوران خود در درخت پوشا به غیر از مجاوری که بسته را از آن دریافت نموده است گسیل می‌کند. درخت پوشا نه تنها بسته‌های فراپخش اضافی را حذف می‌کند بلکه از درخت پوشا می‌توان به وسیله‌ی هر گره جهت آغاز یک فراپخش بهره گرفت. دو شکل ۴-۴۵ (الف) و (ب) این موارد را نشان می‌دهند. بدیهی است یک گره نیازمند آگاهی از اطلاعات تمام درخت نمی‌باشد. تنها موردی که نیازمند به آگاهی آن می‌باشد این نکته است که کدام یک از مجاورانش در G درخت پوشای همسایگان می‌تواند باشد.



شکل ۴-۴۵ ♦ فراپخش در امتداد یک درخت پوشا

پیچیدگی اساسی مرتبط با رویکرد درخت پوشا خلق و نگهداری آن می‌باشد. از میان ده‌ها الگوریتم درخت پوشای پیشنهاد شده فقط یکی از ساده‌ترین آن‌ها در این بخش تحلیل می‌شود. در رویکرد بر **مبنای ریشه**، ساختمان یک درخت پوشا از طریق تعریف یک گره مرکزی به نام **نقطه‌ی تلاقی**[2] و یا **هسته** آغاز می‌شود. گره‌ها آنگاه به گونه‌ی نقطه به نقطه پیام‌ها را درخت گونه به گره‌ی هسته گسیل می‌کنند. یک پیام درخت گونه با بهره‌گیری از مسیریابی نقطه به نقطه به سوی مرکز هدایت می‌شود تا آنکه وارد گره‌ای شود که متعلق به درخت پوشا است و یا آنکه گره‌ی هسته باشد. در هر یک از دو حالت مسیری که این درخت دنبال نموده است شاخه‌ای از درخت پوشا بین گره و یالی است که آغازگر پیام درخت گونه و هسته است. این مسیر جدید را می‌توان حالت گراف درخت پوشای موجود به شمار آورد.

1- Minimum spanning tree
2- Rendezvous point



شکل ۴-۴۶ ♦ ساختار بر مبنای ریشه در درخت پوشا

شکل ۴-۴۶ مراحل ساخت یک درخت پوشا بر مبنای ریشه را نشان می‌دهد. اگر فرض شود گزینش گره‌ی E به عنوان مرکز یاریشه‌ی درخت باشد و گره‌ی F نخستین موردی است که به درخت می‌پیوندد و یک پیام درخت گونه را به E گسیل می‌کند. پیوند EF نخستین درخت پوشا به شمار می‌آید. پس از آن گره‌ی B از طریق ارسال پیام درخت گونه‌ی خود E به درخت پوشا می‌پیوندد. اگر مسیر نقطه به نقطه‌ی از B به E از طریق D انجام شود آنگاه پیام درخت گونه‌ای در مسیر BDE حاصل می‌شود که بر روی درخت پوشا گراف می‌شود. پس از آن گره‌ی A به گروه پوشا از طریق پیام خود به E می‌پیوندد. اگر مسیر نقطه به نقطه‌ی A به E از درون B عبور نماید آنگاه از آنجائیکه B به درخت پوشا پیوسته است ورود پیام درخت گونه‌ی A در گره‌ی B نتیجه‌اش پیوند AB است، که بی‌درنگ بر روی درخت پوشا گراف می‌شود. گره‌ی C پس از آن از طریق گسیل پیام مستقیماً به E به درخت پوشا می‌پیوندد و سرانجام از آنجائیکه مسیریابی نقطه به نقطه‌ی G به B بایستی از طریق D انجام شود، هنگامی که G پیام به E می‌فرستند پیوند GD بر روی درخت پوشا در گره‌ی D گراف می‌شود.

۴- الگوریتم‌های فراپخش در عمل

از پروتکل‌های فراپخش در عمل در هر دو لایه‌ی کاربرد و شبکه استفاده می‌شود. پروتکل نیوتلا از فراپخش لایه‌ی کاربرد جهت انتقال درخواست در بین نظیرهای خود بهره می‌گیرد. در این حالت یک پیوند بین دو فرآیند نظیر در لایه‌ی کاربرد شبکه‌ی نیوتلا در حقیقت یک ارتباط TCP است. نیوتلا از یک ساختار ویژه‌ی شماره ردیف سیلاب کنترل شده (SNCF) بهره می‌گیرد که در آن یک شناسه‌گر 16 بیت و یک توصیف گر محموله‌ی 16 بیت که نوع پیام نیوتلا را شناسایی می‌کند جهت شناسایی این که یک درخواست فراپخش تحویل شده قبلاً دریافت، نسخه‌برداری و یا گسیل شده است، استفاده می‌کند. علاوه بر آن نیوتلا از یک محدوده‌ی زمان خروج (TTL) نیز جهت محدود نمودن تعداد پرش‌های کوتاه بر روی یک درخواست سیلابی که بایستی به جلو گسیل گردد، بهره می‌گیرد. هنگامی که یک فرآیند نیوتلا دریافت و نسخه‌برداری یک تقاضا را انجام می‌دهد، زمان TTL را قبل از گسیل تقاضا کاهش می‌دهد. بنابراین یک تقاضای نیوتلا به صورت سیلابی فقط به نظیرهایی تحویل می‌شود که در محدوده‌ی یک عدد تعریف شده قرار دارد. یعنی ارزش اولیه‌ی TTL پرش کوتاه در لایه‌ی کاربرد که از آغاز تقاضا مقداردهی شده است. فرآیند سیلاب نیوتلا بنابراین عملکرد اصطلاحاً **سیلاب قلمرو و محدود**[1] نامیده می‌شود.

1- Limited scope flooding

۴-۲۳ چندپخش

در بخش قبل ملاحظه شد در سرویس فراپخش بسته‌ها به همه‌ی گره‌های شبکه تحویل می‌گردید. در این قسمت از موضوعات فصل ۴ فرایند چندپخش که در آن یک بسته فقط به زیرمجموعه‌ای از گره‌های شبکه تحویل می‌شود مورد توجه قرار می‌گیرد. تعدادی از کاربردهای شبکه نیازمند تحویل بسته‌ها از یک یا چندین فرستنده به گروهی از دریافت کننده‌ها می‌باشند. این کاربردها شامل انتقال داده‌های حجمی مانند انتقال ارتقاءهای نرم‌افزاری از تولید کننده به کاربرانی که این به روزرسانی‌ها را درخواست نموده‌اند، اطلاعات جریان پیوسته‌ی داده مانند انتقال فایل‌های صوتی ، تصویری و متن یک سخنرانی یا تدریس زنده و بی‌درنگ به مجموعه‌ای از حاضران، اشتراک داده‌ها مانند کاربرد همایش راه دور (teleconference) و یا تخته سفید (with board) ، تامین پیوسته‌ی داده مانند اطلاعات بازارهای بورس ، به روزرسانی نهان وب، بازی‌های تعامل‌گرا و بی‌درنگ مانند محیط‌های مجازی تعاملی توزیع شده و یا بازی‌های چند بازیگری و نظایر آن می‌باشند.

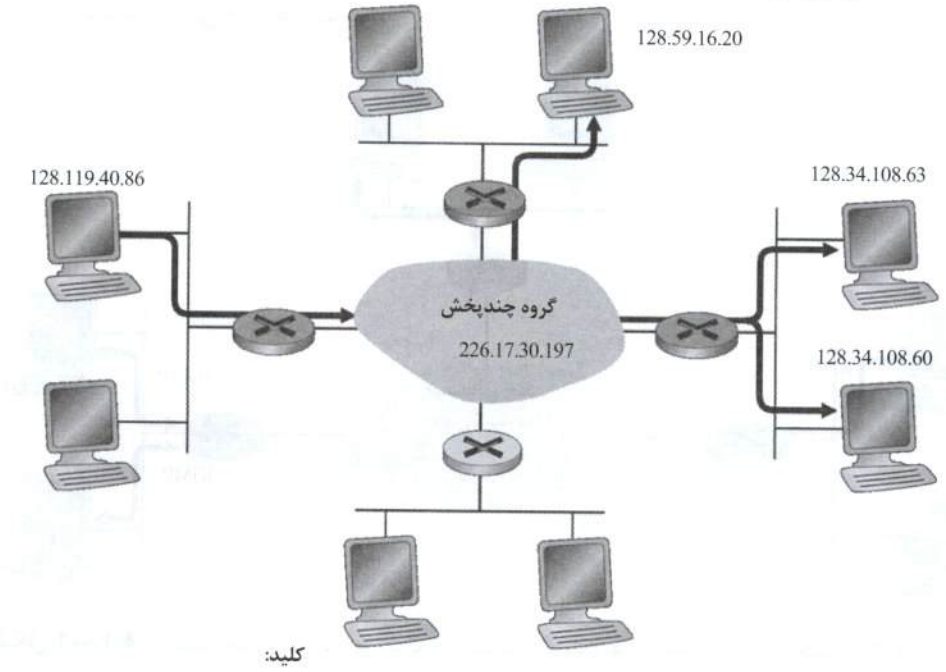
در ارتباط چندپخش بی‌درنگ دو چالش وجود دارد. (۱) چگونه گیرنده‌های بسته‌های چندپخش شناسایی می‌شوند؟ و (۲) چگونه گسیل بسته‌ها به این گیرنده‌ها آدرس‌دهی و یا هدایت می‌شوند؟ در حالت ارتباط تک‌پخش یا نقطه به نقطه آدرس IP گیرنده از سوی هر یک از داده‌گرام‌های IP حمل ودر نتیجه‌گیرنده‌ی تک شناسایی می‌شود. در حالت فراپخش همه‌ی گره‌های دریافت‌کننده‌ی بسته این گونه می‌باشند و بنابراین آدرس‌های مقصد نیاز نمی‌باشند. در حالت چندپخش چندین گیرنده‌ی اطلاعات وجود دارند. آیا معقول است هر بسته‌ی چندپخش آدرس‌های IP متعلق به همه‌ی گیرنده‌ها را حمل نماید؟ این رویکرد ممکن است با تعداد اندکی دریافت‌کننده امکان‌پذیر باشد اما در مورد صدها و یا هزاران دریافت‌کننده حجم اطلاعات آدرس‌دهی در داده‌گرام بر حجم داده‌ی واقعی که هدف انتقال می‌باشند و در دامنه‌ی محموله‌ی داده‌گرام قرار می‌گیرند بسیار چشمگیر خواهد بود. شناسایی انحصاری دریافت‌کنندگان از سوی فرستنده نیازمند این واقعیت است که ارسال کننده از شناسه و آدرس‌های همه‌ی دریافت کنندگان آگاه باشد. مواردی وجود دارد که این نیازمندی‌ها ممکن است لازم نباشد.

به این دلایل در معماری اینترنت و دیگر شبکه‌ها مانند ATM یک بسته با ویژگی چندپخش با بهره‌گیری از **آدرس غیرمستقیم**[1] تعریف می‌شود. به عبارت دیگر یک شناسه‌گر فقط جهت یک گروه دریافت کننده مورد استفاده قرار می‌گیرد و نسخه‌ای از بسته که به آن گروه آدرس‌دهی شده است با بهره‌گیری از این شناسه تحویل همه‌ی گیرنده‌های چندپخش مرتبط با آن گروه می‌شوند. در شبکه‌ی اینترنت شناسه‌ی تک که گروهی از گیرنده‌ها را نمایندگی می‌کند آدرس IP چندپخش کلاس D نامیده شده است. گروه گیرنده‌های وابسته به یک آدرس کلاس D اصطلاحاً **گروه چندپخش**[2] خوانده می‌شود. شکل ۴-۴۷ نمایش خلاصه شده‌ی گروه چندپخش است. در شکل تعداد 4 میزبان که با رنگ سایه‌دار مشخص شده‌اند وابسته به یک گروه آدرس چندپخش 226.17.30.197 می‌باشند و داده‌گرام‌های آدرس‌دهی شده به این شناسه‌ی چندپخش را دریافت می‌کنند. چالش موجود که هنوز بایستی مورد توجه قرار گیرد این حقیقت است که هر میزبان دارای یک آدرس تک‌پخش IP انحصاری است که کاملاً مستقل از آدرس گروه چندپخش است که در آن شرکت نموده است.

پرسش‌های دیگری که در شکل ۴-۴۷ ظاهر می‌شوند آن است که چگو.نه یک گروه آغاز می‌شود و

1- Address indirection
2- Multicase group

چگونه خاتمه پیدا می‌کند؟ چگونه آدرس گروه گزینش می‌شود؟ چگونه میزبان‌های جدید مانند فرستنده‌ها و یا گیرنده‌ها به گروه اضافه می‌شوند؟ آیا هر میزبانی می‌تواند به گروه جهت ارسال و یا دریافت اضافه شود؟ آیا پیوستن به گروه محدودیت دارد؟ اگر آری چه کسی این محدودیت را تعریف می‌کند؟ آیا اعضای گروه از شناسه‌های دیگر اعضای گروه به عنوان بخشی از پروتکل لایه‌ی شبکه آگاهی دارند؟ چگونه گره‌های شبکه با یکدیگر تعامل می‌کنند تا داده‌گرام چندپخش تحویل همه‌ی اعضای گروه گردد؟ همه‌ی این پرسش‌ها برای شبکه‌ی اینترنت در قالب یک پروتکل به نام **پروتکل مدیریت گروهی اینترنت (IGMP)**[1] در مستند [RFC 3376] پاسخگویی شده است. در ادامه پروتکل IGMP تحلیل می‌شود.



شکل ۴-۴۷ ♦ گروه چندپخش : یک داده‌گرام به گروهی تحویل می‌شود که اعضای گروه چندپخش می‌باشند.

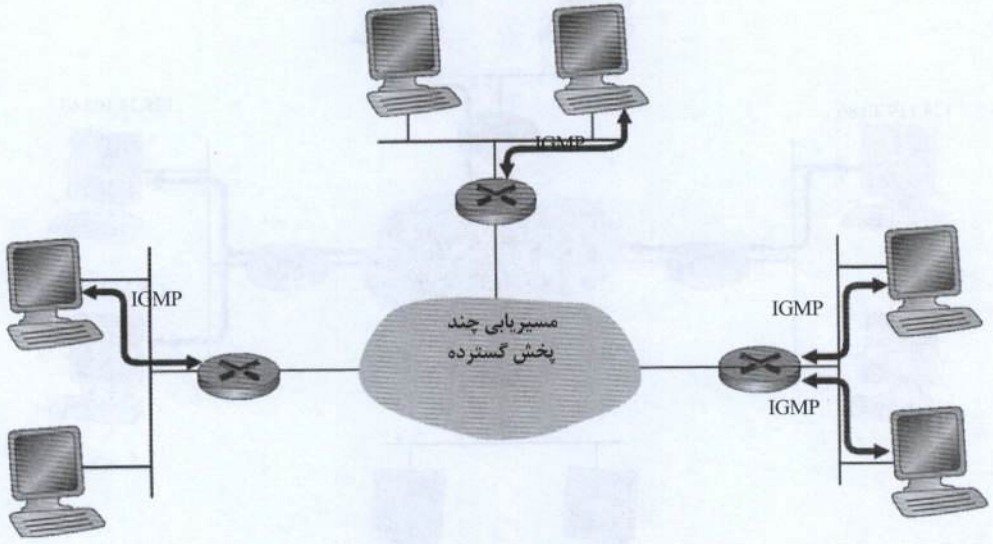
۱- پروتکل مدیریت گروهی اینترنت (IGMP)

پروتکل (IGMP) نسخه‌ی سوم، بین یک میزبان و مسیریاب مستقیماً متصل به آن عمل می‌کند. روتر مستقیماً متصل به میزبان را می‌توان نختیسن پرش کوتاه به شمار آورد که از طریق آن میزبان قادر به رویت مسیری به هر یک از میزبان‌های خارج از شبکه‌ی محلی خود می‌باشد و یا ممکن است آخرین مسیریاب پرش کوتاه به هر مسیر آن میزبان باشد. شکل ۴-۴۸ سه مسیریاب چندپخش را که هر کدام

1- Internet group management protocol

به میزبان‌های خویش از طریق یک واسط محلی خروجی متصل شده‌اند نشان می‌دهد. این واسط محلی در مثال شکل ۴-۴۸ به یک LAN متصل است و هر شبکه‌ی محلی چندین میزبان متصل به خود را دارد که تعدادی از آن‌ها در هر لحظه از زمان متعلق به یک گروه چند پخش می‌باشند.

پروتکل IGMP امکانات لازم برای یک میزبان به منظور آگاهی مسیریاب متصل به آن را فراهم می‌آورد تا از این طریق مشخص نماید یک کاربرد در حال اجرا بر روی میزبان قصد پیوستن به یک گروه ویژه‌ی چندپخش را دارد. با توجه به آنکه دامنه‌ی تعامل IGMP به میزبان و مسیریاب متصل به آن محدود می‌شود آشکار است که پروتکل دیگری جهت هماهنگ‌سازی مسیریاب‌های چندپخش که شامل روترهای متصل در سراسر اینترنت می‌شود مورد نیاز است تا داده‌گرام‌های چندپخش از طریق آن به مقصدهای نهایی خود هدایت شوند.



شکل ۴-۴۸ ♦ دو عنصر چندپخش لایه‌ی شبکه در اینترنت : پروتکل‌های مسیریابی چند پخش و (IGMP)

عملکرد اشاره شده‌ی اخیر از طریق الگوریتم‌های مسیریابی چندپخش لایه‌ی شبکه امکان‌پذیر می‌شود. لایه‌ی شبکه‌ی چندپخش اینترنت بنابراین شامل دو عنصر مکمل یکدیگر می‌باشد این دو عبارتند از IGMP و پروتکل‌های مسیریابی چندپخش. پروتکل IGMP تنها محدود به سه نوع می‌شود: مانند ICMP پیام‌های پروتکل IGMP در درون یک داده‌گرام IP قالب‌بندی می‌شوند که شماره پروتکل IP آن 2 می‌باشد. پیامی در قالب membership_query از سوی یک مسیریاب به همه‌ی میزبان‌های متصل به واسط مانند همه‌ی میزبان‌های موجود بر روی شبکه‌ی محلی گسیل می‌شود تا مجموعه‌ای از همه‌ی گروه‌های چندپخش که از سوی میزبان‌ها به واسط متصل شده‌اند را شناسایی نماید. میزبان‌ها به پیام membership_query با یک پیام پاسخ IGMP به صورت : membership_report پاسخ می‌دهند. نوع دوم پیام‌های membership_query نیز می‌تواند از سوی یک میزبان هنگامی تولید شود که کاربردی نخست به یک گروه چندپخش بدون آنکه انتظار دریافت چنین پیامی را از مسیریاب داشته باشد به جمع گروه چندپخش وارد می‌شود. نوع سوم پیام‌های IGMP در قالب پیام leave_group می‌باشد. اگرچه این پیام حالت اختیاری دارد اما چگونه یک مسیریاب میزبانی را که از گروه چندپخش

خارج می‌شود شناسایی می‌کند؟ پاسخ این پرسش آن است که اگر میزبان واکنش در قبال پیام membership_query با گروه آدرس تعریف شده‌ای را از خود نشان ندهد آنگاه مسیریاب به این نتیجه می‌رسد که آن میزبان دیگر در گروه چندپخش حضور ندارد. این حالت اصطلاحاً **وضعیت نرم**[1] در یک پروتکل اینترنت نامیده می‌شود. در یک پروتکل وضعیت نرم حالت یا وضعیت تعریف شده از طریق رخداد زمان خروج حذف می‌شود چنانچه به گونه‌ی انحصاری بازسازی نگردد. در این شکل از (IGMP) میزبان‌های پیوسته به یک گروه چندپخش حذف می‌شوند و زمان حذف آن‌ها نیز بر مبنای ارسال متناوب پیام membership_query از سوی مسیریابی است که از سوی پیام ارسالی میزبان متصل به آن در قالب membership_report پاسخگویی نمی‌شود. باور کلی بر این است که پروتکل وضعیت نرم کنترل ساده‌تری را در مقابل پروتکل‌های وضعیت سخت دنبال می‌کند، زیرا پروتکل‌های وضعیت سخت نه تنها نیاز به فرآیند حذف و یا اضافه‌ی حالت دارند بلکه روش‌هایی را جهت بازیابی موقعیتی نیاز دارند که عامل مسئول حذف حالت به صورت دایمی خاتمه یافته است و یا آنکه دچار عیب شده باشد.

۲- الگوریتم‌های مسیریابی چندپخشی

دشواری بنیادی مسیریابی چندپخش در شکل ۴-۴۹ نشان داده شده است. میزبان‌های پیوسته به گروه چندپخش به صورت سایه‌دار نشان داده شده‌اند. مسیریاب بی‌درنگ متصل به آن‌ها نیز در شکل سایه‌دار نشان داده شده است. در شکل ۴-۴۹ فقط زیرمجموعه‌ای از مسیریاب‌ها نیازمند دریافت ترافیک چندپخش می‌باشند. بر مبنای شکل فقط مسیریاب‌های A ، B ، E و F ترافیک چندپخش را دریافت می‌کنند زیرا هیچ یک از میزبان‌های متصل به مسیریاب D به گروه چندپخش تعلق ندارند. از آنجائیکه میزبانی وابسته به مسیریاب C نمی‌باشد روترهای C و یا D نیز نیازی به دریافت ترافیک گروه چندپخش ندارند. هدف مسیریابی چندپخش فقط یافتن یک درخت از پیوندهایی است که همه ی مسیریاب‌های گروه چندپخش که میزبان‌های متصل به خود دارند را مشخص نماید. بسته‌های چندپخش سپس در امتداد این درخت مسیریابی از فرستنده به همه‌ی میزبان‌های متعلق به آن درخت انتشار چندپخش می‌شوند. البته ممکن است درخت شامل مسیریاب‌هایی باشد که میزبان‌های وابسته به گروه چندپخش را شامل نگردد. در شکل ۴-۴۹ به عنوان مثال ارتباط با مسیریاب‌های A ، B ، E و F در یک درخت، بدون وارد نمودن مسیریاب‌های C یا D امکان‌پذیر نمی‌باشد.

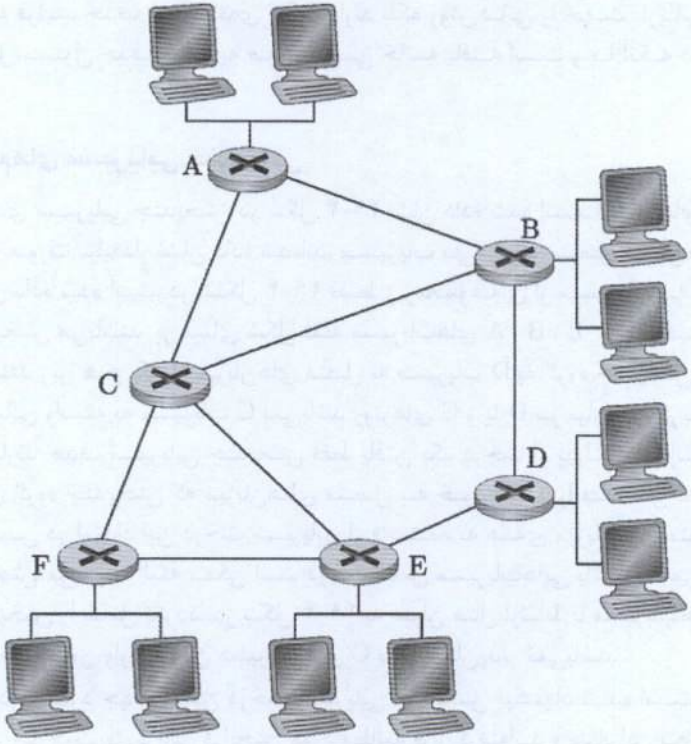
در عمل دو رویکرد جهت تعیین درخت مسیریابی چندپخش پیشنهاد شده است. هر دو رویکرد پیشنهادی در چارچوب مسیریابی فراپخش مورد مطالعه قرار گرفته‌اند و بنابراین به صورت سطحی آن‌ها بررسی مجدد می‌شوند. تفاوت بین دو رویکرد بر این اساس است که آیا از یک درخت گروهی مشترک جهت توزیع ترافیک همه‌ی فرستنده‌های گروه می‌توان بهره گرفت و یا آنکه یک درخت مسیریابی مبدا ویژه برای هر فرستنده‌ی منفرد بایستی ساخته شود.

• مسیریابی چندپخش با بهره‌گیری از یک درخت گروهی مشترک

مانند حالت درخت پوشای فراپخش، مسیریابی چندپخش بر روی یک درخت گروهی مشترک بر مبنای ساختمان درختی پایه‌ریزی می‌شود که همه‌ی مسیریاب‌های مرزی یا نبشی آن با میزبان‌های متصل متعلق به گروه چندپخش می‌باشند. در عمل از رویکرد بر مبنای هسته جهت ساختن درخت مسیریابی

1- Soft state

استفاده می‌شود که در آن مسیریاب‌های نبش به همراه میزبان‌های متصل آن‌ها متعلق به گروه چندپخش فرستنده‌ای می‌باشند که از طریق یک ارتباط نقطه به نقطه پیام‌های دعوت به گروه را به مسیریاب هسته آدرس‌دهی می‌کند. مانند حالت فراپخش پیام دعوت گسیل شده از مسیریابی نقطه به نقطه به سوی روتر هسته هدایت می‌شود تا ورود آن به یک مسیریاب که متعلق به درخت چندپخش است و یا مسیریاب هسته‌ای که از آن بهره می‌گیرد. همه‌ی روترهای در امتداد مسیر که پیام دعوت آن‌ها را دنبال می‌کند بسته‌های چندپخش دریافتی را به مسیریاب‌های نبش که آغازگر دعوت چندپخش بوده‌اند، گسیل می‌کنند. پرسش حیاتی در رابطه با درخت مسیریابی چندپخش بر مبنای هسته ی فرآیند مورد استفاده جهت گزینش هسته است. این الگوریتم‌ها در مستندهای (IGMP) بررسی شده‌اند.

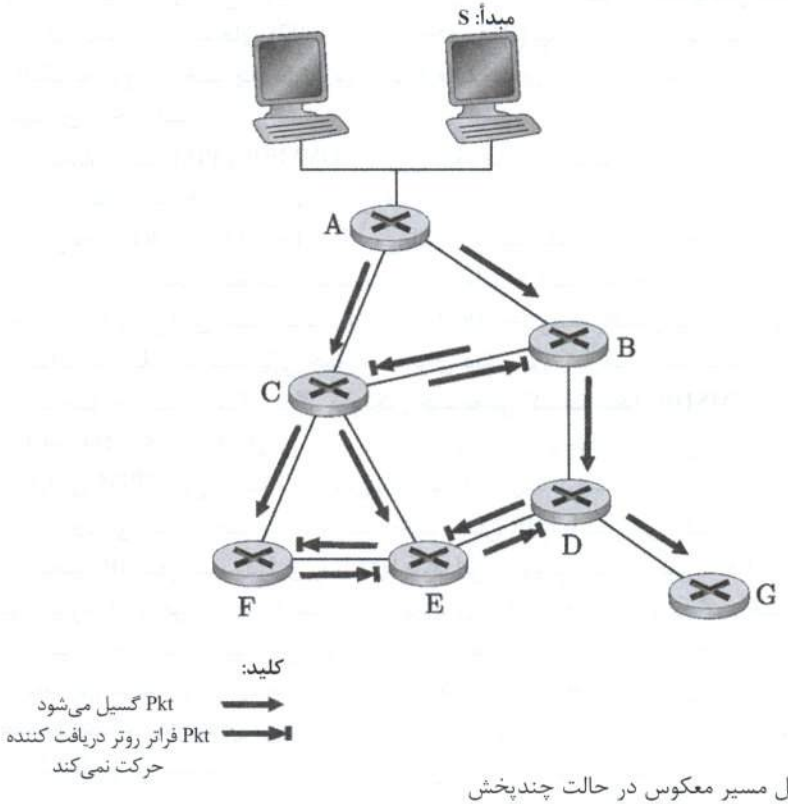


شکل ۴-۴۹ ♦ میزبان‌های چندپخش، مسیریاب‌های متصل به آن‌ها و دیگر روترها

- **مسیریابی چندپخش با بهره‌گیری از درخت مسیریابی مبدا**

در حالیکه درخت مسیریابی گروهی مشترک چندپخش یک درخت مشترک را تعریف می‌کندکه مسیریابی بسته‌ها را از همه‌ی فرستنده‌ها مسیردهی می‌کند. رویکرد دوم، درخت مسیریابی چندپخش را برای هر مبدا در گروه چندپخش انتشار می‌دهد. در عمل از الگوریتم (RPF) فراپخش با گره‌ی مبدا مانند x جهت ساختن یک درخت چندپخش استفاده می‌شود که انتقال بسته‌های از این نوع را که از مبدا x سرچشمه گرفته‌اند انتشار می‌دهد. الگوریتم فراپخش RPF که قبلاً مطالعه شد به اندکی تغییرات جهت کاربرد در فرآیند چندپخش نیازمند است. لزوم این تغییرات را می‌توان از طریق مثالی

که در ادامه توضیح داده می‌شود درک نمود. مسیریاب D در شکل ۴-۵۰ را در نظر بگیرید. در حالت فراپخش با بهره‌گیری از RPF این مسیریاب بسته‌ها را به G هدایت می‌کند اگرچه G میزبانی متصل به خود ندارد که به گروه چند پخش پیوسته باشد. این موضوع در این مقطع چندان نگران کننده نیست زیرا D فقط یک مسیریاب جریان یعنی G دارد. حال اگر تعداد روترهای جاری به هزاران بالغ شود هر یک از آن‌ها بسته‌های چندپخش را دریافت می‌کنند که مورد نیاز نمی‌باشند. راه حل چالش دریافت بسته‌های چندپخش ناخواسته از طریق RPF اصطلاحاً **هَرَس**[1] می‌باشد. یک مسیریاب چندپخش که بسته‌هایی از این نوع را دریافت نموده است و دارای میزبان‌های متصل به خود است که وابستگی به گروه چندپخش ندارد، یک پیام هرس را به سمت مسیریاب بالاتر خود گسیل می‌کند و آنگاه با دریافت پیام هرس از سوی هر یک از روترهای پایین، این مسیریاب پیام هرس دریافتی را به جریان بالا دست خود گسیل می‌کند.



شکل ۴-۵۰ ♦ گسیل مسیر معکوس در حالت چندپخش

۳- مسیریابی چندپخش در اینترنت

نخستین پروتکل مسیریابی چندپخش که در اینترنت مورد استفاده قرار گرفت **پروتکل مسیریابی چندپخش بردار فاصله (DVMRP)** بر اساس مستند [RFC 1075] بود. پرتکل DVMRP درخت بر مبنای مبدا را با رویکرد هرس و گسیل مسیر معکوس پیاده‌سازی می‌کند. این پروتکل با بهره‌گیری از یک الگوریتم RPF به همراه هرس که در بالا به آن اشاره رفت عمل می‌نماید. پروتکل با کاربرد به

1- Prunning

مراتب گسترده‌تر در اینترنت پروتکل مسیریابی چندپخش به نام **پروتکل مستقل (PIM)** می‌باشد که دو نوع توزیع چندپخش را تعریف می‌کند. **(۱) روش فشرده** و **(۲) روش پراکنده**. در روش فشرده که در مستند [RFC 3973] تشریح شده است اعضای گروه چندپخش حالت فشرده دارند. به عبارت دیگر اغلب و یا بسیاری از مسیریاب‌ها در هر قلمرو نیازمند وارد شدن به عملیات مسیریابی چندپخش داده‌گرام‌ها می‌باشند. پروتکل PIM در حالت فشرده یک روش گسیل به جلو به صورت مسیر معکوس سیلاب و هرس می‌باشد که بسیار مشابه فلسفه‌ی پروتکل DVMRP است

در حالت دوم که معروف به روش پراکنده است و در مستند [RFC 4601] تشریح شده است تعداد مسیریاب‌های مرتبط با اعضای گروه فراپخش با توجه به مجموع تعداد روترها اندک است و اعضای گروه در سطح گسترده‌ای پراکنده شده‌اند. روش پراکنده‌ی PIM از نقاط تلافی جهت ایجاد درخت توزیع چندپخش بهره می‌گیرد. در رویکرد دیگری که از پروتکل **چندپخش مبدا ویژه (SSM)**[1] توصیف شده در مستندهای [RFC 4607 , RFC 3569] بهره می‌گیرد تنها یک فرستنده مجاز به گسیل ترافیک به درون درخت چندپخش می‌شود که نتیجه‌ی آن ساده شدن قابل ملاحظه‌ی ساختار و نگهداری درخت است.

هنگامی که PIM و DVMRP در درون یک قلمرو استفاده می‌شوند مدیر شبکه می‌تواند مسیریاب‌های چندپخش را در درون آن قلمرو پیکربندی نماید. این عمل مشابه انجام عملیات برای پروتکل‌های RIP ، OSPF و IS– IS [2] می‌باشد. اگرمسیریاب‌های چندپخش بین قلمروهای مختلف مورد نیاز باشند آنگاه وضعیت چگونه است؟ آیا معادل فرایند چندپخش برای پروتکل بین قلمرو iBGP وجود دارد؟ پاسخ آری است. مستند [RFC 4271] نوع توسعه یافته‌ی پروتکل چندگانه BGP را تعریف می‌نماید که اطلاعات مسیریابی حمل شده برای دیگر پروتکل‌ها را که شامل اطلاعات چندپخش است حمل می‌کند. پروتکل دیگر **پروتکل چندپخش کشف مبدا (MSDP)**[3] نام دارد که دو مستند [RFC 4611 , RFC 3618] آن را تعریف نموده‌اند. از این پروتکل می‌توان به منظور ارتباط نقاط تلاقی به گونه‌ی PIM از نوع پراکنده در قلمروهای گوناگون بهره گرفت.

جمع‌بندی مطالب فصل و نیز موضوع چند پخش IP را با بیان این نکته به انتها می‌بریم که فرایند چندپخش IP هنوز مسیر درازی را تا انتها دارد. این سرویس هنوز به صورت فراگیر در شبکه‌ی اینترنت مورد بهره‌برداری قرار نگرفته است اگرچه نهادهای رسانه‌ای قدرتمند و بزرگی چون BBC به صورت آزمایشی از خدمات آن بهره می‌گیرند. بخش وسیعی از پروتکل‌های چندپخش بایستی در لایه‌ی شبکه پیاده‌سازی شوند. آیا در آینده پیاده‌سازی پروتکل‌های چندپخش در لایه‌ی شبکه و به ویژه در بخش هسته‌ی شبکه پیاده‌سازی می‌شوند و یا آنکه در لایه‌ی کاربرد و به ویژه در نبش شبکه موضوعی است که طراحان و برنامه‌نویسان به آن بایستی پاسخ دهند. در کوتاه مدت با توجه به پیاده‌سازی پروتکل‌های چندپخش در لایه‌ی کاربرد و نیز پروتکل نظیر به نظیر می‌توان این نتیجه را گرفت که فرایند چندپخش ممکن است در لایه‌ی کاربرد به جای لایه‌ی،شبکه پیاده‌سازی گردد.

1- Sourse specific multicast
2- Intermediate to intermediste systems
3- Discovery protocol

تمرین‌های فصل چهارم

۱- دو عملکرد مهم لایه‌ی شبکه در یک شبکه‌ی داده‌گرام چیست؟ سه عملکرد مهم لایه‌ی شبکه در شبکه‌ی مدار مجازی چیست؟

۲- چه تفاوتی مابین مسیریابی و گسیل به جلو می‌باشد؟

۳- آیا مسیریاب‌های هر دو شبکه‌ی داده‌گرام و مدار مجازی از جدول گسیل به جلو استفاده می‌کنند؟ اگر پاسخ آری است جداول گسیل به جلو را برای هر دو کلاس شبکه توضیح دهید.

۴- توصیف کنید که چگونه فقدان بسته در پورت‌های ورودی رخ می‌دهد؟ توضیح دهید چگونه از فقدان بسته در پورت‌های ورودی بدون استفاده از بافرهای نامحدود می‌توان جلوگیری نمود؟

۵- سه نوع ساختار راهیابی در مطالب فصل ۴ بررسی شد هر یک از آن‌ها را به اختصار توصیف نمائید.

۶- توضیح دهید چرا هر پورت ورودی در یک مسیریاب سریع نسخه‌ی سایه‌ای از جدول گسیل به جلو را نگهداری می‌کند؟

۷- مفهوم انسداد HOL چیست؟ در کدام نقطه پورت ورودی و یا خروجی روی می‌دهد؟

۸- فقدان بسته چگونه در پورت‌های خروجی روی می‌دهد؟

۹- آیا مسیریاب‌ها آدرس IP دارند؟ اگر آری تعداد آن‌ها چقدر است؟

۱۰- فرض کنید میزبان A قطعه‌ی TCP کپسول‌بندی شده‌ای در قالب یک داده گرام IP را به میزبان B ارسال می‌کند. هنگامی که B داده‌گرام را دریافت می‌کند چگونه لایه‌ی شبکه در میزبان B می‌داند که قطعه یا همان محموله‌ی داده‌گرام بایستی به TCP و نه UDP و یا مورد دیگری تحویل گردد؟

۱۱- به یک میزبان DHCP مراجعه کنید و آدرس‌های IP ، پوشش شبکه، روتر پیش فرض و آدرس IP سرور DNS محلی آن‌ها را مشاهده و فهرست آن‌ها را بنویسید.

۱۲- گفته می‌شود هنگام تونل‌زنی IPV6 به درون مسیریاب‌های IPV4 انجام می‌شود، پروتکل IPV6 تونل‌های IPV4 را در قالب پروتکل لایه‌ی پیوند می‌بیند آیا شما با این گفته موافق هستید؟ چرا؟

۱۳- معادل باینری آدرس IP در قالب 223.1.3.27 چیست؟

۱۴- تحلیل و مقایسه‌ای را در مورد محدوده‌های سرپیام IPV4 و IPV6 انجام دهید. آیا این محدوده‌ها نقاط مشترکی با یکدیگر دارند؟ توضیح دهید.

۱۵- لازم است که هر سیستم خودمختار از الگوریتم مسیریابی درون AS استفاده نماید؟ علت آن چیست؟

۱۶- مقایسه ای را بین الگوریتم‌های مسیریابی بردار فاصله و و ضعیت پیوند انجام دهید؟

۱۷- توضیح دهید چگونه سلسله مراتب موجود در اینترنت امکان مقیاس‌پذیری میلیون‌ها کاربر را فراهم کرده است؟

۱۸- علت استفاده از روترهای درونی و بیرونی AS در اینترنت چیست؟

۱۹- با در نظر گرفتن شکل ۴-۳۵ اگر آغاز جدول از D باشد و گره‌ی D از A اطلاعات دریافت نماید که فهرست آگهی آن بنابر جدول ذیل باشد آیا جدول در D تغییر می‌کند؟ اگر آری چگونه؟

زیرشبکه‌ی مقصد	مسیریاب بعدی	تعداد پرش‌های کوتاه به مقصد
Z	C	10
W	-	1
X	-	1
- - -	- - -	- - -

۲۰- توضیح دهید چگونه یک مدیر شبکه در یک ISP سطح بالاتر می‌تواند سیاست‌گذاری شبکه را هنگام پیکربندی BGP پیاده‌سازی نماید؟

۲۱- تحلیل و مقایسه‌ای را بین آگهی‌های RIP و OSPF انجام دهید.

۲۲- مفاهیم و تفاوت‌های هر یک از عبارات: زیرشبکه، پیش‌وند و مسیر BGP را توضیح دهید.

۲۳- چگونه BGP از ویژگی NEXT – HOP بهره‌گیری می‌کند؟ چگونه از AS – PATH استفاده می‌کند؟

۲۴- هنگامی که یک میزبان به گروه چندپخش می‌پیوندد آیا آدرس IP آن که به ادرس گروه چندپخش پیوسته است تغییر می‌کند؟ توضیح دهید.

۲۵- چه تفاوت‌هایی بین درخت گروهی مشترک و درخت بر مبنای مبدا در چارچوب مسیریابی چندپخش وجود دارد؟ توضیح دهید.

۲۶- نقش پروتکل IGMP و یک پروتکل مسیریابی چندپخش در قلمرو گسترده چیست؟

۲۷- با توجه به شبکه‌ی مدار مجازی (VC) اگر شماره‌ی VC یک دامنه‌ی 16 بیت باشد:

الف) حداکثر تعداد مدارهای مجازی قابل حمل بر روی یک پیوند چقدر است؟

ب) اگر شماره‌های VCمختلف در هر پیوند در امتداد مسیر VC مجاز باشد. هنگام ایجاد ارتباط پس از تعریف مسیر انتها به انتها توضیح دهید چگونه پیوندها شماره‌های VC خود را گزینش می‌کنند و نیز چگونه پیکربندی جداول گسیل به جلو را در روش غیر متمرکز بدون تکیه بر گره‌ی مرکزی انجام می‌دهند؟

ج) اگر گره‌ی مرکزی مسیرها و شماره‌های VC را در هنگام ایجاد ارتباط مشخص نماید و اگر شماره VC مشابهی برای هر پیوند در امتداد مسیر VC به کار گرفته شود چگونه گره‌ی مرکزی ممکن است شماره VC را در این حالت محاسبه نماید؟ آیا امکان دارد که تعداد VCهای کمتری در مقایسه با حداکثر تعریف شده در بخش (الف) در جریان باشد در عین حال که شماره VC مشترک و آزاد وجود نداشته باشد؟

۲۸- در مطالب فصل از عبارت‌های اتصال‌گرا و ارتباط جهت توصیف به ترتیب خدمات لایه‌های

انتقال و شبکه استفاده شد. چه تفاوتی میان این دو وجود دارد؟

۲۹- در بخش ۴-۴ ملاحظه شد می‌توان از وجود صف در ورودی جلوگیری نمود مشروط بر آنکه ساختار راهیابی n برابر سریع‌تر از نرخ پیوند ورودی باشد. اگر n تعداد پیوندهای ورودی با سرعت های برابر باشند توضیح دهید عدم تشکیل صف چگونه امکان‌پذیر است؟

۳۰- یک شبکه داده‌گرام که از آدرس‌های میزبان 32 بیت بهره می‌گیرد را در نظر بگیرید. اگر مسیریابی در این شبکه دارای تعداد 4 پیوند با شماره‌های از صفر تا 3 باشد و بسته‌های گسیل شده به واسط‌های پیوند به گونه‌ی ذیل باشند:

محدوده‌ی آدرس مقصد	واسط پیوند
1110 0000 0000 0000 0000 0000 0000 0000 تا 1110 0000 1111 1111 1111 1111 1111 1111	0
1110 0001 0000 0000 0000 0000 0000 0000 تا 1110 0001 0000 0000 1111 1111 1111 1111	1
1110 0001 0000 0001 0000 0000 0000 0000 تا 1110 0001 1111 1111 1111 1111 1111 1111	2
غیره	3

الف) جدول مسیریابی به جلو که دارای 5 ورودی و بهره‌گیری از طولانی‌ترین نگاشت پیش‌وند و گسیل بسته‌ها به واسط‌های صحیح پیوند تشکیل می‌شود را نشان دهید.

ب) توضیح دهید چگونه جدول گسیل به جلو که شما تشکیل داده‌اید واسط پیوند مناسب برای داده‌گرام‌های با آدرس‌های مقصد ذیل را فراهم می‌آورد:

1100 1000 1001 0001 0101 0001 0101 0101
1110 0001 0000 0000 1100 0011 0011 1100
1110 0001 1000 0000 0001 0001 0111 0111

۳۱- مسیریابی را در نظر بگیرید که سه زیرشبکه 1، 2 و 3 را به یکدیگر ارتباط می‌دهد. اگر همه‌ی واسط‌ها در هر یک از این سه زیر شبکه دارای پیش‌وند 223.1.17/24 باشند و زیرشبکه‌ی 1 تعداد 125 واسط و زیرشبکه‌های 2 و 3 هر کدام 60 واسط را پشتیبانی کنند. سه آدرس شبکه‌ی مورد نیاز این شرایط را در قالب a.b.c.d/x به دست آورید.

۳۲- شبکه‌ی داده‌گرام با 8 بیت آدرس میزبان را در نظر بگیرید. اگر مسیریاب از طولانی‌ترین پیشوند نگاشت و نیز جدول گسیل به جلو در ذیل بهره گیرد:

پیش‌وند نگاشت	واسط
1	0
11	1

111	2
غیره	3

برای هر 4 واسط محدوده‌ی آدرس‌های وابسته به میزبان مقصد و نیز تعداد آدرس‌های در محدوده را تعریف کنید.

۳۳- زیر شبکه‌ای با پیش‌وند 101.101.101.64/26 را در نظر بگیرید. مثالی از یک آدرس IP که می‌توان به این شبکه اختصاص داد را بنویسید. اگر یک ISP عهده‌دار توده آدرس 101.101.128/17 باشد و مایل به تشکیل 4 زیرشبکه از این توده باشد که هر توده‌ی تشکیل شده آدرس‌های IP مشابه را اختیار نمایند چه پیش‌وندهایی در قالب a.b.c.d/x برای 4 زیر شبکه بایستی تعریف شود؟

۳۴- تشکیلات شبکه‌ی شکل ۲۲-۴ را در نظر بگیرید اگر ISP آدرس 126.13.89.67 را به مسیریاب شبکه اختصاص دهد و آدرس شبکه‌ی خانگی 192.168/16 باشد آنگاه:

الف) آدرس‌های اختصاص یافته به همه‌ی واسط‌های شبکه‌ی خانگی را فهرست نمائید.

ب) اگر فرض شود هر میزبان شامل دو ارتباط TCP و همگی به پورت 80 در میزبان 128.119.40.86 باشند مطلوب است تعداد 6 ورودی مربوط به جدول ترجمه‌ی NAT در این حالت.

۳۵- اگر ارسال یک داده‌گرام به طول 3000 بایت در یک پیوند با مقدار MTU برابر 500 بایت مورد نظر باشد و داده‌گرام اصلی با شماره شناسه‌ی 422 مشخص شده باشد چه تعداد قطعه بایستی تولید گردد؟ مشخصات هر یک از آن‌ها چیست؟

۳۶- فرض کنید داده‌گرام‌های یک شبکه به همراه سرپیام آن‌ها محدود به طول 1500 بایت بین میزبان‌های A و مقصد B شده‌اند. اگر فرض شود طول سرپیام IP برابر 20 بایت است. چه تعداد داده‌گرام لازم است یک فایل MP3 به حجم 3 میلیون بایت ارسال گردد؟

۳۷- با توجه به شکل ۴-۲۷ مسیرهای از v تا y را که شامل حلقه نمی‌باشند شماره‌گذاری نمائید.

۳۸- پرسش ۳۷ را برای مسیرهای از x به w ، w به u و z به x تکرار کنید.

۳۹- ساختار سه گره‌ای شکل ۴-۳۰ را در نظر بگیرید هزینه ی مسیرهای شکل به ترتیب برابر است با: c(x,y)=5 ، c(y,z)=6 و c(z,x)=2 مطلوب است محاسبه‌ی جدول فاصله پس از مقداردهی اولیه و بعد از تکرار الگوریتم بردار فاصله به صورت عملکرد همزمان بر مبنای مطالب تشریح شده در فصل در مورد شکل ۴-۳۰.

۴۰- در شکل ۴-۴۲ اطلاعات مسیر شبکه‌های ریشه‌ی W ، X و Y را در نظر بگیرید. بر مبنای اطلاعات موجود در W و X توپولوژی شبکه چگونه خواهد بود؟

۴۱- با توجه به عملیات الگوریتم مسیر معکوس (RPF) در شکل ۴-۴۵ و با بهره‌گیری از توپولوژی مشابه مجموعه‌ای از مسیرهایی که از همه‌ی گره‌ها به A می‌رسند را مشخص کنید به طوری که اگر این مسیرها دارای حداقل هزینه باشند آنگاه B نسخه‌ای از پیام فراپخش A را از طریق گره‌های A ، C و D با هدایت RPF دریافت نمائید.



فصل ۵

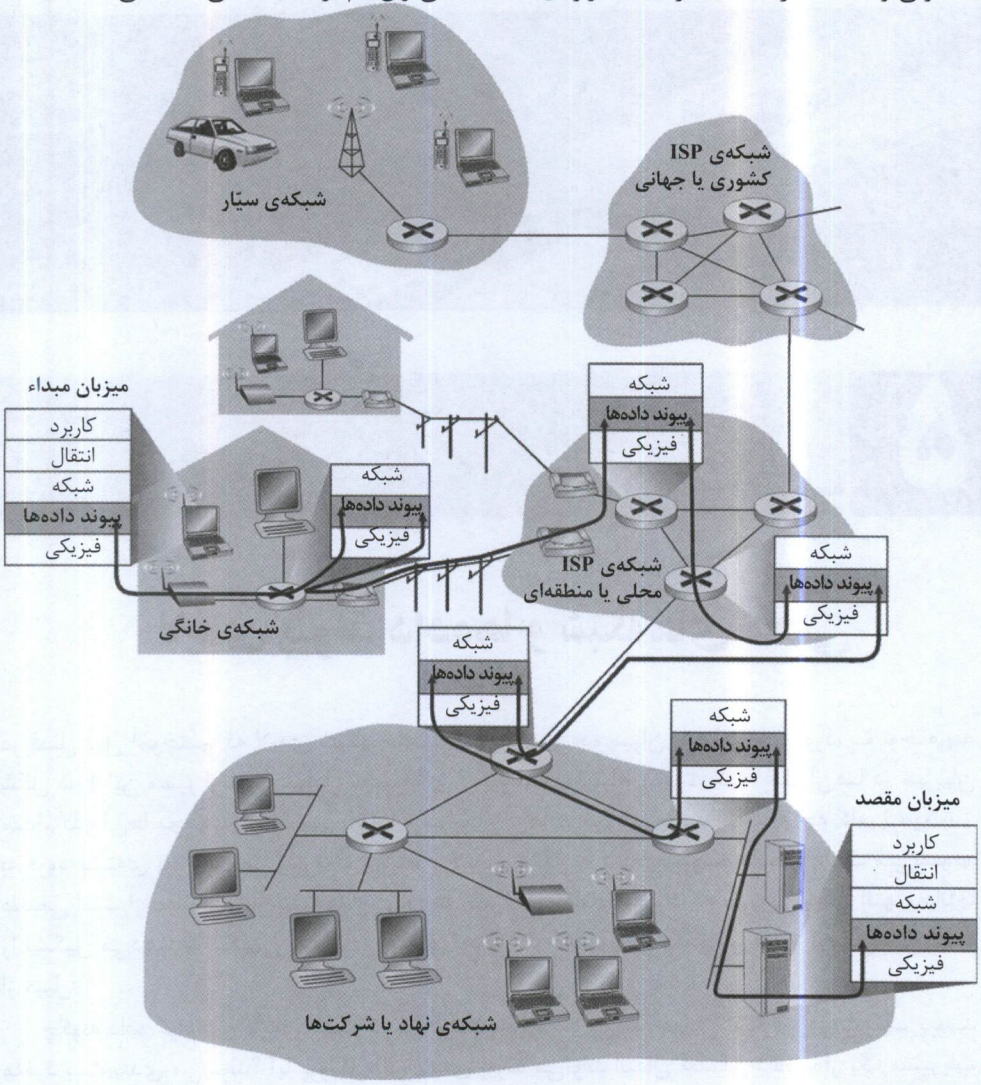
لایه‌ی پیوند داده‌ها و شبکه‌های محلی

در فصل قبل آموختیم که لایه‌ی شبکه خدمات ارتباطی بین دو میزبان را فراهم می‌آورد. با توجه به شکل ۵-۱ این مسیر ارتباطی شامل مجموعه‌ای از پیوندهای ارتباطی است که آغاز آن‌ها در میزبان مبدأ، گذر آن‌ها مجموعه‌ای از مسیریاب‌ها و سرانجام آن‌ها در میزبان مقصد است. هرچه گام، نزدیک‌تر به فرود پشته‌ی پروتکلی می‌گذاریم و به‌ویژه از لایه‌ی شبکه به لایه‌ی پیوند داده، نزدیک می‌شویم طبیعی است از لحاظ گسیل بسته‌ها از درون هر یک از پیوندهای منفرد که مسیر ارتباطی انتها به انتها را تشکیل می‌دهند با شگفتی بیش‌تری روبه‌رو گردیم. سرچشمه این شگفتی‌ها ظهور پرسش‌هایی است از قبیل:

چگونه داده گرام‌های لایه‌ی شبکه در قالب قاب‌های لایه‌ی پیوند جهت انتقال از درون یک پیوند منفرد بسته‌بندی می‌شوند؟ آیا پروتکل‌های لایه‌ی پیوند می‌تواند انتقال مطمئن داده را از یک مسیریاب به مسیریاب دیگر تأمین نماید؟ آیا از پروتکل‌های گوناگون لایه‌ی پیوند می‌توان در پیوندهای گوناگون در امتداد یک مسیر ارتباطی بهره گرفت؟ پاسخ به همه‌ی این پرسش‌های مهم و حتی پرسمان‌های بیش‌تر در این زمینه در این فصل پاسخگویی می‌شوند.

در چارچوب بحث لایه‌ی پیوند دوگونه‌ی مختلف و بنیادی از کانال‌های لایه‌ی پیوند را می‌توان شناسایی نمود. نوع نخست شامل کانال‌های فراپخش است که کاربرد آن‌ها در شبکه‌های محلی (LANS) بدون سیم محلی (WLANS)، شبکه‌های ماهواره‌ای و شبکه‌های دستیابی ترکیبی نوری و هم‌محور (HFC) متداول می‌باشند. در یک کانال فراپخش تعداد قابل توجهی از میزبان‌ها به یک کانال ارتباطی مشترک می‌پیوندند و نیاز به یک پروتکل دستیابی رسانه جهت هماهنگ‌سازی انتقال و جلوگیری از برخورد قابل‌ها (فریم‌ها) با یکدیگر می‌باشد. گونه‌ی دوم کانال‌های پیوند داده عبارتند از پیوندهای ارتباطی نقطه به نقطه. از پیوندهای ارتباطی بین دو رونر یا بین یک مودم شماره‌گیر

مسکونی و یک مسیریاب ISP از جمله مواردی است که می‌توان نام برد. هماهنگی دستیابی به یک



شکل ۵-۱ ♦ لایه‌ی پیوند داده‌ها

پیوند نقطه به نقطه بسیار ساده می‌باشد اما هنوز هم موارد بسیار مهمی پیرامون چگونگی تبدیل داده گرام‌ها به قاب‌ها، انتقال مطمئن داده‌ها، آشکارسازی خطا و کنترل جریان فریم‌ها وجود دارند که نیازمند پاسخگویی و تجزیه و تحلیل می‌باشند.

چندین فنآوری مهم لایه‌ی پیوند در این فصل بررسی می‌شوند. نگاه عمیقی به موضوع **اترنت**[1] که یکی از مهم‌ترین و کاربردی‌ترین پروتکل‌های فنآوری شبکه‌های محلی کابلی است انداخته می‌شود. دیگر موضوع مطالعه‌ی فصل پروتکل نقطه به نقطه (PPP) است. این پروتکل گزینش شده‌ی

1- Ethernet

میزبان‌های شماره‌گیری محلی[1] مسکونی است. اگرچه WiFi و شبکه‌های بدون سیم محلی (WLAN) نیز از موارد پوشش یافته‌ی لایه‌ی پیوند به‌شمار می‌آیند اما بررسی آن‌ها به فصل ۶ که موضوع آن بررسی انحصاری شبکه‌های بدون سیم و سیار است موکول می‌شود.

۵-۱ لایه‌ی پیوند: معرفی و خدمات

قبل از ورود به موارد مهم موضوع فصل واژگانی چند معرفی می‌شوند. در سراسر این فصل میزبان‌ها و مسیریاب‌ها با عنوان **گره‌ها** مورد اشاره قرار می‌گیرند زیرا در این فصل حساسیت چندانی براین مهم که آیا یک گره مسیریاب است و یا یک میزبان وجود ندارد. در این فصل هم چنین کانال‌های ارتباطی که گره‌های مجاور را در امتداد یک مسیر ارتباطی به یکدیگر متصل می‌کنند **پیوند** نامیده می‌شوند. بنابراین برای آن‌که یک داده گرام از میزبان مبدأ به میزبان مقصد انتقال یابد آن داده گرام بایستی از طریق هریک از پیوندهای منفرد در دو انتهای مسیر جابه‌جا گردد. در یک پیوند یک گره‌ی انتقال دهنده، داده گرام را در قالب فریم‌ها بسته‌بندی می‌کند و قاب را به درون پیوند انتقال می‌دهد. گره‌ی دریافت کننده پس از دریافت قاب‌ها داده گرام‌ها را از آن‌ها تجزیه می‌کند.

یک پروتکل لایه‌ی پیوند جهت جابه‌جایی داده گرام‌ها برروی پیوندها مورد استفاده قرار می‌گیرد. **پروتکل لایه پیوند** قالب بسته‌ها که بایستی بین گره‌ها در انتهای پیوندها مبادله شوند را تعریف می‌کند. علاوه برمورد اشاره شده این پروتکل عملیات مورد نیاز گره‌ها را به هنگام ارسال و دریافت بسته‌ها نیز تعریف و مشخص می‌کند. از مطالب فصل یک این نکته یادآوری می‌شود که عناصر داده که از طریق لایه‌ی پیوند داده‌اه جابه‌جا می‌شوند **قاب**[2] خوانده می‌شوند، و هر قاب لایه‌ی پیوند یک داده گرام لایه‌ی شبکه را در درون خود بسته‌بندی می‌کند. عملیات انجام شده از سوی پروتکل لایه‌ی پیوند به هنگام ارسال و یا دریافت فریم‌ها شامل آشکارسازی خطا، انتقال مجدد، کنترل جریان و دستیابی تصادفی می‌شود. مثال‌هایی از پروتکل‌های لایه‌ی پیوند شامل اترنت، 802.11(WLAN) یا همان WiFi، شبکه‌ی حلقوی و پروتکل نقطه به نقطه (PPP) می‌باشند.

لایه‌ی شبکه وظیفه‌ی جابه‌جایی انتها به انتهای قطعه‌های لایه‌ی انتقال را از میزبان مبدأ به میزبان‌های مقصد برعهده دارد در صورتیکه یک پروتکل لایه‌ی پیوند وظیفه‌ی ساده‌تر انتقال گره به گره را در رابطه با داده گرام‌های لایه‌ی شبکه برروی یک **پیوند منفرد** در مسیر برعهده دارد. یکی از مهم‌ترین ویژگی‌های لایه‌ی پیوند آن است که داده‌گرام‌ها ممکن است از سوی پروتکل‌های گوناگون در لایه‌ی پیوند برروی پیوندهای مختلف در مسیر حمل گردند. به‌عنوان مثال یک داده گرام ممکن است از سوی پروتکل اترنت برروی نخستین پیوند و پروتکل PPP برروی آخرین پیوند حمل گردد در حالیکه پروتکل لایه‌ی پیوند WAN در پیوندهای واسط بین این دو پیوند ابتدا و انتها عمل نموده است. بدیهی است خدمات ارایه شده از سوی پروتکل‌های مختلف لایه‌ی پیوند در امتداد یک مسیر انتها به انتها ممکن است گوناگون نیز باشند. به‌عنوان مثال برخی از پروتکل‌های لایه‌ی پیوند تحویل مطمئن داده را شامل می‌شوند در صورتیکه برخی دیگر این ویژگی را ندارند. بنابرین لایه‌ی شبکه بایستی قادر به تأمین مسئولیت‌های انتها به انتها در برابر مجموعه‌ای از خدمات غیرهمگن لایه‌ی پیوند باشد.

جهت درک بهتر لایه‌ی پیوند و نیز چگونگی ارتباط آن با لایه‌ی شبکه، یک عملیات ترابری در قلب

1- Dialup residential hosts
2- Frame

یک مثال تشریح می‌شود. یک آژانس مسافرتی را در ذهن خود تجسم نمائید و فرض کنید یک جهانگرد قصد مسافرت از شهر آکسفورد انگلستان به شهر ناکازاکی ژاپن را دارد. این آژانس به جهانگرد مورد اشاره پیشنهاد می‌کند که برنامه‌ی سفر وی را به این گونه برنامه‌ریزی می‌نماید که وی با یک خودروی تشریفاتی از آکسفورد به فرودگاه بین‌المللی هیثرو در لندن سفر نموده و با پرواز مستقیم از لندن به مقصد توکیو در ژاپن عزیمت می‌نماید و سپس از طریق ترن از ایستگاه توکیو به شهر ناکازاکی مسافرت خویش را کامل می‌نماید. در این برنامه‌ی سفر که آژانس مسافرتی آن را طراحی نموده است. به ترتیب مسئولیت انتقال جهانگرد از آکسفورد به لندن برعهده آژانس تشریفات، از لندن به مقصد فرودگاه توکیو برعهده شرکت هواپیمایی صادر کننده بلیط و از ایستگاه قطار توکیو به مقصد ناکازاکی بر دوش شرکت راه آهن ژاپن است. هریک از این سه قطعه در این برنامه‌ی مسافرتی متقسیماً بین دو مکان مجاور است. بدیهی است هر سه قطعه‌ی ترابری از سوی شرکت‌های مختلف مدیریت می‌شوند و روش‌های مختلف حمل و نقل را نیز (خودرو، هواپیما، ترن) شامل می‌شوند. با وجود این که روش‌های انتقال هریک متفاوت است اما در هر حال هریک خدمات اساسی جابه‌جایی مسافران از یک مکان به یک نقطه مجاور آن را انجام می‌دهند. در این شبیه‌سازی عملیات انتقال، جهانگرد داده گرام است و هریک از قطعه‌های انتقال یک پیوند ارتباطی به شمار می‌آیند. بدیهی است در این شبیه‌سازی روش‌های انتقال مانند خودرو، هواپیما و ترن هریک پروتکل لایه‌ی پیوند و آژانس مسافرتی نیز پروتکل مسیریابی قلمداد می‌شود. اگرچه خدمات اساسی هرلایه‌ی پیوند جابه‌جایی داده گرام‌ها از یک گره به یک گره‌ی مجاور از طریق یک پیوند ارتباطی می‌باشد اما جزئیات خدمات فراهم شده می‌تواند از یک پروتکل لایه‌ی پیوند به پروتکل لایه‌ی پیوند بعدی تغییر نماید. خدمات امکان‌پذیر که پروتکل‌های لایه‌ی پیوند قادر به تأمین آن‌ها می‌باشند عبارتند از:

۱- **تشکیل قاب یا فریم.** تقریباً همگی پروتکل‌های لایه‌ی پیوند، داده گرام‌های متعلق به لایه‌ی شبکه را قبل از انتقال از طریق یک پیوند در قالب قاب‌های داده بسته‌بندی می‌کنند. یک فریم داده شامل چندین دامنه است از جمله محدوده‌ی داده که داده گرام لایه‌ی شبکه به آن تزریق می‌شود، تعدادی دامنه‌ی سرپیام و علاوه بر آن به یک قاب ممکن است دامنه‌ی دنباله **(trailer)** نیز افزوده گردد. معمولاً این دامنه‌ی اخیر به همراه سرپیام در حالت کلی دامنه‌ی سرپیام خوانده می‌شود. ساختار هر قاب نیز از سوی پروتکل لایه‌ی پیوند تعریف می‌شود. در بخش دوم فصل ۵ چندین نمونه از فریم‌های مختلف از لحاظ ساختاری بررسی می‌شوند.

۲- **دستیابی پیوند.** یک **پروتکل کنترل دستیابی رسانه**[1] **(MAC)** قواعد انتقال قاب داده برروی یک پیوند را تعریف می‌کند. برای پیوندهای نقطه به نقطه که فقط یک فرستنده در یک انتهای پیوند و نیز یک گیرنده در انتهای دیگر پیوند قرار دارند پروتکل کنترل دستیابی رسانه (MAC) بسیار ساده است از این رو که فرستنده هرگاه پیوند آزاد است می‌تواند نسبت به گسیل قاب اقدام نماید. این امکان وجود دارد که در این حالت نقطه به نقطه MAC وجود نداشته باشد. جالب‌ترین حالت هنگامی است که چندین گره از یک پیوند فراپخش به‌صورت مشترک بهره می‌گیرند. این چالش اصطلاحاً مسئله‌ی دستیابی چندگانه نامیده شده است. در این حالت پروتکل MAC هماهنگ کننده‌ی انتقال قاب‌های تعداد زیادی گره می‌شود.

۳- **تحویل مطمئن.** هنگامی که پروتکل لایه‌ی پیوند خدمات تحویل مطمئن را ارائه می‌دهد این

1- Medium access control

ضمانت را فراهم می‌سازد که جابه‌جایی هر داده گرام لایه‌ی شبکه از درون یک پیوند عاری از خطا انجام می‌شود. با توجه به مطالب گذشته برخی از پروتکل‌های لایه‌ی انتقال مانند TCP نیز یک انتقال مطمئن داده را ضمانت می‌کردند. مشابه آن‌ها پروتکل‌های لایه‌ی پیوند نیز ممکن است این خدمات را از طریق پیام‌های تأئید (ACK) و انتقال مجدد تأمین نمایند. معمولاً از امکانات تحویل مطمئن قاب‌های داده برای پیوندهای بی‌سیم نام برد. در این پروتکل‌ها هدف تصحیح خطا در پیوندی است که داده‌ها در آن دچار خطا شده‌اند تا آنکه ضرورت به انتقال مجدد داده‌ها از طریق پروتکل‌های لایه‌ی انتقال و یا کاربر، در دو انتهای پیوند باشد. بدیهی است تحویل مطمئن داده از سوی لایه‌ی پیوند می‌تواند به مفهوم افزایش سرآیند غیرضروری برای پیوندهایی باشد که احتمال وقوع خطا در آن‌ها نسبتاً اندک است. از جمله‌ی این پیوندها تارهای نوری، کابل هم‌محور و بسیاری از پیوندهای از نوع TP می‌باشد. به همین منظور اغلب پروتکل‌های کابلی پیوند داده خدمات تحویل مطمئن را به قاب‌های داده ارایه نمی‌دهند.

۴- **کنترل جریان.** گره‌ها در هر سوی یک پیوند ظرفیت افرسازی یا ذخیره‌ی محدودی از قاب‌های داده را دارند. هنگامی که یک گره‌ی دریافت کننده، قاب‌های داده را با سرعت بیش از توان پردازش خود دریافت نماید این محدودیت ظرفیت سبب نگرانی است. بدون فرآیند کنترل جریان بافرهای گیرنده دچار سرریزی می‌شوند و در نتیجه فقدان قاب روی می‌دهد. مانند لایه‌ی انتقال یک پروتکل لایه‌ی پیوند می‌تواند جهت جلوگیری از سیلابی شدن گره‌ی گیرنده در یک طرف پیوند از جانب گره‌ی فرستنده در طرف مقابل آن پیوند، خدمات کنترل جریان را فراهم سازد.

۵- **آشکارسازی خطا.** در گره‌ی گیرنده امکانات سخت‌افزاری لایه‌ی پیوند می‌توانند در مورد یک بیت در قاب که دارای ارزش صفر است دچار خطا شوند در حالیکه بیت انتقال یافته ارزش یک را داشته است. بدیهی است عکس این رویداد نیز امکان‌پذیر است. چندین خطاهایی برروی بیت‌ها به واسطه‌ی سیگنال‌های ضعیف و نویز حاصل از فرآیند الکترومغناطیسی روی می‌دهد که اصطلاحاً نویز EM[1] نامیده می‌شود.

۶- **تصحیح خطا.** چگونگی انجام عملیات تصحیح خطا مشابه آشکارسازی آن می‌باشد. با این تفاوت که گره‌ی گیرنده نه تنها آشکارسازی بیت‌های دچار خطا را در یک قاب انجام می‌دهد بلکه مکان دقیق خطا در فریم را نیز مشخص و سپس نسبت به تصحیح آن اقدام می‌کند. برخی از پروتکل‌ها امکانات تصحیح خطای لایه پیوند را فقط برای سرپیام بسته تأمین می‌کنند.

۷- **خدمات دوطرفه همزمان و غیرهمزمان.** در حالت **خدمات دوطرفه‌ی همزمان**[2] گره‌ها در دو سوی لایه‌ی پیوند انتقال بسته‌ها را همزمان انجام می‌دهند. در حالت **خدمات دوطرفه‌ی غیرهمزمان**[3] یک گره نمی‌تواند هر دو عملیات ارسال و دریافت را همزمان انجام دهد.

همان‌گونه که در مطالب بالا ملاحظه گردید بسیاری از خدمات فراهم شده از سوی لایه‌ی پیوند حالت موازی را با خدمات تأمین شده از سوی لایه‌ی انتقال دارند به‌عنوان مثال هر دو لایه‌ی پیوند و انتقال می‌تواند خدمات تحویل مطمئن را ارایه نماید. اگرچه روش‌های تأمین این خدمات در دو لایه

1- Electromagnetic noise
2- Full duplex service
3- Half duplex service

مشابه است اما دو سرویس ارایه شده مشابه یکدیگر نمی‌باشند. یک پروتکل لایه‌ی انتقال تحویل مطمئن قطعه‌ها بین دو فرآیند را براساس الگوی انتها به انتها انجام می‌دهد در حالیکه پروتکل لایه‌ی پیوند تحویل مطمئن قاب‌هی داده را بین دو گره‌ی متصل به یکدیگر از طریق یک پیوند انجام می‌دهد. هم‌چنین پروتکل‌های هر دو لایه‌ی پیوند و انتقال امکانات آشکارسازی خطا و کنترل جریان را نیز فراهم می‌آورند. کنترل جریان در لایه‌ی انتقال برمبنای روش انتها به انتها پایه‌گذاری می‌شود در صورتیکه لایه‌ی پیوند این عملیات را برمبنای یک گره به گره‌ی مجاور انجام می‌دهد.

۵-۲ مکان پیاده‌سازی لایه‌ی پیوند

قبل از ورود به جزئیات پروتکل‌های لایه‌ی پیوند ابتدا مکان پیاده‌سازی لایه‌ی پیوند بررسی می‌شود. نقطه‌ی تمرکز را بر سیستم انتهایی قرار می‌دهیم زیرا همان‌گونه که از فصل ۴ آموختیم پیاده‌سازی لایه‌ی پیوند در کارت خط مسیریاب انجام می‌گرفت. اکنون این پرسش مطرح می‌شود آیا لایه‌ی پیوند میزبان در سخت‌افزار پیاده‌سازی می‌شود یا نرم‌افزار؟ آیا برروی یک کارت مجزا و یا تراشه‌ای پیاده‌سازی می‌شود؟ چگونه با بقیه‌ی عناصر سخت‌افزار و سیستم عامل ارتباط برقرار می‌کند؟

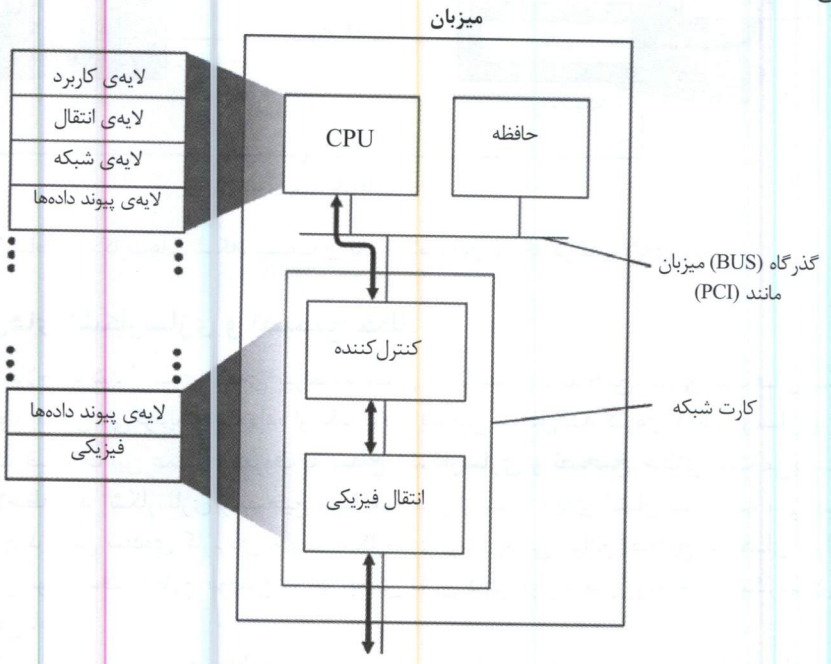
شکل ۵-۲ معماری یک میزبان را نشان می‌دهد. در اغلب قطعه‌های سخت‌افزاری لایه‌ی پیوند در **واسط شبکه** و یا آن‌چه اصطلاحاً **کارت واسط شبکه (NIC)**[1] نامیده شده است پیاده‌سازی می‌شود. در هسته‌ی کارت شبکه (NIC) مدار کنترل لایه‌ی پیوند قرار دارد. این مدار معمولاً یک تراشه‌ی ویژه است که بسیاری از خدمات لایه‌ی پیوند از قبیل تشکیل قاب، دستیابی پیوند، کنترل جریان، آشکارسازی خطا و غیره را پیاده‌سازی می‌کند. بنابراین بسیاری از عملیات لایه‌ی پیوند در سخت‌افزار پیاده‌سازی شده است. به‌عنوان مثال تراشه‌ی 8254x از شرکت اینتل کنترل کننده‌ای است که پروتکل‌های اترنت را پیاده‌سازی می‌کند. تراشه‌ی دیگر AR5006 از شرکت اترس کنترل کننده‌ای است که پروتکل 802.11 که متعلق به WiFi می‌باشد را پیاده‌سازی می‌کند. تا اواخر دهه‌ی 1990 میلادی اغلب کارت‌های شبکه حالت فیزیکی مجزا اختیار می‌کردند. مانند کارت PCM-CIA و یا کارت‌های از نوع PCI که در مکان‌های درنظر گرفته‌شده‌ی برد مادر یک کامپیوتر شخصی نصب می‌شدند. در حال حاضر اکثریت قریب به اتفاق کارت‌های واسط به‌صورت فیزیکی در درون سیستم‌ها جایگذاری شده‌اند.

در سمت راست شبکه تراشه‌ی کنترل، داده گرام تعریف و ذخیره شده از سوی لایه‌های بالاتر پشته‌ی پروتکلی در حافظه‌ی میزبان را در قالب یک قاب داده‌ی پیوند بسته‌بندی می‌نماید و سپس آن را به درون پیوند ارتباطی برمبنای راهبردهای پروتکل دستیابی پیوند گسیل می‌کند. در سمت دریافت شبکه کنترل کننده قاب داده‌ی ارسالی را دریافت و داده گرام لایه‌ی شبکه را از آن جداسازی می‌کند. اگر لایه‌ی پیوند عملیات آشکارسازی خطا را انجام دهد آنگاه مدار کنترل فرستنده مجموعه‌ای از بیت‌های آشکارساز خطا را به‌همراه اطلاعات دیگر در سرپیام قاب قرار می‌دهد. مدار کنترل گیرنده براساس این اطلاعات عملیات آشکارسازی خطا را انجام می‌دهد. چنانچه لایه‌ی پیوند مکانیزم کنترل جریان را نیز شامل شود آن‌گاه هر دو مدار کنترل فرستنده و گیرنده اطلاعات مربوط به کنترل جریان را بین یکدیگر مبادله می‌کنند به‌گونه‌ای که فرستنده قاب‌ها را با سرعتی گسیل نماید که گیرنده قادر به دریافت آن‌ها باشد.

شکل ۵-۲ کارت شبکه‌ای را نشان می‌دهد که متصل به یک گذرگاه میزبان مانند گذرگاه PCI و یا

1- Network Interface Card

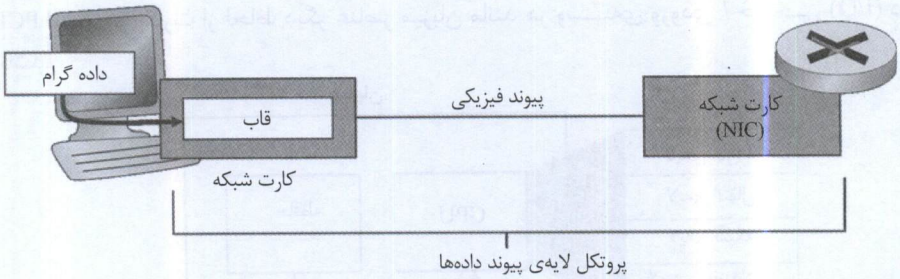
PCI-X است. این کارت از لحاظ دیگر عناصر میزبان مانند هر وسیله‌ی ورودی / خروجی (I/O) دیگر می‌باشد.



شکل ۵-۲ ♦ کارت شبکه، رابطه‌ی کارت شبکه با دیگر عناصر میزبان و عملیات پشته‌ی پروتکلی

در شکل ۵-۲ هم چنین نشان داده شده است که بخش وسیعی از پروتکل لایه‌ی پیوند در سخت افزار کارت شبکه پیاده‌سازی شده است اما بخشی دیگر از پروتکل نیز در نرم‌افزار اجرایی CPU در میزبان قرار دارد. عناصر نرم‌افزاری لایه‌ی پیوند اساساً عملیات سطح بالاتر لایه‌ی پیوند مانند دریافت داده گرام از لایه‌ی شبکه، افزون اطلاعات آدرس لایه‌ی پیوند و فعال‌سازی مدار سخت‌افزار کنترل را انجام می‌دهند. در سمت دریافت شبکه، نرم‌افزار لایه‌ی پیوند واکنش نسبت به وقفه‌های مدار کنترل در ارتباط با دریافت یک یا چند قاب، بررسی شرایط خطا و عبور داده، گرام‌ها به لایه‌ی بالاتر یعنی لایه‌ی شبکه را از خود به نمایش می‌گذارد. بنابراین لایه‌ی پیوند، ترکیبی از سخت‌افزار و نرم‌افزار است در نقطه‌ای از پشته‌ی پروتکلی که نرم‌افزار با سخت افزار تلاقی می‌کند.

شکل ۵-۳ کارت‌های فرستنده و گیرنده را نشان می‌دهد. عملیات اصلی پروتکل لایه‌ی پیوند از سوی مدار کنترل اجرایی می‌شود. کارت‌ها حالت نیمه مستقل را دارند که نقش آن‌ها انتقال یک قاب از یک کارت شبکه به کارت دیگر است. پژوهش‌های متعددی از نقطه نظر افزایش عملیات در کارت‌های شبکه انجام گرفته است. به‌عنوان مثال تراشه‌ی کنترل 8254x می‌تواند کد مجموع مقابله‌ی TCP/IDP و نیز کد مجموع مقابله‌ی سر پیام IP در سخت افزار را محاسبه نماید. این دودر حقیقت عملیات مربوط به لایه‌های انتقال به شبکه می باشند که از سوی مدار کنترل لایه‌ی پیوند انجام می‌گیرد. اگرچه به ظاهر انجام این عملیات مداخله و تجاوز به استقلال لایه‌ها و اصولاً فلسفه‌ی لایه‌بندی است اما نقطه‌ی قوت آن در این نکته خلاصه می‌شود که محاسبه‌ی کد مجموع مقابله را می‌توان از طریق سخت افزار سریع‌تر در مقایسه با نرم‌افزار انجام داد. بنابراین می‌توان به نوعی از تأثیر این مداخله صرف‌نظر نمود.



شکل ۵-۰۳♦ ارتباط بین کارت‌های شبکه، بسته‌بندی داده گرام لایه‌ی شبکه در قاب لایه‌ی پیوند

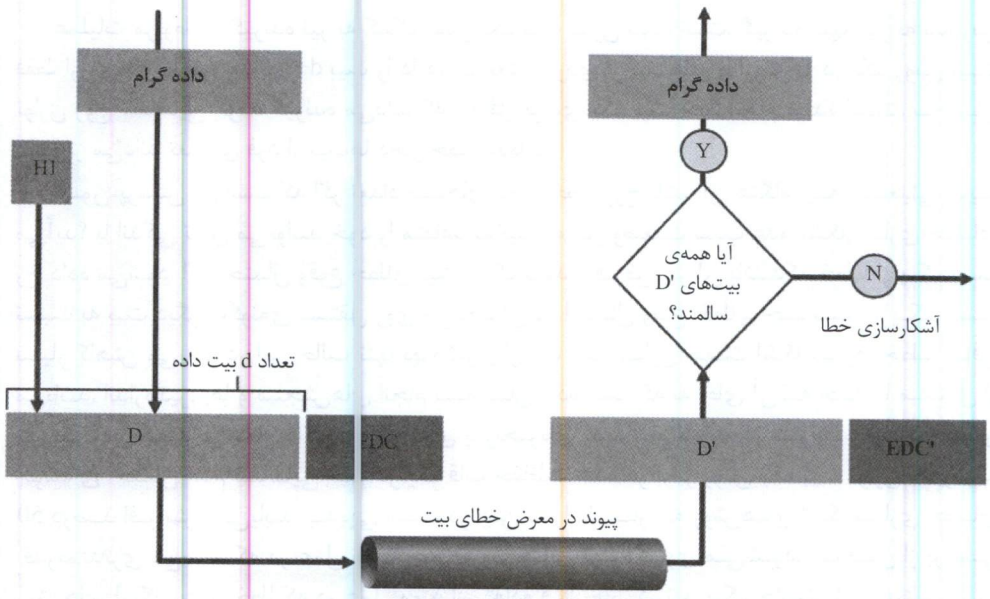
۵-۳ روش‌های آشکارسازی و تصحیح خطا

در بخش قبل ملاحظه شد از سوی لایه‌ی پیوند دو امکان آشکارسازی و تصحیح خطای بیت‌های یک قاب لایه‌ی پیوند تأمین می‌گردید. قاب داده از یک گره‌ی فیزیکی متصل به گره‌ی دیگر ارسال و یا دریافت می‌شد. اصطلاحاً این عملیات معروف به **سطح آشکارسازی و تصحیح خطای بیت** می‌باشد. در فصل ۳ ملاحظه شد آشکارسازی و تصحیح خطا هم‌چنین از سوی لایه‌ی انتقال نیز فراهم می‌آید. در این بخش چند روش ساده‌ی کاربردی جهت آشکارسازی و در بعضی مواقع تصحیح بیت‌های دارای خطا بررسی می‌شود. هدف از طرح موضوع در این بخش نشان دادن توان‌مندی روش‌های آشکارسازی و تصحیح خطا می‌باشد.

شکل ۵-۴ مبنای موضوع مورد مطالعه در این بخش را نشان می‌دهد. در سمت گره‌ی ارسال کننده، داده‌ی D بایستی در برابر خطای بیت محافظت گردد و بنابراین بیت‌های آشکارسازی و تصحیح خطا که به اختصار **(EDC)** نامیده شده‌اند به داده‌ی D اضافه می‌گردد. اصولاً داده‌ی محافظت شده فقط شامل داده گرام عبوری از لایه‌ی شبکه جهت انتقال از یک پیوند نمی‌باشد بلکه اطلاعات آدرس‌دهی سطح پیوند، شمارگان ردیف و دیگر دامنه‌های سرپیام قاب لایه‌ی پیوند را نیز شامل می‌شود. هر دو D و EDC در یک قاب داده‌ی لایه پیوند به گره‌ی گیرنده گسیل می‌شوند. در گره‌ی گیرنده ردیفی از بیت‌های 'D و 'EDC دریافت می‌شود. بدیهی است 'D و 'EDC ممکن است از بیت‌های D و EDC اصلی در اثر تغییر برخی از بیت‌ها متفاوت باشند.

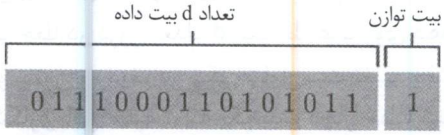
چالشی که گیرنده با آن مواجه است شناسایی این موضوع است که آیا 'D مشابه D می‌باشد یا خیر؟ با فرض آن‌که گیرنده فقط 'D و 'EDC را دریافت نموده است. بیان دقیق تصمیم‌گیری گیرنده از لحاظ این که آیا خطایی روی داده است یا خیر و یا آن‌که خطا آشکارسازی شده است یا خیر در شکل ۵-۴ بسیار حائز اهمیت است. روش‌های آشکارسازی و تصحیح خطا اجازه می‌دهند بعضی اوقات و نه همیشه گیرنده به آشکارسازی بیت‌های خطا بپردازد. حتی با بهره‌گیری از بیت‌های آشکارساز خطا هنوز هم این امکان وجود دارد که برخی از **بیت‌های خطا آشکارسازی نگردند**. به عبارت دیگر گیرنده آگاهی از بیت‌های خطا در اطلاعات دریافتی نداشته باشد. در نتیجه گیرنده ممکن است داده گرام آلوده را به لایه‌ی شبکه تحویل نماید و یا آنکه از آلوده بودن اطلاعات یک دامنه از سرپیام قاب بی‌خبر باشد. بنابراین لازم است یک روش آشکارسازی خطا که احتمال چنین مخاطراتی را کاهش دهد گزینش گردد. بدیهی است هرچه روش‌های آشکارسازی و تصحیح خطا که احتمال عدم شناسایی بیت‌های خطا را کاهش می‌دهد پیچیده‌تر و پیشرفته‌تر باشد سرآیند افزون‌تری را نیز سبب خواهد شد.

این سرآیند افزون، محاسبات بیش‌تری را جهت انتقال تعداد قابل توجهی از بیت‌های آشکارسازی و تصحیح خطا نیز به همراه دارد.



شکل ۵-۴ ♦ چگونگی آشکارسازی و تصحیح خطا

در ادامه‌ی مطلب سه روش آشکارسازی خطا که عبارتند از: **آزمون داده‌ی توازن** انتقال یافته که در اصل مبنای فلسفه‌ی آشکارسازی و تصحیح خطا می‌باشد، روش‌های کدهای مجموع مقابله که متعلق به لایه‌ی انتقال است و سنجش افزونگی دوره‌ای (CRC) که، یک کارت شبکه‌ی لایه‌ی پیوند از آن بهره می‌گیرد، بررسی و تحلیل می‌شوند.



شکل ۵-۵ ♦ توازن زوج یک بیت

۱- سنجش بیت توازن

شاید ساده‌ترین شکل آشکارسازی خطا بهره‌گیری از یک **بیت توازن**[1] است. اگر فرض شود اطلاعات ارسال شده در شکل ۵-۵ در قالب قاب داده‌ی D برابر d بیت باشد. آن‌گاه در روش بیت توازن زوج فرستنده یک بیت اضافه را به گونه‌ای گزینش می‌کند که مجموع تعداد بیت‌های با ارزش 1 در تعداد کل d+1 بیت که شامل اطلاعات اصلی D به همراه یک بیت توازن، است یک رقم زوج گردد. در روش بیت توازن فرد گزینش ارزش بیت به‌گونه‌ای است که تعداد بیت‌های حامل ارزش 1 عددی فرد شود.

1- Parity bit

شکل ۵-۵ الگوی بیت توازن زوج را نشان می‌دهد. در این شکل یک بیت توازن در یک دامنه‌ی مجزا ذخیره شده است.

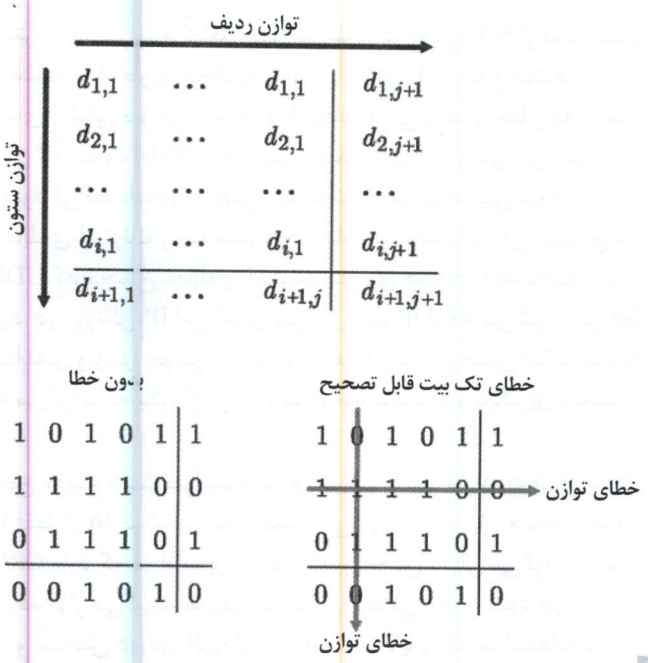
عملیات مربوط به گیرنده نیز به کمک روش یک بیت توازن ساده است. گیرنده تنها نیاز به شمارش فقط ارزش‌های 1 در تعداد d+1 بیت را دارد. اگر تعداد فردی از بیت‌های با ارزش 1 در یک روش بیت توازن زوج شناسایی گردد گیرنده می‌داند که حداقل برروی یک بیت خطا انجام شده است. به بیان دقیق‌تر می‌داند تعدادی فرد از بیت‌ها دچار خطا شده‌اند.

اکنون پرسش آن است که اگر تعداد بیت‌های دچار خطا، زوج باشد آن هنگام چه وضعیتی پدید می‌آید؟ با اندکی تأمل می‌توانید خود را متقاعد نمائید که این وضعیت سبب عدم آشکارسازی خطای رخ داده می‌شود. اگر احتمال وقوع خطای بیت اندک باشد و فرض بر آن باشدکه خطا از یک بیت نسبت به بیت دیگر به‌گونه‌ی مستقل روی می‌دهد آن‌گاه احتمال وقوع خطای چند بیت در یک بسته بسیار کاهش می‌یابد. در این حالت تنها بهره‌گیری از یک بیت توازن جهت آشکارسازی خطا کافی می‌باشد. اندازه‌گیری‌ها و سنجش‌های انجام شده نشان داده است که به جای آن‌که خطاها مستقل از یکدیگر روی دهند در عمل به‌گونه‌ی خوشه‌ای یا زنجیره‌ای پدیدار می‌شوند. در شرایط رخداد خطای خوشه‌ای احتمال عدم شناسایی خطا در یک قاب حفاظت شده در برابر روش یک بیت توازن به نزدیک 50 درصد افزایش می‌یابد. بدیهی است در این شرایط نیاز به روش‌های آشکارسازی خطای قدرتمندتری می‌باشد که در عمل نیز از چنین روش‌هایی بهره‌گیری می‌شود. اما قبل از بررسی روش‌های آشکارسازی خطا که در عمل مورد استفاده قرار می‌گیرند یک حالت کلی یعنی بیت توازن یک بیت که برخی حقایق درون روش‌های آشکارسازی خطا را مشخص می‌کند تحلیل می‌شود.

شکل ۵-۶ نماد کلی از روش یک بیت توازن را در قالب دوبعدی آن به نمایش می‌گذارد. در این شکل تعداد d بیت درون D به ترتیب به تعداد i و j ردیف و ستون تقسیم شده‌اند. یک ارزش توازن برای هر ردیف و هر ستون محاسبه شده است. نتیجه‌ی حاصل از تعداد (i+j+1) بیت در درون قاب داده‌ی لایه‌ی پیوند به‌عنوان بیت‌های آشکارسازی خطا قرار می‌گیرند.

اگر فرض شود یک بیت خطا در درون تعداد d بیت اطلاعات روی دهد. آن‌گاه در این روش **توازن دوبعدی**[1] بیت توازن هر دو ستون و ردیف در شکل شامل بیت تغییر یافته و در نتیجه خطا می‌شوند. گیرنده آن‌گاه نه تنها به آشکاری این حقیقت که یک بیت دچار خطا شده است می‌پردازد بلکه از شاخص ردیف و ستون با خطای توازن می‌تواند دقیقاً مکان خطا را نیز شناسایی و آن را تصحیح نماید! شکل ۵-۶ مثالی از این رویداد را نشان می‌دهد. در شکل یک بیت با ارزش 1 در مکان (2,2) ارزش واقعی **0** یک بیت را تغییر داده است و بنابراین خطای رخ داده در گیرنده هم قابل شناسایی است و هم می‌توان آن را تصحیح نمود. اگرچه تمرکز بحث برروی وقوع خطا در d بیت اطلاعات اصلی می‌باشد اما وقوع تنها یک خطای بیت در بیت‌های توازن را نیز می‌توان آشکارسازی نمود و هم می‌توان آن‌ها را تصحیح کرد. از طرف دیگر در روش توازن دوبعدی می‌توان هر ترکیبی از دو خطا در یک بسته را آشکارسازی نمود بدون آن‌که عملیات تصحیح آن‌ها انجام‌پذیر باشد.

1- Two dimensional parity



شکل ۵-۶ ♦ روش توازن زوج دوبعدی

توانمندی گیرنده در آشکارسازی و تصحیح خطا اصطلاحاً **تصحیح خطا به جلو**[1] **(FEC)** نام گرفته است. این روش‌ها در تجهیزات ذخیره‌سازی صوتی و بخش رسانه‌ی مانند CD متداول می‌باشد. در شبکه‌های کامپیوتری از روش‌های FEC به‌گونه‌ی مستقل و یا در ارتباط با الگوریتم ARQ در لایه‌ی پیوند مشابه آن‌چه در فصل ۳ مطرح گردید بهره‌برداری می‌شود. روش‌های FEC از آن جهت ارزشمند می‌باشند که سبب کاهش تعداد مراتب انتقال مجدد فرستنده می‌شوند. شاید مهم‌ترین نکته آن باشد که تصحیح بی‌درنگ خطاها را در گیرنده سبب می‌شوند. در این حالت فرستنده جهت دریافت پیام NAK نیازمند انتظار زمان تأخیر انتشار (TTL) نمی‌باشد. هم‌چنین این انتظار زمانی برای حالتی هم نیاز نمی‌باشد که بسته‌ی مجدداً انتقال یافته به گیرنده بازگشت می‌نماید. این ویژگی برای کاربردهای زمان واقعی و نیز پیوندهای در عمق فضا که تأخیر انتشار نسبتاً طولانی دارند بسیار با اهمیت به‌شمار می‌آید.

۲- روش‌های کد مجموع مقابله

در روش‌های کد مجموع مقابله تعداد d بیت شکل ۵-۵ به‌عنون ردیفی از تعداد k بیت اعدادکامل درنظر گرفته می‌شوند. یکی از ساده‌ترین روش‌های مجموع مقابله جمع این k بیت عدد کامل است که از نتیجه‌ی حاصل جمع آن‌ها به‌عنوان بیت‌های آشکارسازی خطا استفاده می‌شود. **کد مجموع مقابله‌ی اینترنت**[2] برمبنای این رویکرد عمل می‌کند. بایت‌های داده در قالب ارزش‌های کامل 16 بیت درنظر گرفته می‌شوند و سپس حاصل جمع آن‌ها به‌دست می‌آید. آن‌گاه **مکمل 1** این مجموع به‌عنوان

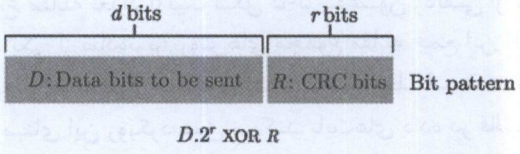
1- Forward Error Correction
2- Internet checksum

کد مجموع مقابله‌ی اینترنت در سر پیام قطعه حمل می‌شود. در بخش ۳-۶ از فصل سوم عنوان گردید که گیرنده کد مجموع مقابله را از طریق محاسبه‌ی ارزش مکمل 1 از جمع بیت‌های داده‌ی دریافتی که شامل کد مجموع مقابله‌ی ارسالی نیز می‌شود مجدداً محاسبه می‌نماید و چنان‌چه حاصل نتیجه شده‌ی همه‌ی بیت‌ها 1 باشد آن‌گاه صحت اطلاعات دریافتی تأئید می‌گردد در غیر این صورت چنانچه یکی از بیت‌ها دارای ارزش 0 شود آن‌گاه داده‌ی دریافتی آلوده شده قلمداد می‌شود. مستند [RFC1071] الگوریتم کد مجموع مقابله‌ی اینترنت را به‌همراه جزئیات پیاده‌سازی آن تشریح نموده است. در پروتکل‌های TCP و UDP کد مجموع مقابله‌ی اینترنت برروی همه‌ی دامنه‌ها یعنی بخش داده و سرپیام محاسبه می‌شود. در پروتکل IP این کد برروی سر پیام IP انجام می‌گیرد زیرا قطعه‌های TCP و یا UDP کد مجموع مقابله‌ی ویژه‌ی خویش را همراه دارند. در پروتکل‌های دیگر مانند XTP دو کد مجموع مقابله محاسبه می‌گردد که یکی از آن‌ها مربوط به سرپیام و دیگری متعلق به تمام پیام می‌باشد.

روش‌های کد مجموع مقابله نسبتاً سرآیند بسته‌ی اندکی را نیاز دارند. به‌عنوان مثال کد مجموع مقابله‌ی TCP و UDP فقط از 16 بیت سرآیند تشکیل می‌شوند. در مقابل حفاظت ضعیفی را در برابر خطاها در مقایسه با CRC دارد که در ادامه‌ی مطالب همین بخش دنبال می‌گردد. روش CRC اغلب در لایه‌ی پیوند مورد استفاده قرار می‌گیرد، طبیعی است این پرسش مطرح شود چرا از روش‌های مجموع مقابله در لایه‌ی انتقال و سنجش دوره‌ی افزونگی یا CRC در لایه‌ی پیوند استفاده می‌شود؟ لازم است یادآوری شود که لایه‌ی انتقال در بخش نرم‌افزارهای یک میزبان به‌عنوان بخشی از عناصر سیستم عامل پیاده‌سازی می‌شود. بنابراین بهره‌گیری از یک روش ساده‌ی آشکارسازی خطا مانند کد مجموع مقابله بسیار بااهمیت می‌باشد. از طرف دیگر آشکارسازی خطا در لایه‌ی پیوند در بخش سخت‌افزار ویژه‌ی کارت‌های شبکه پیاده‌سازی می‌شود که می‌توان با سرعت عملیات CRC به مراتب پیچیده‌تر را انجام دهند.

۳- سنجش افزونگی دوره‌ای (CRC)

آن‌چه امروز در شبکه‌های کامپیوتری به‌عنوان یک روش گسترده‌ی آشکارسازی خطا مورد استفاده قرار می‌گیرد برمبنای **کدهای سنجش افزونگی دوره‌ای**[1] **(CRC)** می‌باشد. کدهای CRC را هم‌چنین عبارت‌های **چندجمله‌ای**[2] نیز می‌نامند زیرا این امکان وجود دارد که رشته بیت ارسالی را در قالب یک عبارت چندجمله‌ای که ضرایب با ارزش‌های 0 و یا 1 را در رشته اختیار می‌کنند به‌شمار آورد. در این نمایش چندجمله‌ای عملیات انجام‌گرفته برروی رشته بیت در قالب ریاضیات چندجمله‌ای‌ها تفسیر می‌شود.



شکل ۵-۷ ♦ کدهای چندجمله‌ای (CRC)

1- Cyclic redundancy check
2- Polynomial codes

چگونگی عملکرد کدهای CRC به این گونه است که: اگر فرض شود تعداد d بیت موجود در قطعه داده‌ی D که یک گره‌ی ارسال کننده مایل به گسیل آن به یک گیرنده است، در قالب یک قاب داده بسته‌بندی شده باشد. در این حالت فرستنده و گیرنده نخست بایستی برروی یک الگوی (r+1) بیت که اصطلاحاً به آن **مولد** گفته می‌شود و با حرف G نیز مشخص می‌شود توافق نمایند. همواره با ارزش‌ترین بیت (MSB) که در انتهای چپ G قرار می‌گیرد دارای ارزش ثابت 1 می شود. شکل ۵-۷ معماری کدهای CRC را نشان می‌دهد. برای یک قطعه داده‌ی D فرستنده تعداد r بیت اضافه را با عنوان R گزینش و به D پیوند می‌دهد. الگوی این پیوند به این گونه است که تعداد (d+r) بیت در قالب یک عدد دو دویی (باینری) چنان تفسیر می‌شود که بر G تقسیم‌پذیر است. این نکته‌ی اخیر به این مفهوم است که محاسبه‌ی تقسیم پیمانه‌ی -2 باقی مانده‌ای را از خود برجای نخواهد گذاشت. نتیجه‌ی حاصل از این فرض سبب ساده شدن فرآیند سنجش خطا با بهره‌گیری از CRC است. گیرنده‌ی قاب داده اطلاعات (d+r) بیت دریافتی را بر G مجدداً تقسیم می‌کند اگر باقی‌مانده‌ی تقسیم داده به‌عنوان اطلاعات بدون خطا مورد پذیرش واقع می‌شود.

کلیه‌ی فرآیندهای محاسبه‌ی CRC در قالب محاسبات پیمانه‌ی -2 بدون بیت نقلی در عملیات جمع و یا بیت قرضی در تفریق می‌باشد. مفهوم این جمله آن است که عملیات جمع و تفریق مشابه می‌باشند و هر دو از لحاظ مقایسه‌ی بیت‌ها معادل عملیات XOR عمل می‌کنند. به عنوان مثال هریک از موارد ذیل را درنظر بگیرید:

1011 XOR 0101 =1110
1001 XOR 1101 =0100

با اندکی تأمل این عملیات مشابه دو مورد تفریق ذیل است:

1011-0101=1110
1001-1101=0100

بدیهی است عملیات ضرب و تقسیم مشابه محاسبه‌ی برمبنای 2 می‌باشد با این تفاوت که هر جمع و یا تفریق مورد نیاز بدون درنظر گرفتن به نیاز بیت نقلی و یا قرضی می‌باشد. مانند عملیات معمولی محاسبه‌ی دو دویی (باینری) حاصلضرب 2^K به مفهوم جابه‌جایی منطقی به چپ الگوی بیت به تعداد دفعه‌های K می‌باشد. بنابراین با در اختیار داشتن D و R کمیت حاصل از رابطه‌ی (۵-۱) در ذیل الگوی (d+r) بیت نشان داده شده در شکل ۵-۷ را نتیجه می‌دهد. از این ویژگی جبری الگوی (d+r) بیت در شکل ۵-۷ در ادامه‌ی مطالب این بخش استفاده خواهد شد.

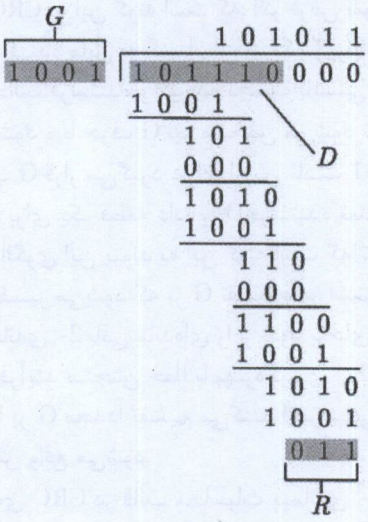
$$(D.2^r)XOR(R) \qquad (۵-۱)$$

اکنون با بازگشت به پرسش اصلی چگونگی محاسبه‌ی R از سوی گیرنده پی‌گیری موضوع CRC را دنبال می‌کنیم، یادآوری این نکته ضروری است که به دنبال محاسبه‌ی R به‌گونه‌ای که یک ارزش n در قالب معادله (۵-۲) ذیل را نتیجه دهد، می‌باشیم:

$$(D.2^r)XOR(R)=NG \qquad (۵-۲)$$

به عبارت دیگر به دنبال گزینش مقدار R به‌گونه‌ای می‌باشیم که نتیجه تقسیم عبارت (۵-۲) بر G باقی مانده‌ای را از خود برجای نگذارد. از نتیجه‌ی انجام عملیات XOR ارزش R با مقادیر دو طرف رابطه‌ی (۵-۲) معادله‌ی (۵-۳) حاصل می‌شود:

$$(D.2^r)=(nG)XOR(R) \qquad (۵-۳)$$



شکل ۵-۸ ♦ محاسبه‌ی یک نمونه از CRC

این معادله می‌گوید اگر عبارت $(D.2^r)$ بر G تقسیم شود ارزش باقی مانده دقیقاً برابر R است. به عبارت دیگر مقدار R از طریق معادله (۵-۴) قابل محاسبه است.

$$R = \frac{(D.2^r)}{G} \qquad (۴-۵)$$

شکل ۵-۸ مراحل این محاسبه را برای مقادیر: (D=101110)، d=6، (G=1001) و r=3 نشان می‌دهد. تعداد 9 بیت گسیل شده در این حالت (101110011) می‌باشد. با انجام یک محاسبه‌ی ساده براساس موارد عنوان شده در خواهید یافت که درستی رابطه‌ی $(D.2^r)=[(101011*G)XOR(R)]$ برقرار است.

از نقطه نظر استانداردهای بین‌المللی مولد G در قالب 8، 12، 16و 32 بیت تعریف شده است. بنابراین کد CRC-32 از رشته‌ی 32 بیت استاندارد گزینش شده‌ی پروتکل‌های لایه‌ی پیوند (IEEE) در قالب کد مولد ذیل استفاده می‌کند:

$$G_{CRC-32}=100000100110000010001110110110111$$

هریک از CRCهای استاندارد اشاره شده می‌توانند تعداد بیت‌های خطای کم‌تر از (r+1) بیت را آشکارسازی نمایند. این به مفهوم آن است که همه‌ی بیت‌های خطا به‌گونه‌ی پی در پی از تعداد r بیت و یا کم تر از آن قابل آشکارسازی است. با درنظر گرفتن یک فرض مناسب طول رشته بیت بزرگ‌تر از (r+1) بیت را می‌توان با احتمال برابر $(1-0.5^r)$ آشکارسازی خطا نمود. علاوه برهمه‌ی موارد اشاره شده هریک از CRCهای استاندارد قادر به آشکارسازی تعدادی فرد از بیت‌های خطا می‌باشند.

۵-۴ پروتکل‌های دستیابی چندگانه

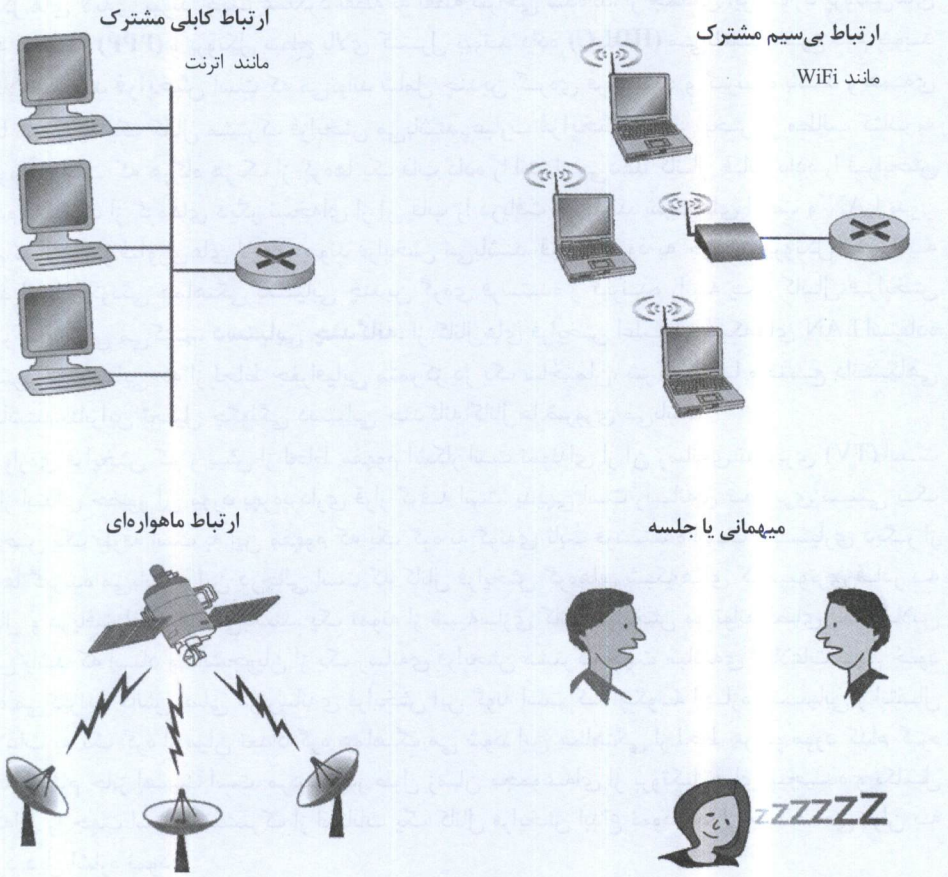
در پیش‌گفتار این فصل عنوان شد که دو نوع لایه‌ی پیوند نقطه به نقطه و فراپخش وجود دارد. یک پیوند **نقطه به نقطه** شامل فقط یک فرستنده و گیرنده در دو انتهای پیوند می‌باشد. بسیاری از

پروتکل‌های لایه‌ی پیوند جهت عملکرد نقطه به نقطه طراحی شده‌اند. از جمله‌ی این موارد پروتکل‌های نقطه به نقطه (PPP) و پروتکل سطح بالای کنترل پیوند داده (HDLC) می‌باشند. نوع دوم پیوند اصطلاحاً **پیوند فراپخش** است که می‌تواند شامل چندین گره‌ی فرستنده و گیرنده باشد و همه‌ی آن‌ها متصل به یک کانال مشترک فراپخش می‌باشند. عبارت **فراپخش** در این بخش از مطالب کتاب به مفهوم آن است که هرگاه هریک از گره‌ها یک قاب داده را انتشار می‌دهد کانال قاب داده را فراپخش نموده و هریک از گره‌های دیگر نسخه‌ای از آن قاب را دریافت می‌کنند. شبکه‌های اترنت و LAN بدون سیم نمونه‌ای از فنآوری‌های لایه‌ی پیوند فراپخش می‌باشند. قبل از ورود به جزئیات پروتکل‌های لایه پیوند ابتدا چگونگی هماهنگی دستیابی چندین گره‌ی فرستنده و گیرنده را به یک کانال فراپخش مشترک بررسی می‌کنیم، **دستیابی چندگانه**. از کانال‌های فراپخش اغلب در شبکه‌های LAN استفاده می‌شود. شبکه‌هایی که از لحاظ جغرافیایی متمرکز در یک ساختمان، شرکت و یا مجتمع دانشگاهی می‌باشند. بنابراین تحلیل چگونگی دستیابی چندگانه کانال‌ها ضروری می‌باشد.

واژه‌ی فراپخش کم و بیش از لحاظ مفهوم آشکار است نمونه‌ای از آن رسانه‌ی تصویری (TV) است که از ابتدای حضور آن مورد بهره‌برداری قرار گرفته است. بدیهی است رسانه‌ی تصویری سنتی یک فراپخش یک طرفه است به این مفهوم که یک گره به گونه‌ی ثابت فرستنده است و بسیاری دیگر از گره‌ها گیرنده می‌باشند. این درحالی است که کانال فراپخش گره‌های شبکه‌های کامپیوتری قادر به ارسال و دریافت اطلاعات می‌باشند. یک نمونه از شبیه‌سازی کانال فراپخش می‌تواند فضای یک کلاس درس باشد که استاد و دانشجویان از یک رسانه‌ی فراپخش مشترک جهت مبادله‌ی اطلاعات بین خود بهره می‌گیرند. چالش اصلی در رسانه‌ی فراپخش این گونه است که چگونه اجازه دستیابی و انتقال اطلاعات به یک گره از میان تعداد گره هماهنگ می شود. این هماهنگی از لحظ هر دو مورد کدام گره و چه هنگام حائز اهمیت است. مردمان در طول زمان مجموعه‌ای از پروتکل‌های پیچیده و تکامل یافته‌ای را جهت استفاده مشترک از امکانات یک کانال فراپخش ابداع نموده‌اند. از جمله می‌توان به موارد ذیل اشاره نمود:

- هر فرد دارای فرصت یکسان جهت گفتگو گردد.
- سخن نگوئید تا هنگامی که با شما سخن نگفته باشند.
- گفتگو و محاوره را در انحصار خود قرار ندهید.
- با بهره‌گیری از علامت دست و اشاره پرسش خود را مطرح کنید.
- هنگام سخن گفتن فرد، ایجاد وقفه مجاز نمی‌باشد.
- در هنگام سخن گفتن یک فرد خواب آلودگی مجاز نمی‌باشد.

به‌گونه‌ی مشابه شبکه‌های کامپیوتری نیز از پروتکل‌هایی جهت تعامل بهره می‌گیرند. پروتکل‌هایی که اصطلاحاً **پروتکل‌های دستیابی چندگانه** نامیده شده‌اند و از طریق آن‌ها گره‌ها انتقال اطلاعات خود به درون کانال فراپخش مشترک را تنظیم و هماهنگ می‌کنند. همان‌گونه که از شکل ۵-۹ ملاحظه می‌شود پروتکل‌های دستیابی چندگانه در مجموعه‌ی وسیعی از شبکه‌های گوناگون مورد نیاز می‌باشند. این شبکه‌ها شامل هر دو حالت محلی باسیم و بدون سیم می‌باشند. در قالب وسیع‌تر شبکه‌های ارتباطی را از هر نوع می‌توانند در پوشش خود قرار دهند. اگرچه از لحاظ فنی هر گره‌ی کانال فراپخش را می‌تواند از طریق کارت واسط خود دستیابی نماید اما در این بخش واژه‌ی **گره** به مفهوم تجهیزات ارسال و دریافت می‌باشد. در عمل صدها و حتی هزاران گره می‌توانند از طریق یک کانال فراپخش مستقیماً با یکدیگر ارتباط برقرار نمایند.



شکل ۵-۹ ♦ کانال‌های دستیابی چندگانه‌ی گوناگون

از آن‌جایی‌که همه‌ی گره قادر به انتقال فریم‌ها می‌باشند بنابراین ممکن است بیش از دو گره نسبت به انتقال قاب‌های داده‌ی خود اقدام نمایند. هنگامی که چنین وضعیتی روی دهد همه‌ی گره‌ها چندین فریم را همزمان دریافت می‌دارند، به عبارت دیگر قاب‌های انتشار یافته با یکدیگر در همه‌ی گره‌های گیرنده **برخورد**[1] می‌کنند. هنگامی که برخورد فریم‌ها روی می‌دهد هیچ یک از گره‌ها قادر به تفسیر مفهوم آن‌ها نمی‌باشند در نتیجه همگی قاب‌های درگیر با یکدیگر تلف می‌شوند و کانال فراپخش در طول دوره‌ی برخورد از لحاظ ظرفیت و بهره‌وری بی فایده خواهد شد. بدیهی است چنان‌چه تعداد قابل توجهی از گره‌ها مرتباً مایل به انتقال فریم باشند بسیاری از انتقال‌ها با برخورد مواجه می‌شوند و پهنای باند قابل توجهی از کانال فراپخش تلف می‌شود.

جهت حصول اطمینان از بهره‌وری مفید کانال فراپخش در هنگام فعال شدن چندین گره لازم است به نوعی هماهنگی انتقال در بین گره‌های فعال انجام شود. این وظیفه‌ی هماهنگی برعهده‌ی پروتکل‌های کنترل دستیابی چندگانه گذارده شده است. در طول 40 سال گذشته هزاران مقاله و

1- Collide

پژوهش در ارتباط با این پروتکل‌ها به نگارش درآمده است. هم اکنون نیز پژوهش‌های فعال و پویا در این زمینه ادامه دارد که نتیجه‌ی آن‌ها پیوندهای جدید به ویژه در ارتباط با شبکه‌های بدون کابل (WiFi) می‌باشد.

در طول سالیان تعداد قابل توجهی از پروتکل‌های دستیابی چندگانه برروی فناوری‌های گوناگون لایه‌ی پیوند پیاده‌سازی شده‌اند. به طور کلی هریک از پروتکل‌های دستیابی چندگانه را می‌توان در قالب یکی از سه گروه ذیل خلاصه نمود:

- **پروتکل‌های تقسیم‌بندی کانال**[1] **(CPP)**
- **پروتکل‌های دستیابی تصادفی**[2] **(RAP)**
- **پروتکل‌های نوبت‌بندی**[3] **(TTP)**

این سه گروه پروتکل‌ها در ادامه‌ی مطالب فصل بررسی و تحلیل خواهند شد. اما قبل از ورود به آن‌ها به طور کلی هر پروتکل دستیابی چندگانه‌ی فراپخش با درنظر گرفتن کانال فراپخش با سرعت R بیت بر ثانیه لازم است ویژگی‌های ذیل را همراه خود داشته باشد:

۱- هنگامی که فقط یک گره داده جهت گسیل دارد آن گره فقط دارای نهفتگی R بیت برثانیه باشد.

۲- هنگامی که تعداد M گره داده جهت گسیل دارند هریک از آن‌ها دارای نهفتگی $\frac{R}{M}$ بیت برثانیه می‌شود. این به مفهوم آن نیست که هریک از M گره همیشه یک سرعت لحظه‌ای $\frac{R}{M}$ بیت برثانیه دارد بلکه هر گره بایستی شامل میانگین سرعت انتقال $\frac{R}{M}$ در هر فاصله‌ی زمانی مناسب و معقول باشد.

۳- پروتکل دستیابی غیر متمرکز است، به عبارت دیگر هیچ گره‌ای نماینده‌ی نقطه عیب شبکه از سوی بقیه‌ی گره‌ها نمی‌باشد.

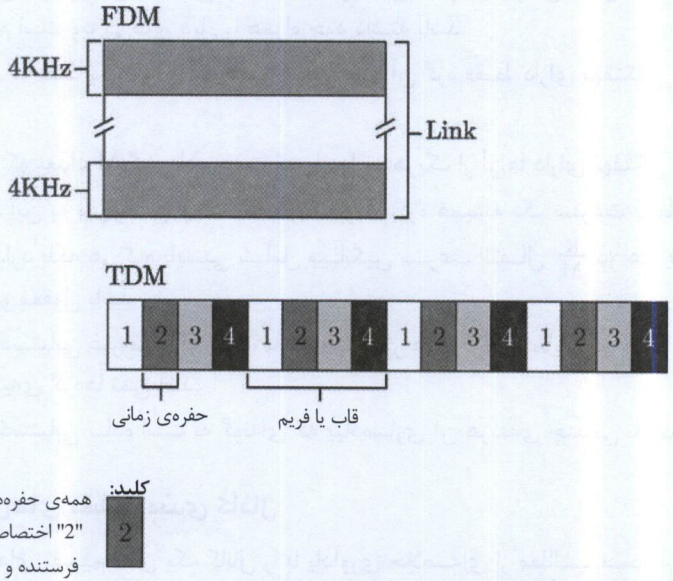
۴- پروتکل دستیابی ساده است به گونه‌ای که پیاده‌سازی آن هزینه‌ی چندانی ندارد.

• پروتکل‌های تقسیم‌بندی کانال

موضوع پروتکل‌های تقسیم‌بندی یک کانال را با یادآوری خلاصه‌ای از مطالب تسهیم تقسیم زمانی (TDM) و تقسیم بسامد (FDM) از فصل اول آغاز می‌کنیم. این دو روش می‌تواند جهت تقسیم‌بندی پهنای باند یک کانال فراپخش در بین همه‌ی گره‌هایی سودمند باشد که از آن کانال به‌صورت مشترک بهره می‌برند. به عنوان نمونه اگر فرض شود یک کانال با سرعت R bps قادر به پشتیبانی تعداد N گره می‌باشد. آن‌گاه در تسهیم تقسیم زمانی (TDM) زمان در قالب قاب‌های زمانی تقسیم خواهد شد که قاب‌های زمانی را نیز می‌توان بین تعداد N حفره‌ی زمانی تقسیم نمود. یادآوری این نکته ضروری است که مفهوم قاب‌های زمانی TDM نبایستی با قاب‌های داده در لایه‌ی پیوند که بین فرستنده‌ها و گیرنده‌ها مبادله می‌شوند یکسان فرض شوند. به همین دلیل و به منظور جلوگیری از تلقی خطا در مورد این دو واژه در این زیربخش از قاب داده‌ی لایه‌ی پیوند با عنوان یک **بسته‌ی داده** نام برده

1- Channel partioning protocols
2- Random access protocols
3- Taking turns protocols

می‌شود. به هر روی هر شیار زمانی پس از تشکیل، به یکی از N گره اختصاص پیدا می‌کند. هنگامی که یک گره بسته‌ی داده‌ای را جهت ارسال دارد بیت‌های متعلق به آن بسته‌ی داده را در فاصله‌ی زمانی که حفره زمانی درون قاب TDM به آن اختصاص می‌یابد، انتقال می‌دهد. بدیهی است قاب TDM از لحاظ زمانی ثابت نمی‌باشد و مرتباً در فاصله هریک ثانیه ممکن است K مرتبه تغییر نماید. در عمل اندازه‌ی هر شیار زمانی (TS) به‌گونه‌ای گزینش می‌شود که در آن فاصله‌ی زمانی فقط یک بسته منتقل گردد. شکل ۵-۱۰ نمونه‌ای از یک TDM با تعداد 4 گره را نشان می‌دهد. اگر به مثال کلاس درس بازگشت کنیم در یک کلاس درس که از طریق روش TDM کنترل می شود به هر دانشجو اجازه داده می‌شود که در طول یک فاصله‌ی زمانی معین پرسش خود را مطرح نماید و سپس همین اندازه‌ی زمانی به دانشجوی بعدی و غیره اختصاص می‌یابد. بنابراین هنگامی که هر عضو کلاس مجال سخن گفتن را دریافت نمود الگو مجدداً تکرار می‌شود.



شکل ۵-۱۰ ♦ نمونه‌ای از دو روش TDM و FDM با تعداد چهارگره

روش TDM از آن جهت جذاب است که از وقوع برخورد داده‌ها جلوگیری می‌کند. از طرف دیگر حالتی کاملاً عادلانه دارد زیرا هر گره سرعت انتقال انحصاری معادل $\frac{R}{N}$ bps را در طول زمان هر قاب دریافت می‌کند. اما از دو نقطه ضعف مهم رنج می‌برد. نخست آن‌که یک گره محدود به یک سرعت میانگین $\frac{R}{N}$ bps می‌شود حتی هنگامی که تنها گره‌ای است که بسته‌ی داده جهت گسیل دارد. دوم آن‌که هر گره همواره بایستی در انتظار نوبت جهت فرآیند انتقال قرار گیرد حتی اگر آن گره تنها گره‌ای است که بسته‌ی داده جهت گسیل در اختیار دارد. بدیهی است با توجه به این کاستی‌ها روش TDM می‌تواند گزینش ضعیفی جهت پروتکل‌های دستیابی چندگانه باشد. درحالی‌که TDM امکانات کانال فراپخش را در یک دوره‌ی زمانی به اشتراک می‌گذارد اما FDM سرعت Rbps که یک کانال

می‌تواند اختیار نماید را در قالب بسامدهای مختلف تقسیم می‌کند. هر بسامد پهنای باند معادل $\frac{R}{N}$ را دارد و هریک از بسامدها به یک گره اختصاص می‌یابد. بنابر این روش FDM تعداد N کانال کوچک‌تر را خلق می‌کند که هریک از آن‌ها سرعت معادل $\frac{R}{N}$ bps را از بین نرخ انتقال کلی Rbps اختیار می‌کند. روش تسهیم تقسیم بسامد از هر دو نقطه‌ی قوت و ضعف TDM بهره برده است. به عبارت دیگر از وقوع برخورد داده‌ها جلوگیری می‌کند و پهنای باند کانال را نیز به‌گونه‌ای عادلانه مابین تعداد N گره تقسیم می‌کند. اما علاوه بر موارد اشاره شده FDM ضعف دیگری نیز نسبت به TDM دارد و آن عبارت است از این که یک گره محدود به یک پهنای باند $\frac{R}{N}$ می‌شود حتی اگر آن تنها گره‌ای است که بسته‌ی داده جهت ارسال در اختیار دارد.

نوع سوم پروتکل تقسیم‌بندی کانال، **دستیابی چندگانه‌ی تقسیم کد**[1] **(CDMA)** می‌باشد. در حالی‌که دو روش TDM و FDM به ترتیب حفره‌های زمانی و بسامدها را به گره‌ها اختصاص می‌دهند. روش CDMA به هر گره یک رمز یا **کد** اختصاص می دهد. هر گره سپس از رمز انحصاری خود جهت رمزگشایی بیت‌های داده‌ی در حال ارسال خود بهره می‌گیرد. اگر کدها با دقت گزینش شده باشند، شبکه‌های CDMA از این ویژگی برجسته بهره‌مند می‌شوند که گره‌های گوناگون می‌توانند **همزمان** به انتقال بسته‌های داده مبادرت نمایند و این در حالتی است که هریک از گیرنده‌های آن‌ها نیز داده‌ی رمزگشایی شده را با وجود تداخل انتقال از سوی گره‌های دیگر بدون خطا دریافت نمایند. بدیهی است که گیرنده بایستی از رمز فرستنده آگاه باشد. از روش CDMA اغلب در کاربردهای نظامی استفاده می‌شود. علت این موضوع مقاومت این گونه رمزها در برابر پارازیت و انسداد می‌باشد. بدیهی است در حال حاضر به صورت گسترده از CDMA در شبکه‌های غیرنظامی به ویژه شبکه‌های تلفنی سیار نیز استفاده می‌شود. از آن‌جائی‌که بهره‌گیری از CDMA وابستگی بسیاری به کانال‌های بی‌سیم دارد در فصل ۶ جزئیات بیش‌تری از CDMA ارائه خواهد شد. در این مقطع کافی است کدهای CDMA را مانند شیارهای زمانی در TDM و بسامدها در FDM قلمداد نمود که آن‌ها را می‌توان به کاربران کانل‌های دستیابی چندگانه اختصاص داد.

- **پروتکل‌های دستیابی تصادفی**

گروه دوم از پروتکل‌های دستیابی چندگانه معروف به پروتکل‌های دستیابی تصادفی می‌باشند. در چنین پروتکل‌هایی یک گره‌ی گسیل کننده همواره با سرعت کامل کانال یعنی Rbps مبادرت به انتقال داده‌ها می‌کند. هنگامی که برخورد یا انسداد روی داده باشد هر گره‌ی عامل برخورد داده‌ها بسته‌های خود را مرتباً تکرار می‌کند تا آن هنگام که بسته بدون برخورد با دیگرین از درون کانال عبور نماید. اما هنگامی که یک گره برخورد داده را تجربه می کند لزوماً و بی‌درنگ به تکرار بسته‌های داده (قاب‌ها) مشغول نمی‌شود بلکه **یک مدت زمان تصادفی منتظر می‌ماند و سپس بسته‌ها را مجدداً انتقال می‌دهد.** هر گره‌ی ذینفع در برخورد داده‌ها زمان انتظار تصادفی خویش را مستقل از دیگر گره‌های شامل برخورد گزینش می‌کند. از آن‌جائی‌که تأخیری را گزینش نماید که از لحاظ مدت زمان در حد کفایت کم‌تر از تأخیرهای دیگر گره‌های عامل برخورد خواهد بود و بنابراین قادر به گسیل بسته‌ی خود به درون کانال بدون برخورد و یا انسداد می‌شود.

ده‌ها و شاید صدها پروتکل دستیابی تصادفی تا امروز تعریف و پیشنهاد شده‌اند اما در این بخش از

1- Code division multiplexing

مطالب از میان این انبوه پیشنهادها چند مورد آن‌ها که عبارتند پروتکل‌های **ALOHA** (به مفهوم خداحافظی در گویش مردمان جزایر هاوایی است)، **CSMA**[1] و غیره بررسی می‌شوند.

۱- پروتکل شیاردار ALOHA

یکی از ساده‌ترین پروتکل‌های دستیابی تصادفی پروتکل **شیاردار ALOHA**[2] می‌باشد. در این پروتکل موارد ذیل جهت تحلیل آن درنظر گرفته می‌شود:

- همه‌ی بسته‌های داده (قاب‌ها) هرکدام طول برابر L بیت اختیار می‌کنند.
- زمان انتقال در بین حفره‌ها و یا شیارها به طول $\frac{L}{R}$ ثانیه تقسیم می‌شوند. به عبارت دیگر هر حفره یا شیار زمانی معادل انتقال یک قاب را اختیار می‌کند.
- فقط از آغاز هر حفره گره‌ها شروع به انتقال قاب‌ها می‌کنند.
- همه‌ی گره‌ها به‌گونه‌ای همزمان می‌شوند تا هر گره از آغاز هر حفره آگاه شود.
- اگر دو یا چندین قاب با یکدیگر در یک حفره برخورد نمایند آن‌گاه همه‌ی گره‌ها وقوع برخورد را قبل از انتهای حفره مشاهده می‌کنند.

اگر فرض شود **p** یک احتمال با ارزش مابین صفر و یک باشد آن‌گاه عملیات پروتکل حفره‌دار ALOHA در هر گره را می‌توان به‌صورت ساده چنین تصور نمود:

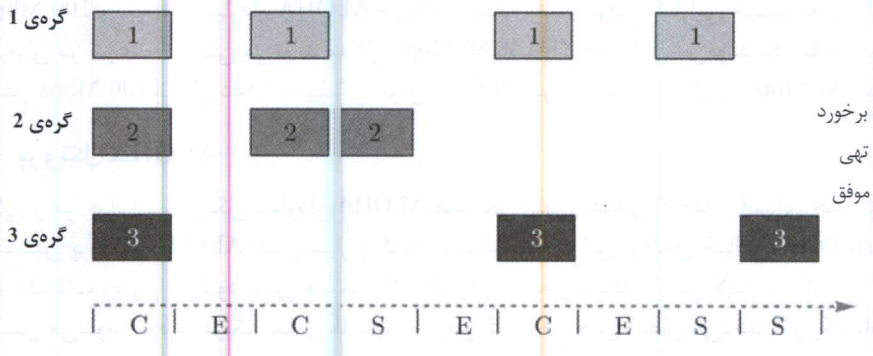
- هنگامی که یک گره قاب جدیدی جهت گسیل دارد آن گره تا آغاز شیار بعدی منتظر می‌شود و سپس تمامی قاب را به درون حفره انتقال می‌دهد.
- اگر برخوردی مشاهده نشد آن گره قاب را با موفقیت گسیل نموده است و بنابراین نیاز به انتقال مجدد قاب نمی‌باشد. گره می‌تواند قاب جدیدی را جهت انتقال آماده‌سازی نماید.
- در صورت وقوع برخورد، گره آن را قبل از انتهای حفره شناسایی می‌کند. گره سپس قاب خود را مجدداً گسیل می‌کند. چگونگی انتقال قاب به این گونه است که قاب مورد نظر گره در هر شیار با احتمال p که انتقال را بدون برخورد انجام می‌دهد صورت می‌پذیرد.

مفهوم انتقال مجدد احتمال p عبارت از آن است که با وقوع p قاب مجدداً منتقل می‌شود و در صورت وقوع (1-p) از شیار در اختیار صرف‌نظر شده و مجدداً احتمال p محاسبه می‌شود و این عمل تا آن‌جا ادامه می‌یابد که سرانجام در احتمال p حفره‌ای گزینش شود. پروتکل شیاردار ALOHA به نظر می‌رسد ویژگی‌های زیادی را در درون خود حمل می‌کند. از جمله می‌توان به موارد ذیل اشاره نمود:

- برعکس پروتکل تقسیم‌بندی کانال، پروتکل شیاردارد ALOHA به یک گره اجازه می‌دهد با سرعت کامل Rbps از کانال هنگامی که آن گره فعال است به صورت مستمر به انتقال ادامه دهد. منظور از گره‌ی فعال وجود قاب جهت ارسال می‌باشد.
- پروتکل **شیاردار ALOHA** به صورت گسترده‌ای غیر متمرکز می باشد زیرا هر گره قادر به آشکارسازی برخورد است و نسبت به زمان ارسال مجدد مستقل تصمیم‌گیری می‌کند. بدیهی است جهت این توانمندی پروتکل شیاردار ALOHA نیازمند همزمانی حفره‌ها در گره‌ها می‌باشد.
- پروتکل شیاردار ALOHA بسیار ساده است.

1- Carrier sense multiple access
2- Slotted ALOHA

به طور کلی این پروتکل هنگامی که یک گره فقط فعال است می‌تواند کارآمد به شمار آید اما میزان کارآمدی آن هنگامی که چندین گره‌ی فعال وجود دارد چه‌مقدار است؟ در این حالت دو عامل نگران کننده در ارتباط با کارآمدی به وجود می‌آید. نخست آن که برمبنای شکل ۵-۱۱ هنگامی که چندین گره‌ی فعال وجود دارند بخشی از حفره‌ها دچار برخورد می‌شوند و بنابراین هدر می‌روند. دوم آن که بخش دیگری از حفره‌ها تهی می‌مانند زیرا همه‌ی گره‌های فعال دست به عملیات انتقال به علت وقوع احتمال (1-p) نمی‌زنند. بنابراین تنها حفره‌های بهره گرفته شده، شامل آن‌هایی می‌شود که در حقیقت فقط یک گره انتقال داده است. شیاری که در آن فقط یک گره انتقال را انجام دهد معروف به **شیار موفق**[1] می‌باشد. کارایی یک پروتکل دستیابی چندگانه‌ی شیاردار به‌گونه‌ای طراحی می شود که شامل بخش وسیعی از شیارهای موفق در حالتی باشد که تعداد گره‌های فعال بسیار چشم‌گیر است. به عبارت دیگر هریک از آن‌ها دارای تعداد انبوهی قاب جهت گسیل می‌باشد. بدیهی است اگر هیچ‌گونه کنترل دستیابی به کار گرفته نشود و هر گره بی‌درنگ نسبت به انتقال مجدد پس از وقوع هر برخورد اقدام می‌نمود آن‌گاه کارایی نزدیک به صفر تقلیل می‌یابد. بدیهی است پروتکل شیاردار ALOHA کارایی به مراتب بیش‌تر از صفر را شامل می‌شود اما مقدار آن چیست؟



شکل ۵-۱۱ ♦ در نخستین شیار گره‌های 1، 2 و 3 برخورد می‌کنند. گره‌ی 2 سرانجام موفق به ارسال در شیار چهارم می‌شود و گره 1 به همین وضعیت در شیار 8 و گره‌ی 3 در شیار 9 دستیابی پیدا می‌کند.

در ادامه‌ی بررسی پروتکل شیاردار ALOHA تلاش می‌شود رابطه‌ای جهت دستیابی به حداکثر کارایی تعریف گردد. به منظور ساده شدن این رابطه تغییراتی در پروتکل اعمال می‌شود و فرض می‌گردد که هر گره تلاش به ارسال مجدد یک قاب در هر حفره با احتمال p می‌کند. به عبارت دیگر فرض می‌شود که هر گره همواره یک قاب جهت ارسال دارد و آن گره این عمل را با احتمال p برای یک قاب جدید و قاب‌های دیگری که قبلاً دچار برخورد شده‌اند، انجام می‌دهد. اگر فرض شود تعداد گره‌ها N باشد آن‌گاه احتمال این که یک شیار موفق تلقی شود برابر است با احتمال آن که یکی از گره‌ها انتقال انجام دهد و بقیه‌ی N-1 گره انتقالی انجام ندهند. احتمال آن که یک گره عمل انتقال را انجام دهد p می‌باشد و احتمال آن که بقیه‌ی گره‌ها عملیات انتقال را انجام ندهند برابر $(1-p)^{N-1}$ است. بنابراین احتمال موفقیت انتقال یک گره برابر است با رابطه‌ی (۵-۵):

1- Successful slot

$$\text{احتمال شیار موفق} = p\,(1-p)^{N-1} \qquad (۵-۵)$$

بنابراین هنگامی که تعداد گره‌های فعال N باشد، کارایی پروتکل شیاردار ALOHA از طریق رابطه‌ی (۵-۶) تعریف می‌شود:

$$\text{کارایی پروتکل شیاردار ALOHA} = Np(1-p)^{N-1} \qquad (۵-۶)$$

برای آن که حداکثر کارایی تعداد N گره‌ی فعال به‌دست آید نیازمند یافتن مقدار *p جهت ماگزیموم عبارت (۵-۶) می‌باشیم. بنابراین جهت یافتن حداکثر کارایی برای تعداد انبوهی گره‌ی فعال حدعبارت (۵-۷) در فاصله‌ی N از یک تا بی‌نهایت محاسبه می‌شود:

$$\text{حداکثر کارایی} = \lim_{N\to\infty} Np^*\,(1-p^*)^{N-1} \qquad (۵-۷)$$

پس از انجام محاسبات ریاضی حداکثر کارایی پروتکل شیاردار ALOHA برابر e^{-1} یعنی 0.37 می‌شود. به‌عبارت دیگر هنگامی که تعداد انبوهی از گره‌ها قاب‌های پرشماری را جهت انتقال دارند آن‌گاه تعداد شیارهای موفق تنها 37% است. بنابراین نرخ مؤثر انتقال کانال Rbps نمی‌باشد بلکه 0.37R)bps) است! از طریق تحلیل مشابه می‌توان نشان داد که فقط 37%از حفره‌ها بدون قاب انتقال می‌یابند و 26% نیز دچار برخورد می‌شوند. با اندکی تأمل می‌توان دریافت که اگر کانالی با سرعت 100 Mbps و سیستم شیاردار ALOHA خریداری شده باشد توقع کارایی جهت تعداد انبوهی از کاربران در سرعت انباشتی می‌تواند معادل 80 Mbps باشد! اگرچه کانال می‌تواند یک قاب را در سرعت کامل 100 Mbps انتقال دهد اما نهفتگی موفق این کانال در دراز مدت کم‌تر از 37 Mbps است.

۲- پروتکل ALOHA

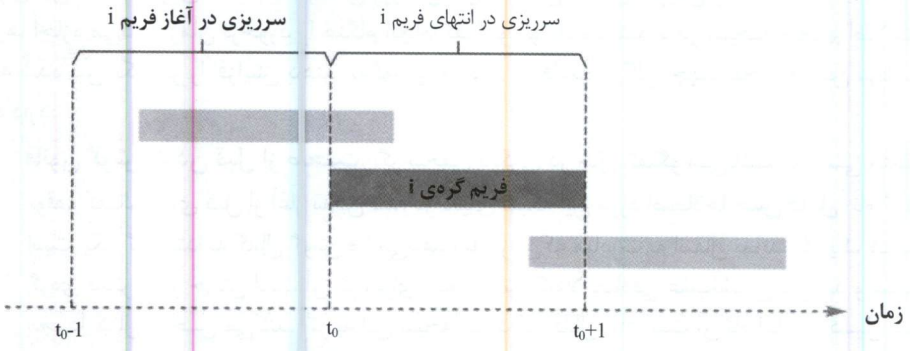
یکی از موارد الزامی پروتکل شیاردار ALOHA همزمانی انتقال همه‌ی گره‌ها از ابتدای حفره زمانی بود. نخستین پروتکل ALOHA بدون شیار و کاملاً متمرکز بود. در این پروتکل خالص ALOHA هنگامی که یک قاب وصول می‌شود یعنی هنگامی که داده گرام لایه‌ی شبکه از این لایه به گره‌ی فرستنده گسیل می‌شود، گره بی‌درنگ تمامی قاب را به درون کانال فراپخش انتقال می‌دهد. اگر یک قاب انتقال یافته دچار برخورد با یک یا چندین انتقال دیگر شود آن گره بی‌درنگ پس از انتقال کامل قاب آلوده از برخورد مجدداً فریم زمانی را با احتمال p انتقال می‌دهد، در غیر این‌صورت گره منتظر زمان انتقال قاب می‌شود. پس از گذر این زمان انتظار قاب را با احتمال p انتقال می‌دهد و در صورت وجود احتمال (1-p) بار دیگر منتظر زمان انتظار بعدی فریم می‌شود.

برای محاسبه‌ی حداکثر بهره‌وری پروتکل خالص ALOHA تمرکز برروی هر گره خواهد بود. کلیه‌ی مفروضات منظور شده در محاسبه‌ی کارایی پروتکل شیاردار ALOHA در تحلیل پروتکل خالص ALOHA نیز درنظر گرفته می‌شود. زمان انتقال فریم نیز برابر یک یکای زمانی فرض می‌شود. با توجه به این موارد در هر لحظه‌ی زمانی یک گره قابی را با احتمال p انتقال می‌دهد. اگر فرض شود ابتدای انتقال فریم در لحظه‌ی t_0 باشد، براساس آن‌چه شکل ۵-۱۲ نشان می‌دهد انتقال قاب وقتی کاملاً با موفقیت همراه خواهد شد که گره‌های دیگر در فواصل زمانی $[t_0\text{-}1,t_0]$ انتقالی را انجام ندهند، در غیر این صورت چنین انتقالی سبب سرریزی برروی نقطه‌ی آغاز انتقال قاب در گره‌ی I خواهد شد. احتمال آن که هیچ کدام از گره‌های دیگر نیز در این فاصله‌ی زمانی مبادرت به انتقال نمی‌کنند برابر $(1-p)^{N-1}$ است. به‌گونه‌ی مشابه هنگامی که گره‌ی i در حال انتقال است هیچ یک از گره‌های دیگر نیز نمی‌توانند انتقالی را انجام دهند زیرا چنین انتقالی سبب سرریزی انتقال گره‌ی i خواهد شد. در این حالت احتمال

آن که همه‌ی گره‌های دیگر نیز آغاز انتقال خود را در چنین فاصله‌ی زمانی انجام ندهند برابر $(1-p)^{N-1}$ می‌باشد. در نتیجه رابطه‌ی (۵-۸) احتمال آن‌که یک گره کاملاً موفق به انتقال قاب خود شود را تعریف می‌کند.

$$\text{احتمال انتقال موفق خالص ALOHA} = p(1-p)^{2N-1} \quad (۵-۸)$$

اگر حد عبارت (۵-۸) برای مقدار بی‌نهایت N محاسبه گردد، حداکثر کارایی پروتکل خالص ALOHA برابر نصف رقیب آن یعنی پروتکل شیاردار ALOHA خواهد شد. به عبارت دیگر حداکثر کارایی پروتکل خالص برابر $\frac{1}{2e}$ یا تقریباً 19% می‌شود.



شکل ۵-۱۲ ♦ تداخل انتقال در پروتکل خالص ALOHA

• پیشینه‌ی تاریخی ALOHA

از آن‌جایی‌که هاوایی شامل تعداد زیادی جزایر کوهستانی است، این ساختار جغرافیایی مانع از نصب و به‌کارگیری شبکه‌های کابلی جهت انتقال داده‌ها می‌باشد. بنابراین، در سال 1969 شخصیتی به نام ابرامسن از دانشگاه هاوایی به فکر طراحی یک شبکه‌ی راهیابی بسته‌های داده از طریق سیستم رادیویی یا بدون کابل افتاد. نتیجه‌ی تلاش او طراحی شبکه‌ای در مجمع جزایر هاوایی شد که از یک میزبان اولیه‌ی مرکزی به همراه چندین گره‌ی پراکنده‌ی ثانوی تشکیل شده بود. این شبکه دارای دو کانال انتقالی با بسامدهای مختلف بود. یک کانال ارسال بسته‌ها را از میزبان مرکزی به دیگر گره‌ها فراپخش می‌نمود و کانال دوم گسیل بسته‌ها از هر یک از گره‌های ثانوی به میزبان مرکزی را انجام می‌داد. علاوه بر ارسال بسته‌های داده، کانال فراپخش میزبان نسبت به ارسال پیام‌های تائید برای هر بسته‌ی داده که از سری گره‌های ثانوی با موفقیت دریافت می‌گردید نیز می‌شد. از آن‌جائی‌که میزبان‌های ثانوی انتقال بسته‌ها را به‌صورت غیرمتمرکز انجام می‌دادند وقوع برخورد در کانال ارسال به طرف میزبان اولیه اجتناب‌ناپذیر می‌شد. این موضوع سبب طراحی پروتکلی از سوی ابرامسن برای ممانعت از برخورد گردید که وی آن را ALOHA نامید که در گویش مردمان هاوایی به مفهوم بدرود می‌باشد. این تلاش همان پروتکل خالص ALOHA می‌باشد که در بخش قبل تحلیل شد. بعدها شخصیت دیگری به نام مت‌کالفه با الهام‌گیری از این پروتکل روش دیگری را نوآوری نمود که امروز به آن پروتکل CSMA/CD در شبکه‌های محلی اترنت گفته می‌شد.

۳- پروتکل دستیابی چندگانه‌ی حسی حامل با آشکارسازی برخورد (CSMA/CD)

در هر دو پروتکل شیاردار و خالص ALOHA تصمیم یک گره جهت انتقال، مستقل از فعالیت گره‌های دیگر متصل به کانال فراپخش انجام می‌گرفت. به‌ویژه آن که نخست یک گره توجهی به این موضوع نمی‌نمود که گره‌های دیگری ممکن است در حال انتقال قاب باشند و دوم آن‌که اگر گره‌ی دیگری اختلال در جریان انتقال قاب‌ها را سبب می‌شد گره‌ی فعال انتقال خود را متوقف نمی‌کرد. در مثال کلاس درس که از پروتکل ALOHA بهره می‌گیرد یک دانشجو ممکن است گفتگویی را آغاز نماید بدون توجه به این نکته که قبل از وی نفر دیگر ممکن است در حال گفتگو در کلاس باشد. البته یادآوری این نکته ضروری است که مردمان پروتکل‌های گفتمان ویژه‌ی خویش را تعریف نموده‌اند که به آن‌ها اجازه می‌دهد زمان برخورد را هنگام انجام گفتگو کوتاه نمایند و در نتیجه حجم اطلاعات مبادله شده بین یکدیگر را افزایش دهند. به‌گونه‌ی ویژه‌ای دو قاعده‌ی کلی جهت محاوره بین مردمان وجود دارد:

۱- **قانون گوش کردن قبل از صحبت.** اگر شخص دیگری در حال گفتگو می‌باشد بایستی منتظر توقف گفتگوی وی قبل از آغاز سخن شد. در دنیای شبکه این مورد اصطلاحاً **حس‌حامل**[1] نام گرفته است. یک گره ابتدا به کانال گوش فرا می‌دهد قبل از آن‌که مبادرت به انتقال نماید. اگر یک قاب از گره‌ی دیگری در جریان است آن گره برای مدت زمانی کاملاً تصادفی عقب‌نشینی می‌کند و سپس مجدداً کانال را حس می‌کند. اگر به این نتیجه رسید که کانال آزاد است آن‌گاه آغاز به گسیل قاب خود می‌کند در غیر این صورت بازهم برای مدت زمان تصادفی دیگری به انتظار می‌رود.

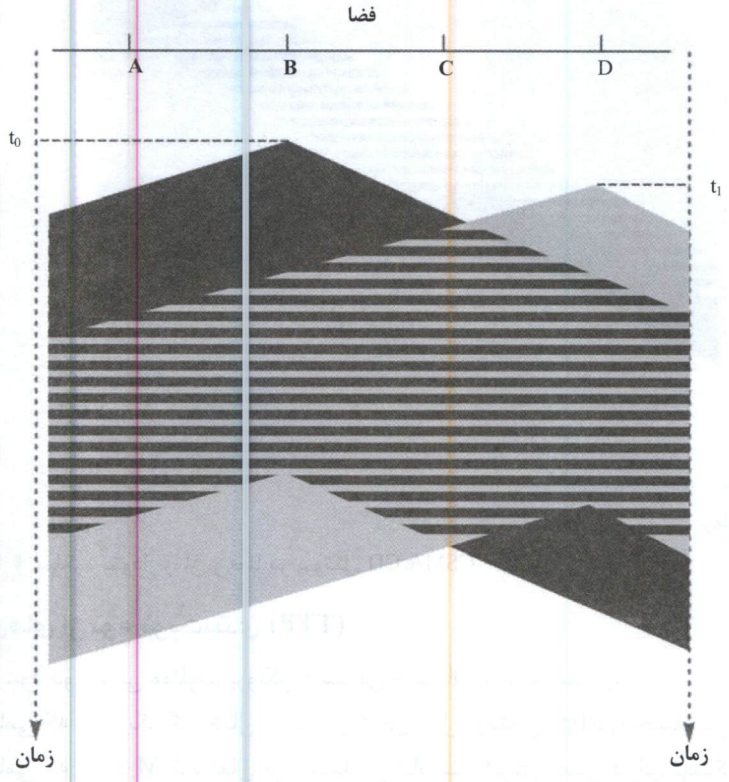
۲- **قانون توقف سخن اگر فرد دیگری همزمان وارد گفتگو شود.** در دنیای انسان‌ها اگر دو فرد همزمان آغاز به سخن نمایند برمبنای رعایت ادب هر دو ناگهانی سخن خود را قطع می‌کنند. در دنیای شبکه این مورد اصطلاحاً **آشکارسازی برخورد**[2] نامیده می‌شود. یک گره هنگام انجام عملیات انتقال قاب به کانال گوش فرا می‌دهد. اگر تشخیص داد که گره‌ی دیگری همزمان در حال ارسال قاب و در نتیجه تداخل در فریم او می‌باشد آن‌گاه ارسال قاب را متوقف و از یک پروتکل ویژه جهت تعیین تلاش بعدی خود جهت گسیل قاب بهره می‌گیرد.

این دو قانون در خانواده‌ای از پروتکل‌های شبکه جایگذاری شده‌اند که اصطلاحاً به پروتکل‌های **CSMA/CD**[3] معروف می‌باشند و مفهوم آن همان‌گونه که قبلاً اشاره گردید دستیابی چندگانه‌ی حس‌حامل با آشکارسازی برخورد می‌باشد. ویژگی‌های پروتکل‌های CSMA/CD در ادامه‌ی مطالب این زیربخش تحلیل می‌شوند.

شاید نخستین پرسشی که برای یک فرد مطرح می‌شود این موضوع باشد که اگر همه‌ی گره‌ها عملیات حس کانال را انجام می‌دهند چرا هنوز برخورد قاب روی می‌دهد؟ این پرسش را می‌توان با بهره‌گیری از یک نمودار در مقابل دو محور مختصات زمان و فضا پاسخ داد. شکل ۵-۱۳ نموداری از چهار گره‌ی A، B، C و D در مقابل دو محور مختصات زمان و فضا می‌باشد که متصل به یک گذرگاه فراپخش خطی می‌باشند. محور افقی نمودار، موقعیت هر گره در فضا را نشان می‌دهد و محور عمودی نیز زمان را معرفی می‌کند.

1- Carrier Sense
2- Collision detection
3- Carrier sense multiple access with collision detection

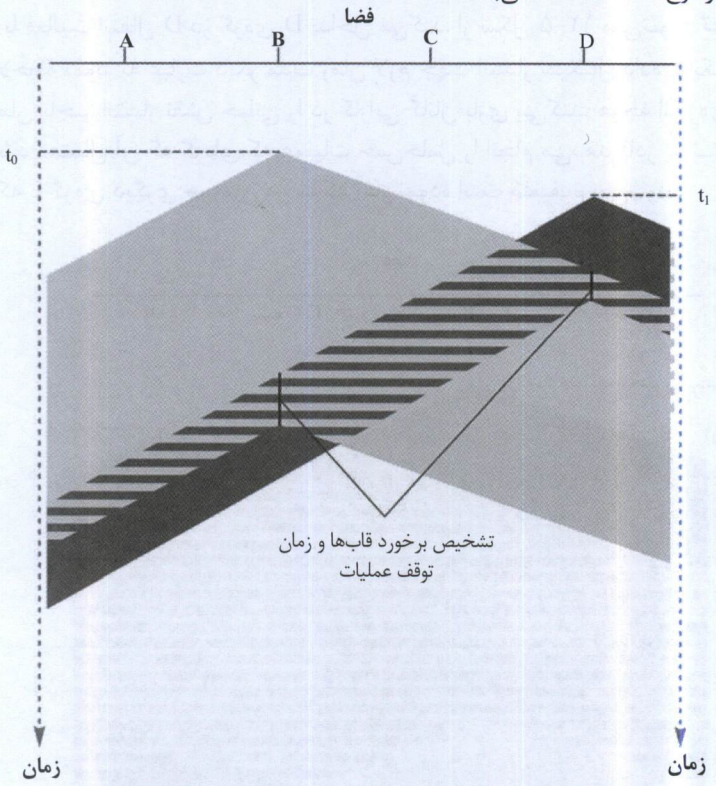
در زمان t_0 گره‌ی B تشخیص می‌دهد که کانال آزاد است از آن‌جایی که گره‌ی دیگر در حال انتقال نمی‌باشد گره‌ی B آغاز به انتقال قاب خود می‌کند با توجه به این نکته که بیت‌های آن در هر دو جهت امتداد رسانه‌ی فراپخش انتشار می‌یابند. حرکت بیت‌های گره‌ی B به طرف پائین در شکل ۵-۱۳ این موضوع را نشان می‌دهد که افزایش زمان به مفهوم آن است که مدت زمانی جهت انتشار بیت‌های B در امتداد طول رسانه لازم است اگرچه سرعت انتشار می‌تواند برابر سرعت نور فرض گردد. در لحظه‌ی t_1 یعنی $(t_1>t_0)$ گره‌ی D قاب جهت ارسال دارد. اگرچه گره‌ی B در حال حاضر مشغول انتقال در لحظه‌ی t_1 است اما بیت‌های متعلق به B که در حال انتقال می‌باشند بایستی با گذر یک فاصله‌ی زمانی خود را به D برسانند. بنابراین در لحظه‌ی t_1 برای گره‌ی D این حس پدیدار می‌شود که کانال آزاد است. بنابر پروتکل CSMA گره‌ی D آغاز به گسیل قابل می‌کند. در فاصله‌ی زمانی کوتاهی پس از آن انتقال بیت‌های B با فعالیت انتقال D در گره‌ی D تداخل می‌کند. از شکل ۵-۱۳ می‌توان **تأخیر انتشار کانال**[1] را ملاحظه نمود. به عبارت دیگر مدت زمان لازم جهت انتشار سیگنال داده از یک گره به میزبان دیگر. این زمان تأخیر انتشار نقش حیاتی را در کارایی کانال بازی می‌کند. هرچه اندازه‌ی تأخیر انتشار طولانی‌تر باشد احتمال آن که گره‌ای که عملیات حس‌حامل را انجام می‌دهد قادر به تشخیص انتقالی باشد که با که گره‌ی دیگری همزمان در شبکه آغاز نموده است ضعیف‌تر می‌شود.



شکل ۵-۱۳ ♦ نمودار زمان و فضا از دو گره‌ی CSMA همراه با برخورد

1- Channel propagation delay

در شکل ۵-۱۳ که مربوط به پروتکل CSMA می‌باشد گره‌ها عملیات آشکارسازی برخورد را انجام نمی‌دهند. هر دو گره‌ی B و D به انتقال کامل قاب‌های خود ادامه می‌دهند حتی اگر برخوردهایی بین آن‌ها روی دهد. هنگامی که پروتکل در قاب CSMA/CD عمل نماید هرگاه گره‌ای عملیات آشکارسازی برخورد را انجام دهد بی‌درنگ انتقال را متوقف می‌سازد. شکل ۵-۱۴ عملیاتی مشابه شکل ۵-۱۳ را نشان می‌دهد با این تفاوت که هر دو گره‌ی عامل برخورد هریک به تنهایی انتقال را برای لحظه‌ی کوتاهی پس از تشخیص برخورد متوقف می‌سازند. آشکار است افزودن مکانیزم شناسایی برخورد به یک پروتکل دستیابی چندگانه سبب افزایش کارایی آن می‌شود زیرا از انتشار یک قاب بی‌ثمر و آلوده جلوگیری می‌شود. پروتکل شبکه‌های اترنت که به همین نام می‌باشد و در بخش‌های بعدی همین فصل تحلیل می‌شود از نوع CSMA/CD می‌باشد.



شکل ۵-۱۴ ♦ عملکرد نمودار زمان و فضا در پروتکل CSMA/CD

۴- پروتکل‌های از نوع نوبت‌بندی (TTP)

یادآوری می‌شود دو ویژگی مطلوب پروتکل دستیابی چندگانه عبارت است از:

۱- هنگامی که فقط یک گره فعال است، آن گره‌ی فعال نهفتگی Rbps را اختیار می‌کند و

۲- هنگامی که تعداد M گره فعال می‌باشند آن‌گاه هر گره‌ی فعال دارای نهفتگی نزدیک به $\frac{R}{M}$bps را اختیار می‌کند.

پروتکل‌های ALOHA و CSMA هر دو ویژگی نخست را شامل می‌شوند اما از امکان دوم بی‌بهره

می‌باشند. این باعث شده است پژوهشگران به دنبال خلق گروه دیگری از پروتکل‌ها باشند که آن‌ها را اصطلاحاً از نوع **رعایت نوبت (TTP)**[1] نام‌گذاری نموده‌اند. تعداد قابل توجهی از این پروتکل‌ها طراحی و پیشنهاد شده‌اند. در این بخش فقط دو گونه‌ی بسیار مهم آن‌ها تحلیل می‌شوند. نخستین آن **پروتکل نظرسنجی (PP)**[2] می‌باشد. این پروتکل بر این مبنا عمل می‌نماید که یکی از گره‌ها را به عنوان جلودار گزینش می‌کند. وظیفه‌ی گره‌ی طلایه‌دار آن است که از هریک از گره‌ها برمبنای نوبت و الویت آن‌ها نظرسنجی به عمل آورد. به این مفهوم که ابتدا پیامی را به گره‌ی شماره 1 ارسال می‌کند و به آن گره اعلام می‌کند نوبت انتقال فرا رسیده و می‌تواند با بهره‌گیری از این فرصت حداکثر تعداد معین قاب را گسیل نماید آیا از این امکان بهره‌گیری می‌کند؟ چنان‌چه گره‌ی شماره 1 قاب‌های داده‌ی خود را گسیل نماید آن‌گاه گره‌ی طلایه‌دار به شماره‌ی 2 اطلاع می‌دهد که می‌تواند تا حداکثر تعیین شده نسبت به انتقال قاب‌های خود در صورت تمایل اقدام نماید. بدیهی است گره‌ی جلودار تعیین می‌کند چه هنگام یک گره ارسال قاب‌های خود را تکمیل نموده است. این مهم از طریق عدم دریافت سیگنال برروی کانال تشخیص داده می‌شود. این عملیات از سوی گره‌ی جلودار دایماً در قالب یک حلقه‌ی چرخشی بارها تکرار می‌شود.

پروتکل نظرسنجی وقوع برخورد را حذف می‌کند و شیارهایی که سبب آلودگی پروتکل‌های دستیابی تصادفی می‌شوند را تهی می کند. این سبب می‌شود روش نظرسنجی از کارایی به مراتب بالاتری بهره‌مند شود. در عین حال این پروتکل نقطه ضعف‌هایی نیز دارد. نخست آن که پروتکل یک زمان تأخیر افزونی را که اصطلاحاً به آن **تأخیر نظرسنجی** گفته می‌شود به عملکرد شبکه تحمل می‌کند. این زمان عبارت است از زمان لازم جهت آگهی به یک گره که آن می‌تواند انتقال را آغاز نماید. به عنوان مثال اگر فقط یک گره فعال باشد آن‌گاه آن گره با سرعت کم‌تر از Rbps انتقال خود را انجام می‌دهد زیرا گره‌ی طلایه‌دار بایستی به نوبت در زمان معینی از گره‌های غیرفعال آمارگیری نماید. این نظرسنجی هر مرتبه که گره‌ی فعال حداکثر تعداد قاب‌های مجاز خود را گسیل می‌نماید انجام می‌شود. ضعف دوم این پروتکل که به مراتب از اهمیت بیش‌تری برخوردار است هنگامی است که گره‌ی طلایه‌دار دچار عیب شود آن‌گاه تمام کانال و در حقیقت شبکه از کار باز می‌ایستد. دو پروتکل **802.15** و **بلوتوث**[3] از جمله‌ی پروتکل‌های آمارگیری می‌باشند که در فصل ۶ تحلیل شده‌اند.

پروتکل نوبت‌بندی دوم اصطلاحاً معروف به **پروتکل انتقال روانه (TTP)**[4] می‌باشد. در این پروتکل گره‌ی طلایه‌دار وجود ندارد. از یک قاب داده‌ی ویژه و هدفمند به نام **پروانه**[5] استفاده می‌شود که بین گره‌ها با رعایت ترتیب ویژه‌ای مبادله می‌شود. به عنوان مثال گره‌ی 1 ممکن است همواره پروانه را به گره‌ی 2 و آن به گره‌ی 3 و سرانجام تا N انتقال دهد و مجدداً گره‌ی N آن را به 1 بازگشت می‌دهد. هنگامی که یک گره پروانه‌ای را دریافت نمود آن را نزد خود در صورت تمایل به ارسال قاب نگهداری می‌کند. در غیر این صورت بی‌درنگ پروانه را به گره‌ی بعدی منتقل می‌کند. چنان‌چه گره قاب‌هایی را جهت گسیل در اختیار داشته باشد هنگام دریافت پروانه با بهره‌گیری از آن حداکثر تعداد قاب‌های مجاز را گسیل می‌کند و سپس پروانه را به گره‌ی بعدی تحویل می‌دهد. روش انتقال پروانه

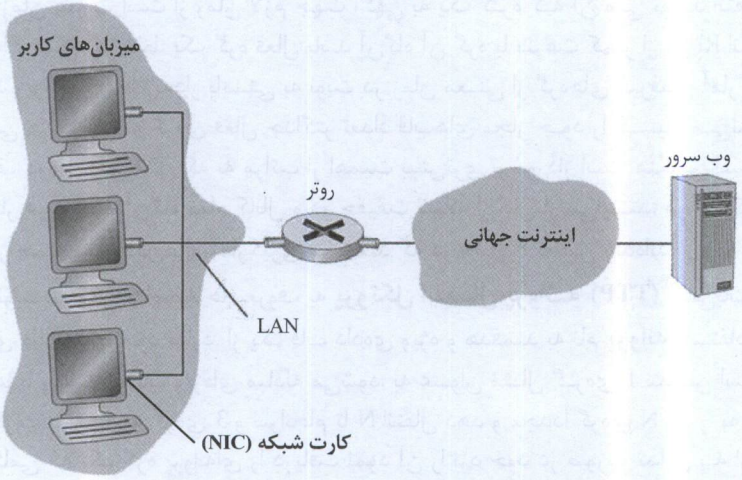
1- Taking turns protocols
2- Polling protocol
3- Bluetooth
4- Token passing protocol
5- Token

حالت غیر متمرکز دارد و بسیار کارآمد می‌باشد اما در عین حال ضعف‌هایی را نیز شامل می‌شود. به عنوان مثال وقوع عیب در یک گره می‌تواند سبب فروپاشی کامل کانال شود و یا اگر یک گره به‌گونه‌ی غیر عمد آزادسازی پروانه را انجام ندهد مراحل ویژه‌ی بازیابی پروانه بایستی به مرحله‌ی اجرا درآید تا پروانه مجدداً وارد حرکت حلقه گونه‌ی خود گردد. در طول سالیان تعداد قابل توجهی از این گونه پروتکل‌ها طراحی و پیشنهاد شده‌اند از میان گروه آن‌ها دو پروتکل مهم **FDDI** و **IEEE802.5** در ادامه تحلیل می‌شوند.

۵- شبکه‌های محلی (LANs)

از پروتکل‌های دستیابی چندگانه در رابطه با بسیاری از انواع مختلف کانال‌های فراپخش استفاده می‌شود. کاربرد آن‌ها در کانال‌های ماهواره‌ای و بدون کابل که گره‌ها از یک طیف فرکانسی مشترک جهت انتقال خود بهره می‌گیرند بسیار چشمگیر می‌باشد. در حال حاضر این کانال‌ها در فرآیند انتقال دستیابی کابلی در اینترنت و شبکه‌های محلی (LAN) به شکل گسترده‌ای بهره‌برداری می‌شوند.

به طور کلی در شبکه‌های LAN دستیابی همواره از یک میزبان به LAN و سپس به مسیریاب و از طریق آن به اینترنت می‌باشد. شکل ۵-۱۵ چگونگی بهره‌گیری از شبکه‌های محلی را نشان می‌دهد. اغلب شبکه‌های LAN دارای سرعت انتقال Rbps بسیار بالا می‌باشند. حتی در اوایل دهه‌ی 1980 میلادی شبکه‌های LAN با سرعت 10Mbps متداول بودند در حال حاضر این شبکه‌ها سرعت‌های 100Mbps، 1 Gbps و 10 Gbps را نیز شامل می‌شوند.



شکل ۵-۱۵ ♦ دستیابی میزبان به سرور وب در اینترنت از طریق کانال فراپخش بین میزبان و مسیریاب

در سراسر دهه‌ی 1980 و 1990 میلادی دو گروه از فنآوری‌های LAN بسیار متداول بودند. نخستین گروه شامل اترنت LAN یا به عبارت دیگر شبکه‌های IEEE802.3 می‌شدند که برمبنای دستیابی تصادفی پیاده‌سازی می‌شوند و گروه دوم فنآوری‌های LAN در شبکه‌های حلقوی با بهره‌گیری از پروانه می‌شوند که آن‌ها نیز به شبکه‌های IEEE802.5 معروف می‌باشند و سرانجام شبکه‌های با پروتکل **واسط نوری توزیع داده**[1] **(FDDI)** می‌باشند. در بخش‌های بعدی همین فصل شبکه‌های

1- Fibre digital distribution interface

اترنت تحلیل بیش‌تری خواهند شد اما در ادامه شبکه‌های حلقوی پروانه‌دار یا **توکن رینگ**[1] بررسی می‌شوند. بدیهی است شبکه‌های حلقوی به دلیل گسترش روز افزون پروتکل‌های اترنت در حال انقراض کامل می‌باشند. معمولاً در شبکه‌های محلی توکن رینگ همگی N گره‌ی میزبان و مسیریاب‌های شبکه‌ی LAN از طریق پیوندهای مستقیم در قالب حلقه به یکدیگر متصل شده‌اند. فناوری توکن رینگ چگونگی ترتیب عبور پروانه را تعریف می‌کند. هنگامی که یک گره پروانه را در دست گرفت قاب داده‌ی خود را به کمک پروانه در محیط حلقه انتشار می‌دهد بنابراین یک کانال فراپخش مجازی را خلق می‌کند. گره مقصد قاب را از رسانه‌ی لایه‌ی پیوند هنگام انتشار آن می‌خواند. گره‌ی گسیل کننده‌ی فریم مسئولیت خارج نمودن قاب از حلقه را برعهده دارد. پروتکل FDDI ویژه‌ی شبکه‌هایی که از لحاظ جغرافیایی وسیع می‌باشند طراحی شده است. بنابراین نام دیگر آن‌ها **شبکه‌های کلان‌شهر**[2] یا MAN می‌باشد. در چنین شبکه‌های گسترده انتشار مجدد قاب به گره‌ی ارسال کننده هنگامی که قاب به گره‌ی مقصد رسید ناکارآمد خواهد بود بنابراین در پروتکل FDDI گره‌ی مقصد قاب را هنگام دریافت حذف می‌کند. به عبارت دیگر FDDI یک پروتکل کاملاً فراپخش نمی‌باشد زیرا همه‌ی گره‌ها قاب ارسالی را دریافت نمی‌کنند.

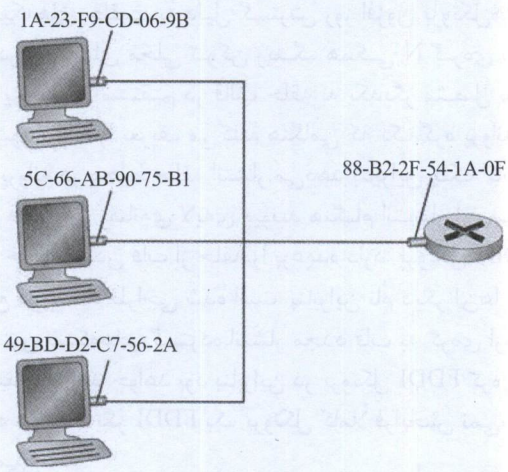
۵-۵ آدرس‌دهی لایه‌ی پیوند

گره‌هایی که شامل میزبان‌ها و مسیریاب‌ها می‌شوند هریک دارای آدرس‌های لایه‌ی پیوند می‌باشند. یادآوری می شود گره‌ها آدرس‌های لایه‌ی شبکه را نیز اختیار می‌نمودند. حال این پرسش مطرح می‌شود چرا نیاز به دو آدرس، یکی در لایه‌ی شبکه (آدرس منطقی) و دیگری در لایه‌ی پیوند (آدرس فیزیکی) می‌باشد؟ در این بخش تلاش شده است ضمن پاسخ‌گویی به این پرسش قواعد و عملکرد آدرس‌های لایه‌ی پیوند تشریح گردد و علل سودمندی بهره‌گیری از دو نوع آدرس نیز توضیح داده شود و سرانجام **پروتکل بازیابی آدرس**[3] **(ARP)** که از طریق آن می‌توان آدرس‌های لایه‌ی پیوند را از نوع لایه‌ی شبکه‌ی آن‌ها بازیابی نمود، تحلیل می‌شود.

• آدرس‌های کنترل دستیابی رسانه (MAC)

در حقیقت کارت واسط شبکه (NIC) است که دارای آدرس لایه‌ی پیوند می‌باشد. شکل ۵-۱۶ این حقیقت را به تصویر کشیده است. آدرس لایه‌ی پیوند در اغلب اوقات به‌عنوان آدرس LAN، آدرس فیزیکی و یا آدرس **MAC** نامیده می‌شود. از آن‌جائیکه MAC متداول‌ترین نام استفاده شده می‌باشد بنابراین در ادامه‌ی مطالب این بخش صرفاً از این عنوان استفاده می‌شود. برای اغلب شبکه‌های LAN فارغ از نوع پروتکل آن‌ها که اترنت، WiFi و غیره می‌باشند، طول آدرس MAC معادل 6 بایت یا 48 بیت می‌باشد. همان‌گونه که در شکل ۵-۱۶ نشان داده شده است این آدرس‌ها در قالب کدهای هگزا بیان می‌شوند که هر بایت از آدرس یک جفت عدد هگزا را تعریف می‌کند. اگرچه آدرس‌های MAC به‌گونه‌ای طراحی شده‌اند که دایمی باشند اما در حل حاضر این امکان نیز وجود دارد که بتوان از طریق نرم‌افزاری آدرس MAC یک کارت شبکه را نیز تغییر داد. در ادامه‌ی بررسی مطالب، فرض بر آن است که آدرس‌های MAC ثابت و غیر قابل تغییر می‌باشند.

1- Token ring
2- Metropolitan area networks
3- Address resolution protocols



شکل ۵-۱۶ ♦ هر کارت شبکه‌ی متصل به یک LAN دارای یک آدرس انحصاری MAC می‌باشد.

یکی از مهم‌ترین ویژگی‌های آدرس‌های MAC آن است که هیچ یک از کارت‌های شبکه نمی‌توانند شامل آدرس‌های مشابه باشند. اگرچه تعداد انبوهی از کارت‌های شبکه در کشورهای مختلف تولید می‌شود اما دو کارت شبکه به هیچ روی دارای آدرس MAC مشابه نخواهند شد. چرا؟ پاسخ آن است که مدیریت آدرس‌های MAC برعهده‌ی IEEE می‌باشد. قبلاً عنوان گردید طول آدرس MAC برابر 6 بایت است این به مفهوم آن است که تعداد 2^{48} ترکیب آدرس MAC گوناگون می‌توان تعریف نمود از میان این تعداد ترکیب آدرس، IEEE تعداد 2^{24} ترکیب از آدرس را که شامل 3 بایت (24 بیت) از 6 بایت نخست آدرس MAC می باشد معمولاً به شرکت‌های تولید کننده‌ی کارت شبکه به‌صورت ثابت اختصاص می‌دهد و آن را به تولید کننده می‌فروشد. هریک از این ترکیب‌های 2^{24} گانه اصطلاحاً شناسه‌ی دایمی و انحصاری یک شرکت تولید کننده‌ی کارت شبکه به شمار می‌آید. 2^{24} ترکیب دیگر که شامل 24 بیت بخش پائین آدرس MAC می‌باشد به عنوان شماره سریال کارت شبکه در اختیار کامل شرکت‌ها جهت بهره‌برداری می‌باشد. با اندکی تأمل ملاحظه خواهد شد به هیچ روی احتمال مشابه بودن یک آدرس MAC برای دو کارت شبکه وجود نخواهد داشت. برخلاف آدرس‌های IP که از یک مکان به موقعیت دیگری تغییر می‌یافتند و هیچ‌گاه حالت ثابت و دایمی نداشتند، آدرس‌های MAC در همه‌ی نقاط ثابت و غیرقابل تغییر از نقطه نظر سخت‌افزاری و مکانی می‌باشند. آدرس MAC را می‌توان شبیه آدرس ثبتی یک ملک دانست در حالیکه آدرس IP مانند آدرس پستی از لحاظ نام خیابان و یا سایر اطلاعات پستی که ممکن است بنابرعلل و شرایط به وجود آمده تغییر نمایند. بنابراین مانند یک ملک که دارا بودن هر دو آدرس ثبتی و پستی می‌تواند مفید باشد. آدرس‌های IP و MAC که به ترتیب متعلق به لایه‌های شبکه و پیوند داده‌ها می‌باشند نیز سودمند است.

در آغاز این فصل توضیح داده شد هنگامی که یک کارت شبکه (گره) مایل به ارسال یک قاب به گره‌ی دیگری است. کارت شبکه‌ی ارسال کننده، آدرس MAC گره‌ی گیرنده را به قاب داده اضافه می‌نماید و سپس آن را به درون شبکه‌ی LAN گسیل می‌کند. چنان‌چه LAN یک شبکه‌ی فراپخش مانند 802.11 و یا بسیاری از شبکه‌های اترنت LAN دیگر باشد قاب داده از سوی گره‌های دیگر دریافت و پردازش می‌شود. به‌ویژه آن که هر کارت شبکه هنگام دریافت قاب آن را از لحاظ آدرس

MAC مقصد ارزیابی می‌کند تا از نگاشت این آدرس با MAC خویش مطمئن گردد. در صورت نگاشت آدرس، آن کارت شبکه از طریق لایه‌ی پیوند داده گرام درون قاب را استخراج و آن را به لایه‌های بالاتر پشته‌ی پروتکلی تحویل می‌دهد، در غیر این صورت از قاب صرف‌نظر می‌کند. بنابراین فقط مقصد به هنگام دریافت قاب داده فعال می‌شود.

بعضی مواقع ضرورت دارد که قاب داده‌ی ارسالی از سوی یک کارت شبکه به‌وسیله‌ی بقیه کارت‌ها در LAN پردازش شوند. در این حالت کارت شبکه‌ی فرستنده آدرس MAC ویژه‌ای را در دامنه‌ی آدرس مقصد قاب تزریق می نماید که اصطلاحاً به آن **آدرس فراپخش MAC** گفته می‌شود. برای شبکه‌های LAN که از 6 بایت آدرس MAC استفاده می‌کنند آدرس فراپخش شامل رشته‌ای از 48 بیت با ارزش 1 می‌باشد. به عبارت دیگر ارزش هگزا در آن معادل: FF-FF-FF-FF-FF-FF می‌باشد.

چند علت مختلف را می‌توان جهت نیاز به دو نوع آدرس لایه‌ی شبکه و پیوند برشمرد. نخستین دلیل آن است که شبکه‌های LAN به‌گونه‌ای طراحی شده‌اند که قادر به بهره‌گیری از هر پروتکل لایه‌ی پیوند باشند به عبارت دیگر فقط محدود به پروتکل‌های IP و اینترنت نمی‌شوند. اگر به کارت‌های شبکه فقط آدرس‌های IP به جای آدرس‌های بی‌طرف (خنثی) MAC اختصاص می‌یافت، کارت‌های شبکه به‌سادگی قادر به پشتیبانی پروتکل‌های دیگر لایه‌ی شبکه از قبیل IPX، DECnet و غیره نمی‌شدند. دلیل دوم آن است که اگر کارت‌های شبکه فقط از آدرس‌های لایه‌ی شبکه بهره می‌بردند این آدرس‌ها بایستی در حافظه‌ی RAM در کارت شبکه ذخیره می‌شدند و هر مرتبه کارت شبکه جابه‌جا و یا خاموش و روشن می‌شد مجدداً آن‌ها نیاز به پیکربندی داشتند. بدیهی است می‌توان این گونه پیشنهاد نمود که کارت‌های شبکه از هیچ نوع آدرسی استفاده نکنند و هر کارت صرفاً با دریافت هر قاب داده آن را به لایه‌های بالاتر انتقال دهد. دشواری این راه کار آن است که میزبان با گسیل هر قاب در شبکه دچار وقفه می‌شود که شامل قاب‌های مقصد کارت‌های دیگر بر روی همان کانال فراپخش LAN نیز می‌شود. درنتیجه برای آن که استقلال لایه‌ها در یک معماری شبکه کاملاً حفظ شود، لایه‌های گوناگون نیازمند روش‌های آدرس‌دهی ویژه‌ی خود می‌باشند. بنابراین تاکنون 3 نوع از آدرس‌ها مطالعه گردید. این سه عبارتند از:

- نام میزبان‌ها برای لایه‌ی کاربرد
- آدرس‌های IP برای لایه‌ی شبکه
- آدرس‌های MAC برای لایه‌ی پیوند داده.

پروتکل بازیابی آدرس (ARP)

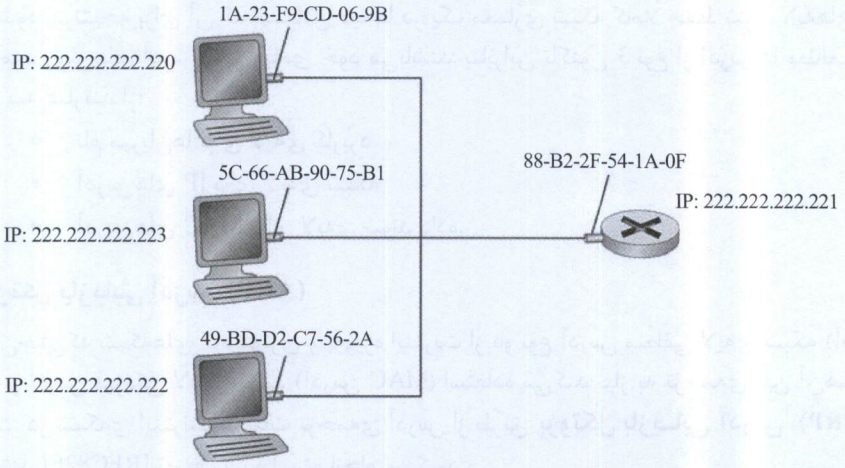
از آن‌جائی‌که شبکه‌های کامپیوتری و به‌ویژه اینترنت از دو نوع آدرس منطقی لایه‌ی شبکه (آدرس‌های IP) و آدرس فیزیکی لایه‌ی پیوند (آدرس MAC) استفاده می‌کند نیاز به ترجمه‌ی بین آن‌ها ضروری است. در شبکه‌ی اینترنت عملیات ترجمه‌ی آدرس از طریق **پروتکل بازیابی آدرس**[1] **(ARP)** که در مستند [RFC826] تعریف شده است، انجام می‌گیرد.

برای درک اهمیت نیاز به پروتکلی مانند ARP، شبکه‌ی تصویر شده در شکل ۵-۱۷ را در نظر بگیرید. در این مثال ساده هر گره دارای یک آدرس IP و یک آدرس MAC می‌باشد. اگر فرض شود گره‌ی با آدرس IP برابر 222.222.222.220 مایل به ارسال یک داده گرام IP به گره‌ی با آدرس IP:

1- Address resolution protocol

222.222.222.222 باشد. در این مثال بنابر مطالب تشریح شده در فصل ۴ هر دو گره‌ی مبدأ و مقصد متعلق به یک شبکه‌ی LAN می‌باشند. برای گسیل یک داده گرام گره‌ی مبدأ مجبور به تحویل داده گرام به‌همراه آدرس MAC گره‌ی مقصد است که آدرس IP آن 222.222.222.222 می‌باشد. بنابراین کارت شبکه‌ی گره‌ی فرستنده یک قاب داده‌ی لایه‌ی پیوند را می‌سازد که محتویات آن قاب، داده گرام IP، و از همه مهم‌تر آدرس MAC گره‌ی مقصد می‌باشد. هنگامی که قاب تشکیل گردید به درون شبکه‌ی LAN گسیل می‌شود.

پرسش مهم چگونگی بازیابی آدرس MAC از سوی فرستنده به کمک آدرس IP: 222.222.222.222 می‌باشد؟ ممکن است درست حدس زده باشید از طریق پروتکل ARP. یک قطعه‌ی نرم‌افزاری ARP در گره‌ی فرستنده هر آدرس IP شبکه‌ی LAN مشابه خود را به عنوان ورودی به‌شمار می‌آورد و آدرس MAC مربوط به آن را بازگشت می‌دهد. در مثال شکل ۵-۱۷ گره‌ی فرستنده (222.222.222.220) برای قطعه‌ی نرم‌افزاری ARP خود آدرس IP مقصد 222.222.222.222 را فراهم می‌کند. در نتیجه‌ی این عمل پروتکل ARP درحال اجرا برروی آن کارت شبکه آدرس MAC به‌صورت: 49-BD-D2-C7-56-2A را بازگشت می‌دهد. بنابراین ملاحظه می شود که ARP بازیابی یک آدرس IP به MAC را انجام می‌دهد. از نگاه دیگری این بازیابی بسیار شبیه به پروتکل DNS می‌باشد که در فصل ۲ تحلیل گردید و بازیابی آدرس IP را از طریق نام انجام می‌داد. بدیهی است تفاوت عمده‌ی بین این دو روش بازیابی آدرس در آن است که DNS بازیابی نام را برای میزبان‌ها در هر نقطه‌ی اینترنت انجام می‌دهد در حالی‌که ARP بازیابی آدرس‌های IP را فقط برای گره‌های یک زیرشبکه‌ی مشابه انجام می‌دهد. اگر گره‌ای در تهران تلاش در بازیابی آدرس IP برای گره‌ای در اصفهان را داشته باشد نتیجه‌ای که پردازش ARP بازگشت می‌دهد یک پیام خطا می‌باشد.



شکل ۵-۱۷♦ هر گره برروی یک LAN شامل یک آدرس IP و یک آدرس MAC می‌باشد.

اکنون که چگونگی تأثیرگذاری ARP را بررسی نمودیم وقت آن رسیده است که کارکرد آن را نیز تحلیل نمائیم. هر گره (میزبان و یا مسیریاب) دارای یک جدول ARP در حافظه‌اش می‌باشد. این جدول شامل نگاشتی از آدرس‌های IP به MAC می باشد. شکل ۵-۱۸ شبیه‌سازی یک جدول ARP شامل یک ارزش TTL نیز می‌باشد که به تعریف زمان حذف هر آدرس نگاشت شده از جدول

می‌پردازد. ملاحظه می‌شود که جدول لزوماً شامل یک ورودی برای هر گره‌ای در زیر شبکه نمی‌باشد زیرا برخی از گره‌ها ممکن است ورودی‌هایی داشته‌اند که در حال حاضر از اعتبار خارج شده‌اند. برخی از گره‌ها نیز ممکن است هرگز شامل ورودی در جدول ARP نشده باشند. زمان خاتمه‌ی اعتبار متداول برای هر ورودی از لحظه نخستین قرارگیری آن در جدول ARP تقریباً 20 دقیقه می‌باشد.

اگر فرض شود گره‌ی 222.222.222.220 مایل به ارسال یک داده گرام است که مقصد آن گره‌ی دیگری با آدرس IP تعریف شده‌ای بر روی همان زیرشبکه می‌باشد گره‌ی گسیل کننده نیاز به اخذ آدرس MAC گره‌ی مقصد با توجه به آدرس IP آن دارد. انجام این وظیفه بسیار ساده می‌شود اگر جدول ARP گره‌ی گسیل کننده شامل یک ورودی برای گره مقصد باشد، اما اگر جدول ARP شامل ورودی موردنظر برای گره مقصد نباشد، چه خواهد شد؟ اگر فرض شود گره‌ی 222.222.222.220 مایل به ارسال یک داده گرام به گره‌ی 222.222.222.222 باشد در این حالت گره‌ی گسیل کننده از پروتکل ARP جهت بازیابی آدرس بهره می‌گیرد. ابتدا گره‌ی ارسال کننده بسته‌ی ویژه‌ای را تشکیل می‌دهد که اصطلاحاً به آن **بسته‌ی ARP** گفته می‌شود. یک بسته‌ی ARP شامل چندین دامنه است که شامل آدرس‌های IP و MAC فرستنده و گیرنده می‌شود. هر دو بسته‌ی درخواست و پاسخ ARP دارای قالب مشابهی می‌باشند. هدف از بسته‌ی متقاضی ARP، پرسش از همه‌ی گره‌های دیگر زیرشبکه برای شناسایی آدرس MAC مربوط به آدرس IP است که هدف بازیابی آدرس است.

با توجه به مثال مورد نظر، گره‌ی 222.222.222.220 یک بسته‌ی تقاضای ARP را به کارت شبکه‌ی خود به همراه اطلاعاتی که از طریق آن کارت شبکه، بسته‌ی ARP را به آدرس فراپخش MAC یعنی FF-FF-FF-FF-FF-FF تحویل نماید، گسیل می‌کند. کارت شبکه آن‌گاه بسته‌ی ARP را در قالب قاب داده‌ی لایه‌ی پیوند بسته‌بندی می‌کند و با استفاده از آدرس فراپخش MAC را به‌عنوان آدرس مقصد قاب داده وارد زیرشبکه می‌کند. بسته‌ی تقاضای ARP را می‌توان معادل فردی دانست که در سالن انتظار اداره‌ی ثبت اسناد فریاد می‌کند شماره آدرس ثبتی پلاک با آدرس پستی خیابان x محله‌ی y و پلاک 3 را چه کسی در اختیار دارد؟ قاب داده در قالب این مثال را همه‌ی افراد حاضر در سالن انتظار می‌شنوند اما یک نفر از میان جمعیت برپا می‌ایستد و فریاد برمی‌آورد من در اختیار دارم و این هم شماره‌ی پلاک ثبتی موردنظر. در این ارتباط قاب حامل تقاضای ARP از سوی همه‌ی کارت‌های شبکه دریافت می‌شود و هریک از آن‌ها به نگاشت آدرس IP خود با آدرس IP پیام تقاضای ARP مقصد می‌پردازند. سرانجام یکی از گره‌ها پاسخ گره‌ی درخواست کننده را از طریق یک بسته‌ی پاسخ ARP که در آن آدرس MAC قرار دارد فراهم می‌آورد. گره‌ی متقاضی با آدرس IP برابر 222.222.222.220 جدول ARP خود را به روزرسانی می‌کند و سپس داده گرام اصلی را پس از بسته‌بندی در قالب قاب داده‌ی لایه‌ی پیوند به درون زیرشبکه گسیل می‌کند.

آدرس IP	آدرس MAC	زمان اعتبارسنجی (TTL)
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

شکل ۵-۱۸♦ نمونه‌ای از یک بخش جدول ARP در گره‌ی 222.222.222.220

موارد جالبی پیرامون پروتکل ARP وجود دارد که ملاحظه‌ی آن‌ها با اهمیت می‌باشد. نخست آن که پیام درخواست ARP در قالب یک قاب داده‌ی فراپخش گسیل می‌شود در حالی‌که پاسخ ARP به

تقاضا در درون یک قاب داده‌ی استاندارد و معمولی صورت می‌پذیرد. آیا می‌توانید علت را توجیح نمائید؟ دوم آن‌که جدول ARP هر گره به صورت خودکار تشکیل می شود به عبارت دیگر نیازی نیست که مدیر شبکه آن را پیکربندی نماید. سرانجام آن‌که اگر یک گره از زیرشبکه قطع شود اطلاعات درون آن نهایتاً از جداول ARP گره‌های دیگر حذف می‌شود. در اغلب موارد دانشجویان دچار این سردرگمی می‌شوند که آیا ARP یک پروتکل لایه‌ی پیوند و یا آن یک پروتکل لایه‌ی شبکه است. در مطالب این زیربخش مشاهده گردید که یک بسته‌ی ARP در درون یک قاب داده‌ی لایه‌ی پیوند بسته‌بندی می‌شود و بنابراین از لحاظ ساختاری در بالای لایه‌ی پیوند قرار دارد. اما یک بسته‌ی ARP شامل دامنه‌هایی است که آدرس‌های لایه‌ی پیوند در آن‌ها قرار می‌گیرد و از این نگاه یک پروتکل لایه‌ی پیوند به‌شمار می‌آید. از طرف دیگر آدرس‌های لایه‌ی شبکه نیز در آن قرار گرفته‌اند و بنابراین می‌تواند یک پروتکل لایه‌ی شبکه نیز به شمار آید! شاید بهترین توصیف برای ARP آن باشد که این پروتکل را در مرز مابین دو لایه‌ی شبکه و پیوند قرار دهیم اگرچه با مطالب مربوط به لایه‌بندی شبکه در فصل اول در تضاد است. اما از طرفی این همان موضوع پیچیدگی پروتکل‌ها در دنیای واقعی است که قبلاً در مطالب آغازین این اثر به آن اشاره شد.

- **چگونگی بازیابی آدرس MAC بین گره‌های دو شبکه‌ی متفاوت**

در مطالب مورد اشاره در زیربخش قبلی توضیح داده شد چگونه ARP عملیات بازیابی آدرس MAC را مابین گره‌های درون یک شبکه‌ی کامپیوتری انجام می‌دهد. اکنون موضوع پیچیده‌تر بازیابی آدرس MAC جهت ارسال داده گرام از گره‌ای در یک زیرشبکه به گره‌ی مقصد در زیرشبکه‌ای دیگر بررسی می‌شود. شکل ۵-۱۹ چارچوب این عملکرد را در یک شبکه‌ی ساده شامل دو زیرشبکه که از طریق یک روتر به یکدیگر متصل شده‌اند نشان می‌دهد.

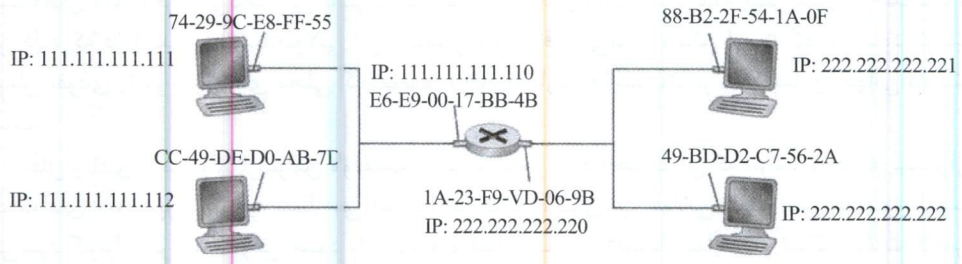
چندین مورد در شکل ۵-۱۹ وجود دارد که بایستی با تأمل آن‌ها را مورد مطالعه قرار داد. نخست دو نوع گره در این شبکه وجود دارد: میزبان‌ها و مسیریاب‌ها.

هر میزبان دارای یک آدرس IP و یک کارت شبکه است. اما همان‌گونه که در فصل ۴ توضیح داده شد یک مسیریاب شامل یک آدرس IP برای هریک از ده‌ها کارت شبکه‌ی خود می‌باشد. بنابراین هریک از واسط‌ها و کارت‌های شبکه در روتر شامل یک مدول ARP ویژه‌ی خویش نیز می‌باشند. از آن‌جائی‌که روتر شکل ۵-۱۹ شامل 2 واسط است بنابراین دو آدرس IP، دو مدول ARP و دو کارت شبکه نیز نیاز می‌باشند. بدیهی است هر کارت در شبکه نیز آدرس MAC انحصاری خود را دارد.

علاوه بر موارد بالا ملاحظه‌ی این نکته نیز اهمیت دارد که زیرشبکه‌ی 1 دارای آدرس شبکه به صورت 111.111.111/24 و آدرس زیرشبکه‌ی 2 نیز به‌گونه‌ی 222.222.222/24 می‌باشد. بنابراین همه‌ی واسط‌های متصل به زیرشبکه‌ی 1 آدرس‌های IP از نوع 111.111.111.XXX و کلیه‌ی واسط‌های زیرشبکه‌ی 2 آدرس‌های IP در قالب 222.222.222.yyy را اختیار می‌کنند. در این مقطع به تحلیل چگونگی ارسال یک داده گرام از زیرشبکه‌ی 1 به مانند آن زیرشبکه 2 می‌پردازیم.

اگر فرض شود میزبان با آدرس IP:111.111.111.111 مایل به ارسال یک داده‌گرام IP به یک میزبان در زیرشبکه 2 با آدرس IP برابر: 222.222.222.222 باشد. میزبان فرستنده، داده گرام را به کارت شبکه‌ی خود مطابق معمول گسیل می‌کند. اما میزبان فرستنده بایستی با فراهم نمودن یک آدرس MAC مناسب کارت شبکه‌ی خود را از آن مقصد واقعی آگاه نماید. چه آدرس MAC بایستی

کارت شبکه‌ی این میزبان استفاده نماید؟ ممکن است این تصور به وجود آید که آدرس MAC مناسب مربوط به کارت شبکه‌ی میزبان 222.222.222.222 یعنی 49-BD-D2-C7-56-2A می‌باشد. این حدس به هر حال اشتباه است. اگر کارت شبکه‌ی فرستنده از یک آدرس MAC استفاده نماید آن‌گاه هیچ یک از کارت‌های زیرشبکه‌ی 1، داده گرام IP را به لایه‌ی شبکه متعلق به خود انتقال نمی‌دهد. زیرا آدرس مقصد قاب داده با هیچ یک از آدرس‌های MAC در زیرشبکه‌ی 1 نگاشت نمی‌شود. داده گرام IP سرگردان در زیرشبکه باقی می‌ماند.



شکل ۵-۱۹ ♦ ارتباط بین دو زیرشبکه از طریق یک روتر

اگر با دقت به شکل ۵-۱۹ توجه کنیم ملاحظه می‌شود برای آن که یک داده گرام از آدرس IP: 111.111.111.111 به گره‌ای در زیرشبکه‌ی 2 منتقل گردد آن داده، گرام نخست بایستی به واسط مسیریاب با آدرس: 111.111.111.110 منتقل شود که آدرس IP نخستین مسیریاب دریچه در مسیر مقصد نهایی می‌باشد. بنابراین آدرس MAC مناسب برای قاب داده عبارت است. از آدرس کارت شبکه برای واسط روتر با آدرس IP: 111.111.111.110 که برابر: E6.E9.00.17.BB.4B می‌باشد. چگونه میزبان فرستنده آدرس MAC موردنظر برای آدرس IP: 111.111.111.110 را به‌دست می‌آورد؟ بدیهی است از طریق پروتکل ARP! هنگامی که کارت شبکه‌ی فرستنده این آدرس MAC را دریافت نمود به تشکیل یک قاب شامل داده گرام به آدرس IP مقصد: 222.222.222.222 می‌پردازد و آن قاب را به درون زیرشبکه‌ی 1 تزریق می‌کند. کارت شبکه‌ی روتر در زیرشبکه‌ی 1 ملاحظه می‌کند که قاب با آدرس لایه‌ی پیوند به سوی آن نشانه‌گذاری شده است، بنابراین قاب را به لایه‌ی شبکه‌ی مسیریاب هدایت می‌کند. نتیجه آن که داده گرام IP با موفقیت از میزبان مبدأ به مسیریاب منتقل می‌شود. اما هنوز کار به پایان نرسیده است زیرا داده گرام بایستی از روتر به مقصد نهایی آن تحویل گردد. در این حالت مسیریاب بایستی واسط صحیح را که داده گرام از طریق آن گسیل به جلو می‌شود به‌دست آورد. بنابر آن‌چه در فصل ۴ توضیح داده شده این عمل از طریق پردازش جدول گسیل به جلو در روتر انجام می‌شود. جدول گسیل کننده به مسیریاب اطلاع می‌دهد که داده گرام بایستی از طریق واسط روتر با آدرس 222.222.222.220 گسیل به جلو گردد. این واسط سپس داده گرام را به کارت شبکه‌ی خود گسیل می‌نماید و داده گرام دریافتی تبدیل به یک قاب داده‌ی جدید می‌شود و سپس به درون زیرشبکه‌ی 2 گسیل می‌شود. در این لحظه آدرس MAC مقصد قاب داده موردنیاز می‌باشد. مسیریاب با بهره‌گیری از ARP آدرس MAC مقصد را نهایتاً دستیابی می‌کند.

پروتکل ARP برای اترنت در مستند [RFC826] تعریف شده است. مستند [RFC1180] نیز ARP را برای مجموعه‌ی پروتکل‌ها TCP/IP تشریح نموده است.

۵-۶ پروتکل اترنت

اترنت از جمله پرنفوذترین پروتکل‌ها در بازار شبکه‌های LAN کابلی به‌شمار می‌رود. در سراسر دهه‌های 1980 و اول 1990 میلادی اترنت در رقابت شدید با دیگر فناوری‌ها LAN مانند پروتکل‌های ATM، FDDI و توکن رینگ قرار داشت. برخی از این فناوری‌های برای مدت زمان قابل توجهی بخشی از بازار شبکه را در تسخیر خود گرفتند. اما از هنگام ظهور اترنت در اواسط دهه 1970 میلادی تا امروز این پروتکل دستخوش تکامل و رشد فزآینده‌ای بوده است که هنوز هم آن را در شمار پرنفوذترین‌های شبکه‌های LAN قرار می‌دهد. این تأثیر نه تنها تاکنون برجای مانده است بلکه پیش‌بینی‌ها حکایت از آن دارد که در آینده‌ای نسبتاً طولانی این استمرار پایدار باقی می‌ماند. ممکن است گفته شود اترنت همان نفوذی را در شبکه‌های محلی کامپیوتری (LAN) دارد که اینترنت در شبکه‌سازی جهانی داشته است.

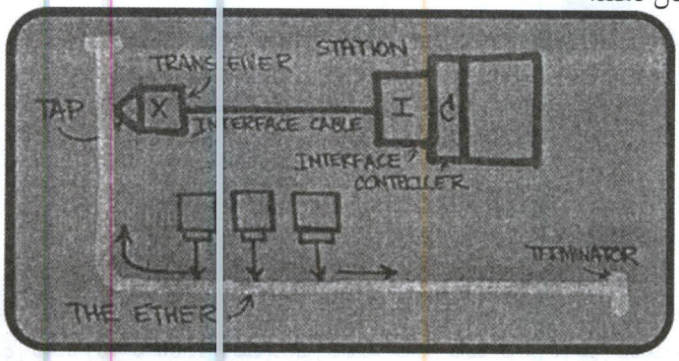
علل زیادی را می‌توان از عوامل موفقیت اترنت به شمار آورد. نخست آن که اترنت از جمله پیشتازان شبکه‌های LAN پرسرعت به شمار می‌آید زیرا قدمت بهره‌گیری آن طولانی است. این موضوع سبب می‌شود کاربران اترنت به نوعی خود را وابسته و معتاد به آن قلمداد نمایند و ترک عادت گاهی غیرممکن و یا آن که به گذر زمان نسبتاً طولانی نیازمند است. علت دوم پیچیدگی و پرهزینه بودن پروتکل‌هایی مانند ATM، FDDI و توکن رینگ می‌باشد. علت سوم را می‌توان از دیدگاه کاملاً متفاوتی تحلیل نمود و آن عبارت است از این واقعیت که اگرچه فناوری‌های دیگر LAN مانند ATM و یا FDDI معمولاً نرخ انتقال به مراتب بالاتری را عرضه می‌کنند اما انعطاف‌پذیری اترنت سبب شده است که این پروتکل همواره با چنین ضعف‌هایی مبارزه نموده و با معرفی نسخه‌های جدیدتر دستیابی با نرخ بالاتر سبب بازگشت مجدد اترنت به صحنه شده است. در اوایل دهه‌ی 1990 میلادی با معرفی و ظهور اترنت راهیابی شده و تأثیر آن بر افزایش سرعت انتقال بازهم این پروتکل را پیشتاز دیگران قرار داد. آخرین علت را نیز می‌توان در محبوبیت این پروتکل دانست. عناصر سخت‌افزاری اترنت مانند کارت‌های شبکه و راهیاب‌ها فوق‌العاده ارزان و قابل دسترس می‌باشند.

اترنت LAN اولیه در اواسط دهه‌ی 1970 میلادی در شرکت زیراکس از سوی دو شخصیت به نام‌ها مت‌کالفه و باگز طراحی و پیشنهاد گردید. شکل ۵-۲۰ نخستین تصویری را که مت کالفه برای این ابداع جدید رسم نمود به نمایش می‌گذارد. در تصویر ملاحظه می‌شود که اترنت LAN اولیه از گذرگاه کابل هم محور جهت ارتباط بین گره‌ها بهره می‌گرفت. فناوری گذرگاه (BUS) برای اترنت تا سراسر دهه‌های 1980 و اواسط 1990 میلادی نیز ادامه یافت. اترنت LAN با ساختار گذرگاه یک شبکه‌ی فراپخش را تعریف می‌کند که در آن همه‌ی قاب‌های داده از سوی همه‌ی کارت‌های شبکه متصل به گذرگاه دریافت و پردازش می‌شوند.

تا اواخر دهه‌ی 1990 اغلب شرکت‌ها و دانشگاه‌ها، شبکه‌ی LAN خود را با ساختار توپولوژی اترنت از نوع ستاره یا هاب جایگزین نمودند. شکل ۵-۲۱ این توپولوژی را نشان می‌دهد. در چنین ساختاری میزبان‌ها و مسیریاب‌ها مستقیماً از طریق جفت رشته کابل مسی (TP) به یک نقطه‌ی مرکزی به نام **هاب**[1] متصل می‌شوند. یک هاب وسیله‌ای از نوع لایه‌ی فیزیکی است که به‌گونه‌ی منفرد برروی هریک از بیت‌ها به جای قاب‌ها عمل می‌نماید. هنگامی که یک بیت با بیان 0 و یا 1 از یک واسط وارد می‌شود، هاب به شکل ساده‌ای بیت‌های دریافتی را مجدداً خلق، تقویت و به سوی همه‌ی واسط‌های

1- Hub

دیگر شبکه گسیل می‌کند. بنابراین اترنت با توپولوژی ستاره و بهره‌گیری از هاب نیز یک LAN فراپخش می‌باشد. هرگاه هاب یک بیت را از یکی از واسط‌های خود دریافت نماید نسخه‌ای از آن بیت را به همه‌ی واسط‌های دیگر خود منتقل می‌کند. چنان‌چه یک هاب قاب‌هایی را از دو واسط متفاوت همزمان دریافت نماید برخورد روی می‌دهد و بنابراین گره‌هایی که قاب‌ها را خلق نموده‌اند بایستی آن‌ها را مجدداً انتقال دهند.



شکل ۵-۲۰ ♦ طراحی اصلی اترنت شامل کابل واسط، کارت شبکه‌ی ا رنت و فرستنده، گیرنده خارجی اترنت.

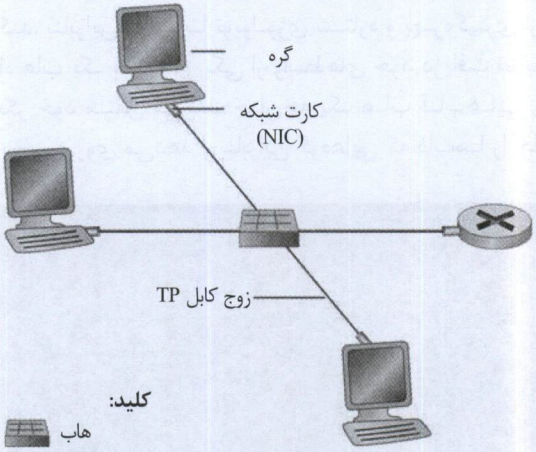
در اواسط دهه‌ی 2000 میلادی اترنت شاهد یک تغییر تکاملی مهم شد. شبکه‌های اترنت به بهره‌گیری از ساختار ستاره ادامه دادند اما **راهیاب**[1] جایگزین هاب گردید.

در بخش‌های بعدی اهمیت اترنت راهیابی شده با جزئیات بیش‌تری بررسی خواهد شد اما در این مقطع کافی است به این مختصر بسنده شود که سوئیچ یا همان راهیاب در واژگان شبکه نه تنها مانع از برخورد می‌شود بلکه یک راهیاب بسته به گونه‌ی ابتدا ذخیره و سپس گسیل می‌باشد. ویژگی که مسیریاب‌ها نیز از آن بهره می‌گیرند اما با این تفاوت که سوئیچ یا راهیاب تا لایه دوم برعکس روتر که وسیله‌ی لایه‌ی سوم است، عمل می‌کند.

• ساختار قاب اترنت

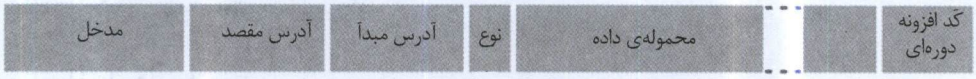
شکل ۵-۲۲ ساختار قاب داده‌ی اترنت را نشان می‌دهد. برای درک عمیق و قابل لمس قاب داده‌ی اترنت حالتی را درنظر بگیرید که یک داده گرام از سوی یک میزبان به دیگری در حال گسیل است و هر دو میزبان برروی یک اترنت LAN مشابه مانند شکل ۵-۲۱ قرار دارند. اگرچه محموله‌ی قاب اترنت موردنظر این بحث یک داده گرام است اما یک قاب اترنت می‌تواند حامل بسته‌های دیگری از لایه‌ی شبکه نیز باشد. اگر فرض شود کارت شبکه‌ی فرستنده‌ی A دارای آدرس MAC به‌صورت: AA-AA-AA-AA-AA-AA می‌باشد و کارت شبکه‌ی گیرنده‌ی B نیز آدرس MAC به گونه‌ی: BB-BB-BB-BB-BB-BB را اختیار می‌کند. کارت شبکه‌ی فرستنده داده گرام IP را در قالب یک قاب اترنت بسته‌بندی و آن را به سوی لایه‌ی فیزیکی هدایت می‌کند. کارت شبکه‌ی گیرنده، قاب را از لایه‌ی فیزیکی دریافت و داده گرام را از آن جدا می‌سازد و آن را به سمت لایه‌ی شبکه هدایت می‌کند. در این چارچوب 6 دامنه‌ی قاب اترنت در شکل ۵-۲۲ تحلیل می‌شود:

1- Switch



شکل ۵-۲۱♦ توپولوژی ستاره‌ی اترنت، اتصال گره‌ها از طریق هاب

۱- دامنه‌ی داده (46 تا 1500 بایت). این دامنه حامل داده گرام IP می باشد. حداکثر جزء انتقال (MTU) اترنت 1500 بایت می‌باشد. این به مفهوم آن است که اگر داده گرام فراتر از 1500 بایت باشد آن‌گاه میزبان فرستنده مجبور است داده گرام را براساس مطالب فصل ۴ قطعه‌بندی نماید. حداقل اندازه‌ی دامنه‌ی داده 46 بایت است. این به مفهوم آن است که اگر داده گرام IP کوچک‌تر از 46 بایت است. دامنه‌ی داده از سوی میزبان فرستنده بایستی اصطلاحاً **تزریق**[1] گردد تا 46 بایت کامل شود. هنگامی که از تزریق قاب استفاده شود داده‌ی انتقال یافته به لایه‌ی شبکه شامل یک داده گرام IP به همراه تزریقات می‌باشد. لایه‌ی شبکه با بهره‌گیری از طول دامنه در سرپیام داده گرام IP تزریقات انجام گرفته را حذف می‌نماید.



شکل ۵-۲۲ ♦ ساختار قاب اترنت

- **دامنه‌ی آدرس مقصد (6 بایت).** این دامنه شامل آدرس MAC کارت شبکه‌ی مقصد در قالبی مانند: BB-BB-BB-BB-BB-BB می‌باشد. هنگامی که کارت شبکه‌ی B قاب اترنت با آدرس مقصد اشاره شده و یا آدرس MAC فراپخش را دریافت می‌کند، محتویات درون دامنه‌ی داده‌ی قاب را به لایه‌ی شبکه انتقال می‌دهد در نتیجه اگر یک قاب با آدرس MAC دیگری دریافت گردد از آن قاب صرف‌نظر می‌شود.
- **دامنه‌ی آدرس مبدأ (6 بایت).** این دامنه شامل آدرس MAC کارت شبکه‌ی مبدأ است که قاب را به درون LAN تزریق می‌کند. در این مثال: AA-AA-AA-AA-AA-AA آدرس موردنظر است.
- **دامنه‌ی نوع (2 بایت).** دامنه‌ی نوع برای اترنت این امکان را فراهم می‌کند که پروتکل‌های لایه‌ی شبکه را تسهیم و به عبارت دیگر مالتی پلکس نماید. جهت درک این موضوع بایستی

1- Stuffed

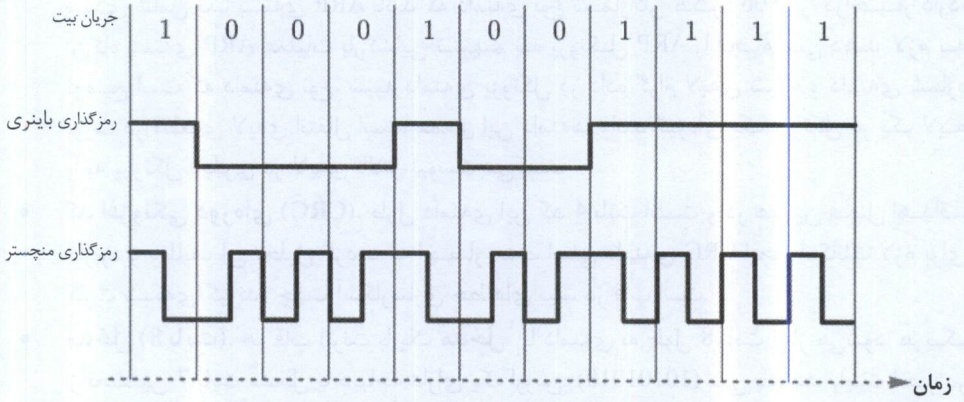
توجه داشت که میزبان‌ها علاوه بر IP از دیگر پروتکل‌های لایه‌ی شبکه نیز می‌توانند استفاده نمایند. در حقیقت یک میزبان ممکن است چندین پروتکل لایه‌ی شبکه را با بهره‌گیری از پروتکل‌های مختلف برای کاربردهای دیگر مورد استفاده قرار دهند. به همین دلیل هنگامی که یک قاب وارد کارت شبکه‌ی B می‌شود آن کارت نیازمند آگاهی از لایه‌ی پروتکل شبکه‌ای است که بخش دامنه‌ی داده به آن بایستی تحویل گردد. پروتکل IP و دیگران مانند IPX و یا Appletalk هریک شمارگان نوع استاندارد ویژه‌ی خویش را دارند. علاوه برآن پروتکل ARP نیز که در بخش قبل بررسی گردید شمارگان نوع ویژه خود را حمل می کند و بنابراین اگر قاب ورودی شامل یک بسته‌ی ARP باشد که دامنه‌ی نوع شمارگان هگزا: 0806 را دراختیار دارد، آن‌گاه بسته‌ی ARP عملیات بازگشایی تسهیم به پروتکل ARP را انجام می دهد. لازم به توضیح است که دامنه‌ی نوع، شبیه دامنه‌ی پروتکل در داده، گرام لایه‌ی شبکه و دامنه‌ی شماره پورت در قطعه‌ی لایه‌ی انتقال است. همه‌ی این دامنه‌ها مانند گیره‌ای یک پروتکل در یک لایه را به پروتکل دیگری در لایه‌ی بالاتر پیوست می‌کنند.

- **کد افزونگی دوره‌ای (CRC).** طول دامنه‌ی این کد 4 بایت است و در همین فصل اهداف، کاربرد و وظایف آن تحلیل گردید. به اختصار هدف اصلی دامنه‌ی CRC ایجاد امکانات لازم برای کارت شبکه‌ی گیرنده جهت آشکارسازی خطاهای بیت در قاب است.
- **مدخل (8 بایت).** هر قاب اترنت با یک **مدخل**[1] با دامنه‌ی به طول 8 بایت آغاز می‌شود. هریک از نخستین 7 بایت مدخل یا دیباچه دارای یک ارزش: (10101010) می باشند. بایت آخر در قاب (10101011) می‌باشد. نخستین 7 بایت مدخل باعث تحریک و فعال شدن کارت‌های شبکه و نیز همزمان شدن ساعت آن‌ها با ساعت فرستنده می‌شود. پرسش مطرح شده در ذهن آن است که چرا همزمانی ساعت‌های فرستنده و گیرنده لازم است؟ یادآوری این نکته ضروری است که اگر یک کارت شبکه‌ی فرستنده‌ای مانند A بستگی به نوع اترنت LAN آن مایل به ارسال قاب‌ها در سرعت‌های 10Mbps، 100 Mbps و یا 1 Gbps می‌باشد. از آن‌جائی‌که همه چیز نمی‌تواند مطابق میل باشد کارت A قادر نیست سرعت ارسال قاب را عیناً مطابق آن‌چه هدف می باشد، به انجام رساند و در نتیجه همواره نوعی انحراف از سرعت وجود دارد که لزوماً کارت‌های دیگر در شبکه‌ی LAN از آن آگاهی ندارند. یک کارت گیرنده می‌تواند به‌سادگی از طریق تمرکز برروی بیت‌های نخستین 7 بایت مدخل یا دیباچه‌ی قاب به هدف خود که تمرکز بر وسعت کارت شبکه‌ی A می باشد، دستیابی حاصل نماید. آخرین 2 بیت از 8 بایت مدخل یعنی دو ارزش 1 پی‌درپی کارت شبکه‌ی B را از جاری شدن اطلاعات که همان قاب‌ها می‌باشد آگاه می‌کنند.

پروتکل اترنت از انتقال باند پایه استفاده می‌کند به‌عبارت دیگر کارت شبکه سیگنال دیجیتالی را مستقیماً وارد کانال فراپخش می‌کند. کارت واسط تغییری را از لحاظ بسامد سیگنال ایجاد نمی‌کند. تجهیزاتی مانند ADSL و سیستم‌های مودم کابلی این تغییر بسامد را در سیگنال، قبل از تزریق آن به کانال فراپخش اعمال می‌کنند. بسیاری از فنآوری‌های اترنت مانند 10BASE-T از رمزگذاری منچستر که در شکل ۵-۲۳ نشان داده شده است بهره می‌گیرند. در رمزگذاری منچستر هر بیت شامل یک مرحله‌ی انتقالی است، به این مفهوم که یک ارزش 1 انتقالی از لبه‌ی بالا رونده به سوی پائین و یک

1- Preamble

ارزش 0 انتقالی از لبه‌ی کاهنده به طرف بالا را به‌ترتیب اختیار می‌کند. علت بهره‌گیری از رمزگذاری منچستر به این علت است که ساعت کارت‌های شبکه‌ی فرستنده و گیرنده کاملاً همزمان نمی‌باشند. از طریق افزودن یک مرحله‌ی انتقالی در میانه‌ی هر بیت میزبان گیرنده می‌تواند ساعت خود را با مانندش در میزبان ارسال کننده همزمان نماید. هنگامی که ساعت کارت شبکه‌ی گیرنده همزمان شود آن‌گاه گیرنده می‌تواند هر بیت را تشخیص داده و تعیین نماید ارزش آن 0 است یا 1. رمزگذاری منچستر یک عملیات لایه‌ی پیوند نمی‌باشد بلکه عملیاتی است از نوع لایه‌ی فیزیکی. علت بررسی آن در این مقطع به علت آن است که از آن در شبکه‌ی اترنت به‌صورت گسترده‌ای استفاده می‌شود.



شکل ۵-۲۳ ♦ رمزگذاری منچستر

• اترنت یک سرویس غیر اتصال‌گرای نامطمئن

کلیه‌ی فنآوری‌های اترنت خدمات از نوع غیر اتصال گرا برای لایه‌ی شبکه فراهم می‌آورند. به عبارت دیگر هنگامی که کارت شبکه‌ی A مایل به ارسال داده گرام به کارت شبکه‌ی B می‌باشد. ابتدا A داده گرام را در قالب یک قاب اترنت بسته‌بندی می‌کند و آن را به درون شبکه‌ی LAN گسیل می‌کند. این عملیات را بدون انجام دست‌دهی با B انجام می دهد. این خدمات غیراتصال گرای لایه‌ی دوم بسیار مشابه خدمات داده گرام IP در لایه‌ی سوم و خدمات غیراتصال گرای UDP در لایه‌ی چهارم است.

فنآوری اترنت یک سرویس غیرمطمئن را نیز برای لایه‌ی شبکه فراهم می‌کند. هنگامی که کارت شبکه‌ی B قاب داده‌ای را از A دریافت نماید آن فریم را از طریق CRC اعتبارسنجی می‌کند اما پیام تأئیدی را در صورت سالم بودن قاب ارسال نمی‌کند. چنانچه قاب آلوده باشد بازهم پیام (NACK) را به سوی فرستنده گسیل نخواهد کرد. هنگامی که یک قاب در اعتبارسنجی CRC مردود گردد، کارت شبکه‌ی B به‌سادگی از قاب صرف‌نظر می‌کند. بنابراین کارت شبکه‌ی A آگاهی از این که قاب انتقال یافته به کارت شبکه‌ی B رسیده است و از آزمون CRC نیز با موفقیت عبور کرده است، ندارد. فقدان در عدم اطمینان از انتقال در لایه‌ی پیوند سبب ساده و کم هزینه شدن اترنت می‌شود. این به مفهوم آن است که جریان داده گرام‌ها به لایه‌ی شبکه با وقفه‌های زمانی همراه است.

اگر فاصله و تأخیری به واسطه‌ی حذف قاب‌های اترنت حاصل شود آیا کاربردی در میزبان B نیز این وقفه را مشاهده می‌کند؟ در فصل ۳ آموختیم این موضوع بستگی به آن دارد که آیا کاربرد از پروتکل TCP و یا UDP بهره می‌گیرد؟ اگر کاربرد از UDP استفاده می‌کند آن‌گاه کاربرد در میزبان B

یقیناً فاصله‌ی ایجاد شده در داده را ملاحظه می‌کند. از طرف دیگر اگر کاربرد از TCP بهره می‌گیرد. آن‌گاه پروتکل TCP در میزبان B داده‌ی حذف شده در قاب را تأئید نمی‌کند و در نتیجه سبب انتقال مجدد TCP در میزبان A می‌شود. بدیهی است هنگام انتقال مجدد، داده از سوی TCP سرانجام داده به کارت شبکه‌ای که آن را حذف نموده است، بازگشت می‌کند. به عبارت دیگر اترنت انتقال مجدد داده را انجام می‌دهد، اگرچه از انتقال یک داده گرام کاملاً جدید یا داده‌ای تازه و یا یک داده گرام حامل داده‌ای که حداقل یک مرتبه گسیل شده است کاملاً ناآگاه است.

- **بازنگری تاریخ اترنت**

پس از اتمام تحصیلات عالی و پیوستن به شرکت زیراکس در اواسط دهه‌ی 1970 میلادی، مت کالفه در آن شرکت با نسل جدیدی از رایانه‌ها آشنا گردید به نام آلتو. این کامپیوترها از بسیاری جوانب مشابه رایانه‌های شخصی بودند که در دهه‌ی 1980 میلادی ظهور کردند. رایانه‌های آلتو ساخت شرکت زیراکس را می‌توان پیش قراولان PC دهه‌ی 1980 میلادی به شمار آورد. در آن زمان مت کالفه نیاز به شبکه‌سازی کامپیوترهای آلتو را از طریق روش‌های کم‌هزینه بررسی می‌نمود. از طرفی با استفاده از دانش جمع‌آوری نموده از شبکه‌های ARPA net و ALOHA net، و نیز پروتکل دستیابی تصادفی، مت‌کالفه به‌همراه همکار دیگرش دیویدباگز شبکه و پروتکل اترنت را پایه‌گذاری کردند.

پروتکل اترنت اولیه‌ی مت‌کالفه و باگز دارای سرعت 2.94 Mbps بود و حداکثر تا 256 میزبان پراکنده در فاصله‌ی تقریباً 2 کیلومتر از یکدیگر را پشتیبانی می‌نمود. مت کالفه و باگز موفق به ایجاد یک ائتلاف بین زیراکس، شرکت دیجیتال و اینتل جهت ایجاد یک شبکه‌ی اترنت استاندارد 10Mps شدند که از سوی IEEE نیز مصوب شد. از آن‌جائیکه شرکت زیراکس علاقه‌ی زیادی به تجاری نمودن اترنت نداشت، در سال 1979 مت کالفه پس از ترک زیراکس، شرکت تازه تأسیس خود 3Com را پایه‌گذاری نمود با این هدف که فنآوری شبکه را به ویژه در قالب اترنت توسعه و تجاری نماید. در اوایل دهه 1980 شرکت 3Com موفق به توسعه، تکمیل و بازاریابی کارت‌های شبکه‌ی اترنت شد و بی‌درنگ این کارت‌ها در PCهای شرکت IBM که بسیار محبوب بودند به‌کار گرفته شدند. مت کالفه در سال 1990 میلادی در حالی شرکت 3Com را ترک گفت که تعداد کارکنان این شرکت 2000 نفر و درآمد سالانه‌ی آن 400 میلیون دلار بود.

۵-۷ پروتکل دستیابی چندگانه‌ی اترنت (CSMA/CD)

هنگامی که اتصال گره‌ها به یکدیگر از طریق هاب به جای یک رایانه‌ی لایه‌ی پیوند انجام می‌شود. شبکه‌ی اترنت LAN یک شبکه‌ی کاملاً فراپخش است. به عبارت دیگر هنگامی که یک کارت شبکه قاب داده‌ای ارسال می‌کند همه‌ی کارت‌های دیگر در LAN، آن قاب را دریافت می‌کنند. از آن‌جائیکه اترنت می‌تواند ویژگی فراپخش را به کار گیرد بنابراین نیازمند یک پروتکل دستیابی چندگانه است. به همین منظور اترنت از پروتکل دستیابی چندگانه و پرآوازه‌ی CSMA/CD بهره می‌گیرد. در بخش ۵-۴ توضیح داده شد که پروتکل CSMA/CD عملیات زیر را انجام می‌دهد:

۱- کارت شبکه هر لحظه ممکن است آغاز به انتقال اطلاعات نماید به عبارت دیگر موضوع شیارهای زمانی در آن مطرح نمی‌باشد.

۲- یک کارت شبکه هنگامی که حس کند کارت‌های دیگر شبکه در حال گسیل قاب می‌باشند هرگز مبادرت به ارسال قاب نمی‌کند. به عبارت دیگر عملیات حس حامل را انجام می‌دهد.

۳- یک کارت شبکه‌ی ارسال کننده بی‌درنگ پس از شناسایی آن که کارت شبکه‌ی دیگری در حال انتقال قاب است عملیات خود را متوقف و عقب‌نشینی می‌کند. به عبارت دیگر عملیات آشکارسازی برخورد را انجام می‌دهد.

۴- قبل از آغاز مجدد انتقال، یک کارت شبکه یک مدت زمان تصادفی که در مقایسه با زمان انتقال فریم کوتاه است را تأمل می‌کند و سپس در صورت مناسب بودن اوضاع عملیات انتقال مجدد را انجام می‌دهد.

این رویکردها کارایی به مراتب بهتری در مقایسه با پروتکل شیاردار ALOHA در محیط LAN برای پروتکل CSMA/CD فراهم می‌آورد. در حقیقت اگر حداکثر تأخیر انتشار بین ایستگاه‌ها بسیار کوچک باشد، آن‌گاه کارایی CSMA/CD می‌تواند به نزدیک 100% نزدیک گردد. اما توجه به این نکته ضروری است که رویکردهای 2 و 3 در فهرست بالا نیازمند آن است که هر کارت شبکه‌ی اترنت قادر به موارد ذیل باشد:

- حس این نکته که چه هنگام برخی دیگر از کارت‌های شبکه در حال انتقال می‌باشند.
- آشکارسازی یک برخورد به هنگام انجام عملیات انتقال.

کارت‌های شبکه‌ی اترنت این وظایف را از طریق اندازه‌گیری سطوح ولتاژ قبل و در هنگام انتقال انجام می‌دهند. هر کارت شبکه پروتکل CSMA/CD را بدون هماهنگی صریح با دیگر کارت‌های شبکه برروی اترنت انجام می‌دهد. در درون خود یک کارت شبکه‌ی ویژه‌ی پروتکل CSMA/CD به‌گونه‌ی ذیل عمل می‌کند:

۱- کارت شبکه یک داده گرام را از لایه‌ی شبکه اخذ می‌کند. یک قاب اترنت را آماده‌سازی می‌کند و قاب را در بافر کارت شبکه قرار می‌دهد.

۲- چنان‌چه کارت شبکه حس نماید که کانال آزاد است، به عبارت دیگر اگر از سوی کانال انرژی سیگنال برای فاصله زمانی معادل 96 بیت وارد کارت شبکه نگردد آن‌گاه آغاز به انتقال قاب می‌کند. اگر کارت شبکه تشخیص دهد کانال مشغول است منتظر می‌ماند تا آن که انرژی سیگنال را در فاصله‌ی زمانی معادل 96 بیت دریافت نکند آن‌گاه آغاز به ارسال قاب می‌کند. (منظور از زمان 96 بیت یعنی طول زمانی که لازم است تا 96 بیت انتقال یابد.)

۳- هنگام انجام عملیات انتقال، کارت شبکه حضور انرژی سیگنال را که از دیگر کارت‌های شبکه وارد می‌شوند سنجش می‌کند. اگر کارت شبکه کل قاب را بدون تشخیص و دریافت انرژی سیگنال از کارت‌های دیگر انتقال دهد. کارت شبکه وظیفه‌اش را در قبال قاب کامل نموده است.

۴- اگر کارت شبکه انرژی سیگنالی را از دیگر کارت‌ها در هنگام ارسال دریافت نماید. انتقال قاب خود را متوقف می‌کند و جایگزین آن یک سیگنال انسداد 48 بیت می‌شود.

۵- پس از توقف ارسال یعنی گسیل سیگنال 48 بیت انسداد، کارت شبکه وارد فرآیندی می‌شود که اصطلاحاً به آن مرحله‌ی **عقب‌نشینی تصاعدی**[1] می‌گویند. این مرحله به مفهوم آن است که هنگام انتقال یک قاب و پس از شناسایی nامین برخورد در یک ردیف برای قاب داده، کارت شبکه یک ارزش تصادفی مانند k را از میان مجموعه‌ی $\{0,1,2,\ldots,2^m-1\}$ به‌گونه‌ای گزینش می‌کند که $m=\min(n,10)$ باشد. کارت شبکه آن‌گاه معادل زمانی 512*k بیت منتظر و آن‌گاه به مرحله‌ی 2 در موارد بالا بازگشت می‌کند.

1- Exponential backoff

چند نکته را بایستی در مورد پروتکل CSMA/CD مطرح نمود. هدف سیگنال انسداد اطمینان از این موضوع است که بقیه‌ی کارت‌های شبکه‌ی فرستنده نیز از برخورد آگاه می‌شوند.

به عنوان مثال اگر فرض شود یک کارت شبکه‌ای مانند A شروع به انتقال یک قاب می‌کند و دقیقاً قبل از آن که سیگنال A به کارت B وارد گردد، کارت شبکه‌ی B انتقال خود را آغاز نماید آن‌گاه B ناچار است پس از انتقال فقط چند بیت، عملیات گسیل خود را متوقف نماید. بدیهی است این چند بیت یقیناً به A انتقال یافته‌اند اما انرژی کافی برای تشخیص برخورد را برای A فراهم نمی‌کنند. برای اطمینان از آن که A برخورد به وقوع پیوسته را تشخیص می‌دهد، کارت شبکه‌ی B نسبت به گسیل سیگنال انسداد 48 بیت اقدام می‌کند.

در ادامه‌ی مطلب الگوریتم عقب نشینی تصاعدی بررسی می‌شود. نخستین مطلبی که در این مقطع لازم است به آن توجه شود آن است که در یک شبکه‌ی اترنت LAN با سرعت 10Mbps زمان لازم جهت انتقال یک بیت برابر 0.1 میکروثانیه است. اکنون فرض کنید یک کارت شبکه تلاش دارد برای نخستین مرتبه و در حالی‌که در حال انتقال است و برخوردی را شناسایی نموده است اقدام به انتقال یک قاب داده نماید. در این حالت کارت شبکه مقدار k=0 را با احتمال 0.5 و یا ممکن است با مقدار k=1 و با احتمال 0.5 گزینش نماید. اگر کارت شبکه k=0 را گزینش نماید آن‌گاه بی‌درنگ به مرحله‌ی 2 پس از ارسال سیگنال انسداد پرش می‌کند و اگر مقدار k=1 را گزینش نماید زمانی معادل 51.2 میکروثانیه قبل از بازگشت به مرحله‌ی 2 منتظر می‌شود. پس از دو برخورد، k با احتمال مشابه از میان مجموعه‌ی {0,1,2,3} گزینش می‌شود. بعد از 3 برخورد k بازهم با احتمال مشابه از میان مجموعه‌ی {0,1,2,3,4,5,6,7} گزینش می‌شود و پس از 10 و یا بیش از آن برخورد، k با احتمال برابر از بین مجموعه‌ی {0,1,2,...,1023} گزینش می‌شود. بنابراین اندازه مجموعه‌ها که k از میان آن‌ها گزینش می‌شود شامل رشدی تصاعدی نسبت به تعداد برخوردها تا مقدار n=10 می‌شود. به همین علت است که الگوریتم عقب‌نشینی اترنت اصطلاحاً معروف به **عقب‌نشینی تصاعدی** شده است.

بدیهی است استاندارد اترنت محدودیت برروی فاصله‌ی بین هر دو گروه را نیز به شبکه تحمیل می‌کند. این محدودیت جهت اطمینان از این موضوع است که اگر کارت شبکه‌ی A ارزش کم‌تری از K را در مقایسه با بقیه‌ی کارت‌های شبکه که دخالت در یک برخورد دارند گزینش نمود آن‌گاه A قادر به انتقال قاب خویش بدون تجربه‌ی یک برخورد جدید خواهد بود.

اکنون این پرسش‌ها مطرح می‌شود چرا از عقب‌نشینی تصاعدی استفاده می‌شود؟ چرا به عنوان مثال گزینش K از مجموعه‌ی {0,1,2,3,4,5,6,7} پس از هر برخورد انجام نگیرد؟ علت آن است هنگامی که یک کارت شبکه نخستین برخورد خود را تجربه نماید آگاهی از این واقعیت ندارد چه تعداد کارت شبکه‌ی دیگری در برخورد حضور دارند. اگر تعداد آن‌ها بسیار اندک باشد آن‌گاه معقول است که گزینش K از بین یک مجموعه‌ی اندکی از ارزش‌ها صورت پذیرد. از طرف دیگر اگر کارت‌های برخورد کننده انبوه باشند عاقلانه است که ارزش K از بین یک مجموعه‌ی بزرگ‌تر و با پراکندگی بیش‌تر از مجموعه‌ی ارزش‌ها انجام گیرد، چرا؟ زیرا با افزایش اندازه‌ی مجموعه پس از هر برخورد کارت شبکه خود را با وقوع این شرایط گوناگون سازگار می‌کند.

توجه به این نکته نیز ضروری است هر مرتبه یک کارت شبکه قاب جدیدی را جهت انتقال آماده‌سازی می‌کند، الگوریتم CSMA/CD که در بالا به آن اشاره رفته است را اجرا می‌کند. این عمل بدون درنظر گرفتن برخوردهایی که ممکن است در گذشته‌ی نزدیک روی داده باشد، انجام می‌شود. بنابراین احتمال آن که یک کارت با یک قاب جدید قادر شود بی‌درنگ قاب خود را پنهانی با موفقیت

در حالی انتقال دهد که چندین کارت شبکه‌ی دیگر در وضعیت عقب‌نشینی تصاعدی قرار دارند دور از انتظار نمی‌باشد.

• کارایی اترنت

هنگامی که فقط یک گره قاب داده جهت گسیل دارد، آن گره می‌تواند با حدکثر سرعت فنآوری اترنت یعنی 10Mbps، 100Mbps و یا 1Gbps عملیات انتقال را انجام دهد. اگر تعداد گره‌ها انبوه و چشمگیر باشند و اکثریت آن‌ها مایل به ارسال قاب‌های داده‌ی متعلق به خود باشند آن‌گاه نرخ مؤثر انتقال کانال به مراتب کم‌تر است. **کارآمدی اترنت** عبارت است از طول زمانی که در آن **(۱)** تعداد گره‌های فعال نسبتاً زیاد است و **(۲)** قاب‌های انتقال یافته برروی کانال بدون برخورد به انتشار خود ادامه می‌دهند و **(۳)** هریک از گره‌های فعال تعداد قابل ملاحظه‌ای قاب داده نیز جهت ارسال در اختیار دارند. برای آن که یک معیار تقریبی نزدیکی از کارایی اترنت ارایه گردد، فرض کنید: d_{prop} حداکثر زمان لازم جهت انتشار انرژی سیگنال بین هر دو کارت شبکه باشد. هم‌چنین: d_{trans} را زمان لازم جهت انتقال یک قاب اترنت با حداکثر طول ممکن در نظر بگیرید. این زمان تقریباً 1.2ms برای اترنت با سرعت 10 Mbps است. در این بخش ما به دنبال اثبات معادله‌ی کارایی اترنت نمی‌باشیم، در صورت علاقه خواننده می‌تواند به مرجع [lam 1980] مراجعه نماید. اما در هرحال حداکثر کارایی اترنت از طریق رابطه‌ی (۵-۹) محاسبه می‌شود:

$$\text{حداکثر کارایی اترنت } y = \frac{1}{1+(5d_{prop}/d_{trans})} \qquad (۹-۵)$$

از این رابطه ملاحظه می‌شود که اگر مقدار d_{prop} به سمت صفر میل نماید، کارایی به طرف 1 نزدیک می‌شود. بدیهی است این واقعیت با آن‌چه در فصل اول تحلیل شد در توافق کامل است، به این مفهوم که اگر زمان تاخیر انتشار برابر صفر شود گره‌های برخورد کننده بی‌درنگ بدون اتلاف ظرفیت کانال، متوقف می‌شوند. هم‌چنین هرچه بر مقدار d_{trans} افزوده و مقدار آن چشمگیر شود آن‌گاه کارایی به سمت 1 نزدیک می‌شود، زیرا هنگامی که یک قاب کانال را در اختیار می‌گیرد تلاش می‌کند تلاش را برای مدت نسبتاً طولانی نزد خود نگاه دارد، در نتیجه در اغلب اوقات کانال کارآمدی خود را نشان می‌دهد.

• فناوری‌های اترنت

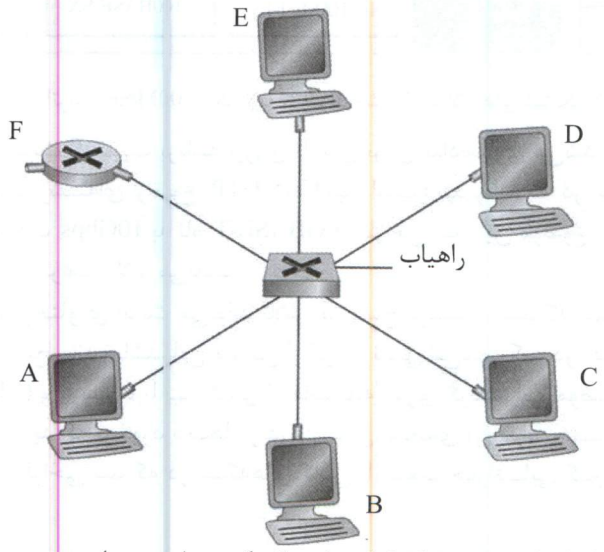
در مطالب بالا اشاره‌ی ما به اترنت مانند یک پروتکل استاندارد بود. اما در حقیقت اترنت در قالب‌های گوناگون و متنوع بی‌شماری ارایه می‌شود. از جمله با نام‌های اختصاری مانند: 10BASE-T، 10BASE-2 ، 100BASE-T ، 100BASE-LX و 10BASE-T ، این نمونه‌ها و بسیاری دیگر از فناوری‌های اترنت در طول سالیان از سوی کارگروه: IEEE802.3CSMA/CD استاندارد شده‌اند. آخرین نسخه‌ی استانداردهای این کارگروه در مستند [IEEE802.3 2012] قابل مطالعه می‌باشد. اگرچه این نام‌های اختصاری گیج کننده به نظر می‌رسند اما نظم و قانون‌مندی ویژه‌ای در آن‌ها رعایت شده است. بخش ابتدایی نام اختصاری اشاره به سرعت‌های استاندارد به ترتیب: 10، 100 و 1000 مگابیت بر ثانیه می‌کند و 10G به مفهوم 10 گیگابیت بر ثانیه است. واژه **BASE** اشاره به باند پایه‌ی اترنت می‌کند که مفهوم آن یعنی رسانه‌ی فیزیکی که فقط ترافیک اترنت را حمل می‌کند. تقریباً همه‌ی استانداردهای 802.3 برای کار در باند پایه‌ی اترنت طراحی شده‌اند. سرانجام بخش آخر که اشاره به رسانه‌ی فیزیکی می‌کند. پروتکل اترنت هر دو ویژگی لایه‌ی پیوند و فیزیکی را در خود جای داده است و از طریق انواع گوناگونی از رسانه‌های فیزیکی مانند کابل‌های هم‌محور، جفت رشته‌ی مسی (TP) و تارهای نوری بهره می‌گیرد.

حرف "T" به مفهوم جفت رشته‌ی مسی (TP) می‌باشد. از لحاظ پیشینه‌ی تاریخی اترنت به صورت یک قطعه‌ی کابل هم‌محور مانند آن‌چه شکل ۵-۲۰ نشان می‌دهد پیاده‌سازی شده است.

مشخصات دو استاندارد ابتدایی 10BASE-2 و 10BASE-5 پروتکل اترنت 10Mbps برروی دو نوع کابل هم‌محور که هریک محدود به طول 500 متر شده بود را پیاده‌سازی می‌نمود. طول فراتر از 500 متر از طریق بهره‌گیری از **تکرارکننده‌ها** امکان‌پذیر است که وسیله‌ای با مشخصات لایه‌ی فیزیکی که انرژی سیگنال را در ورودی خود دریافت و از طریق تولید مجدد، سیگنال تقویت شده‌ی آن را در خروجی تکرار می‌نماید. یک کابل هم‌محور براساس شکل ۵-۲۰ جلوه‌ی درستی را از نقطه‌نظر اترنت به عنوان یک رسانه‌ی فراپخش در پیش روی می‌گذارد. همه‌ی قاب‌ها از سوی یک کارت شبکه ارسال می‌شوند و از سوی هریک از کارت‌های دیگر که اتصال فیزیکی با رسانه دارد دریافت می‌شوند. پروتکل اترنت CSMA/CD نیز در این رابطه چالش دستیابی چندگانه را نیز برطرف می سازد زیرا گره‌ها (کارت‌های شبکه) به‌سادگی به کابل متصل می‌شوند و یک شبکه‌ی محلی (LAN) را تشکیل می‌دهند.

بدیهی است پس از تکامل اترنت در طول سالیان اکنون این پروتکل کاملاً دگرگون با نوع اصلی خود عمل می‌نماید. در اغلب شبکه‌های اترنت امروزی کارت‌های شبکه از طریق قطعه‌های نقطه به نقطه که هریک از آن‌ها از جفت رشته‌های مسی (TP) و یا تار نوری تشکیل شده‌اند به یک راهیاب (سوئیچ) متصل می‌شوند. شکل ۵-۲۴ ساختار اترنت LAN امروز را نشان می‌دهد.

در اواسط دهه‌ی 1990 میلادی اترنت برمبنای سرعت 100Mbps استاندارد شد که 10 برابر سرعت اترنت 10Mbps اولیه بود. پروتکل اصلی اترنت MAC و قالب فریم داده‌ی اترنت به همان صورت اولیه باقی مانده‌اند اما لایه‌ی فیزیکی سریع‌تر برای رشته‌های مسی (100BASE-TX) و تار نوری که عبارتند از: 100BASE-FX ، 100BASE-SX و 100BASE-BX تعریف، طراحی و پیشنهاد شده‌اند. شکل ۵-۲۵ استانداردهای گوناگون اترنت و پروتکل MAC مشترک و قالب فریم‌ها را در اترنت نشان می‌دهد. اترنت 100Mpbs محدود به طول 100 متر برروی کابل (TP) و چندین کیلومتر برروی رسانه‌ی تارنوری می‌باشد که اجازه می‌دهد راهیاب‌های اترنت در ساختمان‌های گوناگون به یکدیگر متصل شوند.



شکل ۵-۲۴ ♦ یک راهیاب لایه‌ی پیوند که 6 گره را به یکدیگر متصل نموده است.

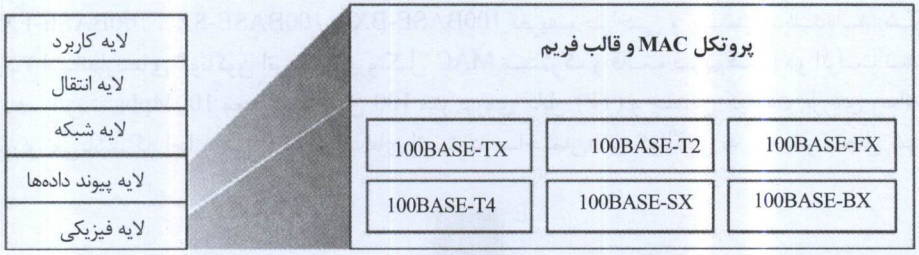
اترنت با سرعت یک یا چندین گیگا بایت برثانیه (Gbps) نوع تکامل یافته‌ی موفق اترنت استاندارد 10 و 100 مگابیت برثانیه (Mbps) می‌باشد. قالب‌بندی قاب داده و نیز پروتکل اصلی اترنت MAC به‌همان صورت اصلی بدون تغییر باقی مانده‌اند. اترنت در سرعت گیگا بیت برثانیه سازگاری کامل خود با تجهیزات ابتدایی اترنت را حفظ نموده است و این استاندارد که اصطلاحاً به آن **IEEE802.33** گفته می‌شود دارای ویژگی‌های ذیل می‌باشد:

۱- از قالب فریم استاندارد اترنت در شکل ۵-۲۲ بهره می‌گیرد و بنابراین خود را سازگار با فنآوری پیشین اترنت یعنی 10BASE-T و 100BASE-T قرار می‌دهد. این موضوع یکپارچه‌سازی ساده‌ی اترنت با سرعت گیگابیت بر ثانیه را در تجهیزات اترنت سنتی نصب شده امکان‌پذیر می‌سازد.

۲- پیوندهای نقطه به نقطه را امکان‌پذیر می‌کند و علاوه برآن بهره‌گیری از کانال‌های فراپخش مشترک را نیز میسر می‌سازد. پیوندهای نقطه به نقطه از راهیاب بهره می‌گیرند در حالی‌که کانال‌های فراپخش از هاب استفاده می‌کنند. در واژگان اترنت اصطلاحاً نام دیگری به هاب داده‌اند و آن را **توزیع کنندگان ذخیره‌سازی**[1] می‌نامند.

۳- این پروتکل از CSMA/CD جهت اشتراک کانال‌های فراپخش استفاده می‌کند. جهت کارایی قابل قبول حداکثر فاصله‌ی بین گره‌ها بسیار محدود می‌باشد.

۴- عملیات دوطرفه‌ی همزمان را کاملاً امکان‌پذیر می‌سازد. به ویژه در سرعت 1000Mbps برای کانال‌های نقطه به نقطه این امکان به آسانی میسر است.



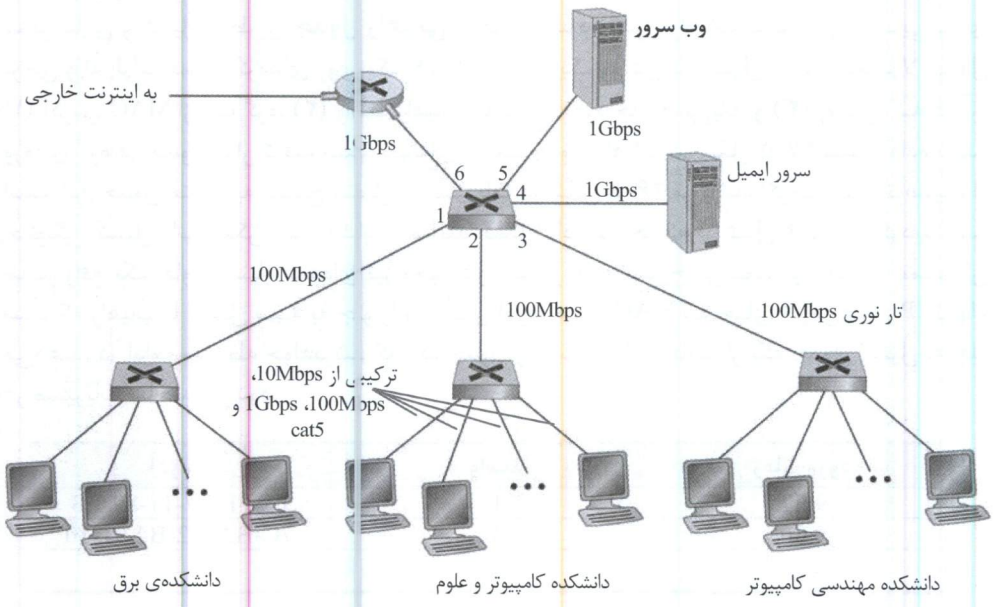
شکل ۵-۲۵ ♦ استاندارد اترنت 100Mbps، یک لایه‌ی پیوند مشترک و لایه‌های فیزیکی گوناگون

در آغاز، پروتکل اترنت گیگابیت برثانیه برروی تارهای نوری پیاده‌سازی می‌شد اما اکنون این امکان برروی کابل‌های جفت رشته‌ای از نوع: Cat5-UTP نیز امکان‌پذیر است. در سال 2007 میلادی استاندارد اترنت با سرعت 10Gbps به نام 10GBASE-T طراحی شد. این موضوع خودگواه دیگری بر ظرفیت اترنت LAN با سرعت بالاتر می‌باشد.

پایان بخش موضوع فنآوری اترنت می‌تواند تلاش در پاسخ پرسشی باشد که ممکن است تاکنون در ذهن خواننده بدون پاسخ مانده باشد. این پرسش از این‌جا ناشی می‌شود که در آغاز اترنت برمبنای ساختار گذرگاه (BUS) و یا ستاره با بهره‌گیری از هاب پایه‌گذاری گردید. این موضوع سبب پیدایش پیوندهای فراپخش گردید که برخورد قاب‌ها در درون کانال نتیجه‌ی آن بود. جهت حل این چالش پروتکل CSMA/CD طراحی شد که در شبکه‌های کابلی با وسعت جغرافیایی کم بسیار مؤثر عمل

1- Buffered distributers

می‌نماید. از طرفی اترنت امروز برمبنای ساختار ستاره با بهره‌گیری راهیاب‌ها (سوئیچ‌ها) می‌باشد که این تجهیزات از روش راهیابی ذخیره و گسیل بسته به جلو بهره می‌گیرند و پرسش آن است که آیا با این رویکرد هنوز هم نیازی به پروتکل اترنت MAC می‌باشد؟ در بخش بعدی همین فصل ملاحظه خواهد شد که وظیفه‌ی اصلی یک راهیاب هماهنگی عملیات انتقال خویش به‌گونه‌ای می‌باشد که هرگز بیش از یک قاب داده در هر لحظه از زمان برروی واسط مشابه قرار نگیرد. علاوه بر آن راهیاب‌های مدرن کاملاً دوطرفه‌ی همزمان بوده به‌گونه‌ای که یک راهیاب و یک گره هریک می‌توانند فریم‌ها را به یکدیگر به روش همزمان بدون تداخل انجام دهند، به عبارت دیگر در یک شبکه‌ی LAN برمبنای اترنت راهیابی شده برخوردی روی نمی‌دهد و بنابراین نیازی هم بری یک پروتکل MAC نمی‌باشد. در یک نتیجه‌گیری کلی می‌توان دریافت که اترنت امروز نسبت به آن چه دیروز تعریف شده بود بسیار تکامل یافته است به‌گونه‌ای که در بسیاری از مواقع نیاز به پروتکل MAC هم نمی‌باشد! اما با این توصیف آیا این پروتکل هنوز اترنت است؟ پاسخ هنوز هم از **نقطه نظر تعریف‌آری** است. آن‌چه جالب توجه است این حقیقت است که اگرچه اترنت دستخوش تغییرات وسیع و گسترده‌ای شده است اما یک بخش آن در طول 30 سال گذشته همواره ثابت و پایدار باقی مانده است و آن قالب **قاب داده‌ی** اترنت می‌باشد. شاید به‌همین دلیل می‌توان هنوز هم شبکه‌های LAN را ترنت نامید.



شکل ۵-۲۶ ♦ شبکه‌ی یک نهاد که از ترکیب هاب، راهیاب و یک مسیریاب بهره می‌گیرد.

۵-۸ راهیاب‌های لایه‌ی پیوند

براساس آن‌چه شکل ۵-۲۶ نشان می‌دهد شبکه‌ی اترنت LAN مدرن از توپولوژی ستاره بهره میگیرد که در آن هر گره به یک راهیاب مرکزی متصل است. تا این مرحله از مطالب فصل ۵ تحلیل آشکاری از چگونگی عملکرد یک راهیاب انجام نشده است. نقش بنیادی یک راهیاب دریافت قاب‌های ورودی لایه‌ی پیوند و انتقال آن‌ها برروی پیوندهای خروجی می‌باشد. یک راهیاب اصطلاحاً در برابر گره‌ها

شفاف است. مفهوم شفافیت گره به این معنی است که یک گره قاب را به گره‌ی دیگر آدرس‌دهی و هدایت می‌کند. قاب‌ها به‌سوی راهیاب آدرس‌دهی نمی‌شوند، بنابراین قاب داده به درون شبکه‌ی LAN وارد می‌شود بدون آگاهی از این واقعیت که یک سوئیچ آن را دریافت و سپس به گره‌های دیگر انتقال می‌دهد. سرعتی که قاب‌ها از طریق آن به هریک از واسط‌های خروجی راهیاب وارد می‌شوند ممکن است موقتاً فراتر از ظرفیت پیوند آن واسط باشد. راه کار این چالش آن است که واسط‌های خروجی راهیاب شامل بافرهایی می‌شوند که بسیار مشابه واسط‌های خروجی مسیریابی است که از بافرها برای گسیل داده گرام‌ها بهره می‌گرفت. در ادامه چگونگی عملکرد راهیاب در حالت‌های گوناگون تحلیل می‌شود.

• صافی‌سازی و گسیل جلو

صافی‌سازی و یا **فیلترینگ** عملیاتی است که سوئیچ انجام می‌دهد تا مشخص نماید آیا یک قاب داده بایستی به واسط شبکه منتقل گردد و یا آن که حذف شود. **گسیل به جلو** که قبلاً نیز در بحث مربوط به مسریاب‌ها به آن اشاره رفت عملیاتی است که راهیاب انجام می‌دهد تا مشخص نماید قاب داده را به کدام‌یک از واسط‌های شبکه بایستی گسیل نماید و پس از آن فریم را به آن واسط منتقل می‌کند. صافی‌سازی و گسیل از طریق **جدول راهیابی**[1] انجام می‌شود. جدول راهیاب شامل ورودی‌هایی برای برخی و نه الزاماً همه‌ی گره‌های روی یک LAN می‌باشد. یک ورودی در جدول راهیاب معمولاً شامل **(۱)** آدرس MAC از یک گره، **(۲)** واسط راهیاب که به یک گره خاتمه می‌یابد و **(۳)** زمانی که در آن ورودی گره در جدول قرار گرفته است. نمونه‌ای از یک جدول راهیاب در شکل ۵-۲۷ نشان داده شده است. این جدول متعلق به سوئیچ متصل به مسیریاب در شکل ۵-۲۶ می‌باشد. اگرچه این توصیف از چگونگی گسیل قاب ممکن است مشابه با فرآیند گسیل داده گرام‌ها که در فصل ۴ بررسی گردید باشد اما در واقع یک تفاوت بسیار عمده‌ای نیز وجود دارد که در ادامه تشریح می‌شود. این تفاوت عمده آن است که راهیاب‌ها گسیل بسته به جلو را برمبنای آدرس‌های MAC به جای آدرس‌های IP انجام می‌دهند. در ادامه ملاحظه خواهد شد که یک جدول راهیاب بسیار متفاوت از یک جدول گسیل به جلو در مسیریاب ساخته می‌شود.

آدرس	واسط	زمان ورود
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
---	---	---

شکل ۵-۲۷ ♦ بخشی از جدول راهیاب برای راهیاب متصل به روتر شکل ۵-۲۶

جهت درک آن که چگونه صافی‌سازی وگسیل جلو در راهیاب عمل می‌کند، قاب داده‌ای را درنظر آورید که آدرس MAC مقصد آن: DD-DD-DD-DD-DD-DD باشد و این قاب وارد راهیاب برروی واسط x می‌شود. راهیاب جدول خود را با آدرس MAC اشاره شد شاخص‌گذاری می‌کند. فرآیند شاخص‌گذاری ممکن است سه وضعیت را در پیش روی جدول راهیاب قرار دهد:

1- Switching tables

۱- در جدول راهیاب یک ورودی، به عبارت دیگر اطلاعاتی برای آدرس مقصد: DD-DD-DD-DD-DD-DD وجود ندارد. در این حالت راهیاب نسخه‌هایی از قاب داده را به بافرهای خروجی همه‌ی واسط‌ها به جز x گسیل می‌کند. به عبارت دیگر اگر ورودی برای آدرس مقصد وجود ندارد، راهیاب فریم را فراپخش می‌کند

۲- در جدول یک ورودی، برای آدرس مقصد اشاره شده‌ی وابسته به واسط x وجود دارد. در این حالت قاب از یک قطعه LAN وارد می‌شود که شامل کارت شبکه‌ی با آدرس: DD-DD-DD-DD-DD-DD می‌باشد. در این حالت نیازی نیست که فریم داده به هر یک از واسط‌های دیگر گسیل گردد، بلکه راهیاب عملیات صافی‌سازی را از طریق حذف قاب انجام می‌دهد.

۳- در جدول اطلاعاتی، برای آدرس مقصد اشاره شده با واسط $y \neq x$ وجود دارد. به عبارت دیگر برای واسطی غیر از x ورودی تعریف شده است. در این حالت قاب نیازمند گسیل جلو به قطعه‌ای از LAN است که گره متصل به واسط y می‌باشد راهیاب عملیات گسیل جلو را از طریق جایگذاری قاب در یک بافر خروجی انجام می‌دهد که به سوی واسط y می‌باشد.

اگر این قواعد را در ارتباط با راهیاب متصل به مسیریاب شکل ۵-۲۶ و جدول آن یعنی ۵-۲۷ اعمال کنیم. آن‌گاه اگر فرض شود یک فریم داده با آدرس MAC مقصد: 62-FE-F7-11-89-A3 وارد راهیاب از واسط 1 شود. راهیاب جدول خود را سنجش و ملاحظه می‌کندکه مقصد موردنظر برروی قطعه‌ای از LAN قرار دارد که آن قطعه به واسط 1 متصل شده است. این به معنی آن است که در حال حاضر قاب برروی قطعه‌ای از LAN فراپخش شده است که شامل مقصد می‌باشد. سوئیچ بنابراین فریم را فیلتر می‌کند به عبارت دیگر از آن صرف‌نظر می‌کند. اکنون فرض کنید یک قاب با آدرس مقصد مشابه از واسط 2 وارد می‌شود. بازهم راهیاب جدول خود را سنجش می‌کند و مشاهده می‌کند که مقصد به سوی واسط 1 می‌باشد، بنابراین فریم را به بافر خروجی در سمت واسط 1 هدایت می‌کند. از این مثال این نکته بایستی آشکار شده باشد که تا زمانی که جدول راهیاب کامل و دقیق است، راهیاب قاب‌ها را به سوی مقصد بدون انجام عملیات فراپخش، گسیل می‌کند. به بیان دیگر سوئیچ به مراتب هوشمندتر از هاب می‌باشد. اکنون این پرسش‌ها مطرح می‌شود:

۱- چگونه جدول راهیاب از ابتدا پیکربندی می‌شود؟
۲- آیا پروتکل لایه‌ی پیوند معادل پروتکل مسیریابی لایه‌ی شبکه وجود دارد؟
۳- آیا پیکربندی جدول راهیاب دستی انجام می‌شود؟

• خود یادگیری

یک راهیاب این ویژگی شگفت‌انگیز را دارد که جدول خویش را به‌گونه‌ی خودکار، پویا و مستقل تعریف می‌کند، بی‌نیاز از دخالت مدیر شبکه و یا هر پروتکل پیکربندی. به عبارت دیگر راهیاب‌ها **خودآموز** می‌باشند. این توان‌مندی سوئیچ از طریق راه‌کارهای ذیل انجام می‌شود:

۱- جدول راهیاب در آغاز تهی می‌باشد.

۲- برای هر قاب ورودی که برروی یک واسط دریافت می‌شود، آن‌چه راهیاب در جدول خود ذخیره می‌کند عبارتند از: **(۱)** آدرس MAC در دامنه‌ی **آدرس مبدأ** قاب، **(۲)** واسطی که از طریق آن قاب دریافت شده است، **(۳)** زمان (ساعت) جاری. در این فرآیند است که راهیاب به

ثبت قطعه‌ای از LAN در جدول خود می‌پردازد که گره‌ی ارسال کننده در آن قطعه ساکن می‌باشد. اگر هریک از گره‌ها در LAN، سرانجام یک قاب گسیل نمایند، آن‌گاه هرگره نهایتاً در جدول ثبت خواهد شد.

۳- راهیاب هنگامی یک آدرس را از جدول حذف می‌کند که قاب‌هایی با آدرس مبدأ پس از گذشت مدت زمانی که اصطلاحاً به آن **زمان کهولت**[1] گفته می‌شود، دریافت نگردد. در این حالت اگر یک PC با نوع دیگری جایگزین شود که از لحاظ کارت شبکه متفاوت می‌باشد. آن‌گاه آدرس MAC متعلق به PC اصلی سرانجام از جدول راهیاب پاکسازی می‌شود.

در ادامه‌ی مراحل خود یادگیری راهیاب متصل به روتر شکل ۵-۲۶ و جدول وابسته به آن در شکل ۵-۲۷ بررسی می‌شود. فرض کنید در لحظه‌ی 9:39 یک قاب با آدرس مبدأ: 01-12-23-34-45-56 از واسط 2 وارد می‌شود. علاوه بر آن فرض کنید این آدرس در جدول راهیاب موجود نمی‌باشد. آن‌گاه راهیاب ورودی جدید را برمبنای شکل ۵-۲۸ وارد جدول راهیاب می‌نماید:

آدرس	واسط	زمان
01-12-23-34-45-56	2	9:39
62-FE-F7-11-89-1A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
---	---	---

شکل ۵-۲۸ ♦ راهیاب در مورد مکان یک کارت شبکه با آدرس: 01-12-23-34-45-56 یادگیری می‌کند.

با ادامه‌ی مثال این راهیاب، فرض کنید زمان کهولت برای سوئیچ 60 دقیقه باشد و هیچ قاب داده‌ی دیگری با آدرس مبدأ: 62-FE-F7-11-89-A3 بین دو زمان 9:32 ,10:32 وارد راهیاب نشده است. آن‌گاه در زمان 10:32، سوئیچ این آدرس را از جدول خود حذف می‌نماید. ملاحظه می‌شود که عملکرد یک راهیاب کاملاً مستقل و بدون نیاز به هیچ‌گونه هدایتی می‌باشد. راهیاب‌ها کاملاً دوطرفه‌ی همزمان می‌باشند زیرا برای هر پیوندی که یک کارت شبکه را به سوئیچ متصل می‌کند آن گره و راهیاب همزمان بدون وقوع برخورد می‌توانند به مبادله‌ی اطلاعات بپردازند.

ویژگی‌های راهیابی لایه‌ی پیوند

پس از توصیف عملیات اساسی یک راهیاب لایه‌ی پیوند، اکنون نوبت آن رسیده است که ویژگی‌های آن‌ها بررسی شود. با بهره‌گیری از شبکه‌ی LAN در شکل ۵-۲۴ چندین برجستگی را می‌توان برای بهره‌گیری از راهیاب برشمرد. این ویژگی‌ها در مقایسه با پیوندهای فراپخش مانند گذرگاه‌ها و یا توپولوژی ستاره برمبنای هاب می‌باشد:

۱- **حذف وقوع برخورد.** در یک شبکه‌ی LAN ساخته شده از راهیاب‌ها و بدون هاب هیچ‌گونه اتلاف پهنای باند در اثر وقوع برخورد وجود ندارد! راهیاب‌ها قاب‌های داده را بافر می‌نمایند و هرگز بیش از یک فریم را برروی یک قطعه در هر لحظه انتقال نمی‌دهد. مانند مسیریاب حداکثر نهفتگی انباشتی یک سوئیچ عبارت است از مجموع همه‌ی سرعت‌های واسط راهیاب. بنابراین راهیاب‌ها سبب بهبودی چشمگیر کارایی در شبکه‌های LAN با پیوندهای فراپخش می‌شوند.

1- Aging time

۲- **پیوندهای متباین.** از آن‌جائیکه یک راهیاب یک پیوند را از دیگری در انزوا قرار می‌دهد، پیوندهای گوناگون در LAN قادر به انجام عملیات در سرعت‌های مختلف می‌شوند و برروی رسانه‌های گوناگون نیز قابل پیاده‌سازی می‌باشند. در شکل ۵-۲۴ به عنوان مثال گره‌ی A ممکن است از طریق یک پیوند 10Mbps از نوع 10BASE-T، گره‌ی B از طریق پیوند 100Mbps تار نوری از نوع 100BASE-FX، و C نیز از طریق پیوند 1Gbps از نوع مسی 1000BASE-T متصل شده باشند. بنابراین می‌توان به این نتیجه رسید که یک راهیاب مناسب ترکیب هر دو تجهیزات جدید و قدیمی با یکدیگر می‌باشد.

۳- **مدیریت.** علاوه بر فراهم نمودن امنیت بهینه و گسترش یافته که در ادامه توضیح داده می‌شود، یک راهیاب مدیریت شبکه را نیز ساده می‌کند. به عنوان مثال اگر یک کارت شبکه دچار عیب باشد و مرتباً قاب‌های اترنت را گسیل نماید که اصطلاحاً به آن **کارت شبکه تندگو**[1] گفته می‌شود یک راهیاب می‌تواند ایراد را شناسایی و به گونه‌ی داخلی کارت شبکه‌ی معیوب را از شبکه قطع نماید. با این ویژگی مدیر شبکه نیازی به انجام عملیات نگهداری و تعمیر ندارد. به گونه‌ی مشابه یک کابل قطع شده نیز سبب می‌شود که راهیاب همان گره‌ای را از شبکه جدا نماید که از طریق آن کابل معیوب به راهیاب متصل شده است. در دوران بهره‌گیری از کابل هم محور بسیاری از مدیران شبکه مجبور بودند ساعت‌ها و حتی روزها به دنبال عیب و سپس رفع آن در شبکه بپردازند. همان‌گونه که در فصل ۹ بررسی خواهد شد راهیاب‌ها قادر به جمع‌آوری اطلاعات آماری از نقطه نظر کاربرد پهنای باند، برخوردها، نوع ترافیک و نیز در دسترس قرار دادن این اطلاعات به مدیر شبکه می‌باشند. این اطلاعات می‌تواند جهت عیب‌یابی و رفع آن مورد استفاده قرار گیرد، هم‌چنین می‌توان جهت روند تکامل شبکه‌ی LAN نیز از آن بهره گرفت. پژوهشگران همواره به دنبال افزودن عملکردهای مدیریتی بیش‌تر به درون اترنت LAN می‌باشند.

۴- **بوئیدن یک شبکه‌ی (LAN) راهیابی شده (مسموم‌سازی راهیاب).** هنگامی که یک گره به راهیابی متصل شده است. آن راهیاب فقط قاب‌هایی را دریافت می‌کند که انحصاراً برای آن گسیل شده‌اند. به عنوان مثال شبکه‌ی LAN راهیابی شده در شکل ۵-۲۴ را درنظر آورید. هنگامی که گره‌ی A قاب داده‌ای را به گره‌ی B گسیل می‌کند و یک ورودی برای گره‌ی B در جدول راهیاب وجود دارد، آن‌گاه راهیاب قاب را فقط به گره‌ی B گسیل می‌کند. اگر گره‌ای مانند C در حال انجام عملیات **استشمام**[2] باشد. در این حالت گره‌ی C قادر به بوئیدن این مبادله‌ی قاب از A به B نمی‌باشد. بنابراین **در** محیط یک LAN راهیابی شده در مقایسه‌ی شبکه‌ی با پیوند فراپخش مانند 802.11 یا اترنت برمبنای هاب عملیات بوئیدن قاب‌ها از سوی یک مهاجم بسیار دشوار است (عملیات بوئیدن در ارتباط با سرقت قاب‌ها و در نتیجه تهدید شبکه می‌باشد). از آن‌جائیکه راهیاب، فراپخش آن گروه از قاب‌هایی را انجام می‌دهد که آدرس‌های مقصد آن‌ها در جدول سوئیچ حضور ندارند، سارق قاب در C هنوز هم قادر است برخی از فریم‌ها که انحصاراً به سوی C آدرس‌دهی نشده‌اند را استشمام نماید. علاوه بر آن یک مسموم‌ساز راهیاب می‌تواند همه‌ی فریم‌های اترنت فراپخش را که دارای آدرس فراپخش مقصد:

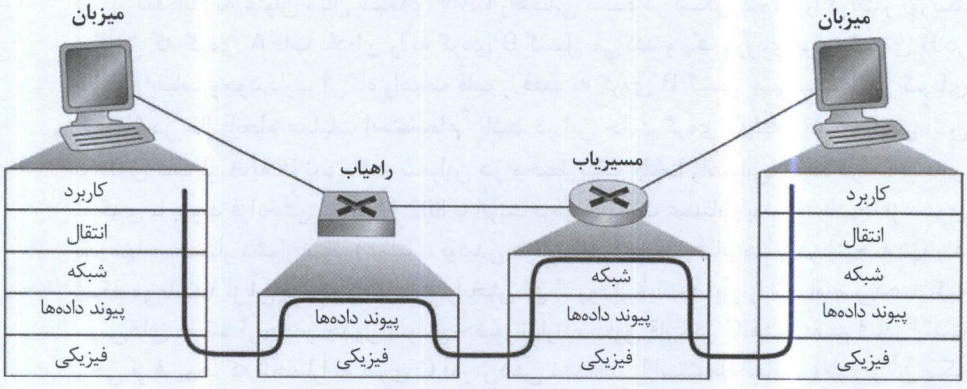
1- Jabbering adapter
2- Sniff

FF-FF-FF-FF-FF-FF می‌باشند، شنود نماید. یک نوع معروف از تهاجم به یک راهیاب، **مسموم‌ساز سوئیچ**[1] نام دارد. این گونه از بوئیدن سبب می‌شود انبوه بی‌شماری از بسته‌ها با آدرس‌های MAC مبدأ جعلی به راهیاب گسیل گردد. در نتیجه‌ی این عمل جدول راهیاب از ورودی‌های جعلی انباشته شده و فضایی برای آدرس‌های MAC گره‌های قانونی و مشروع وجود نخواهد داشت. این موضوع باعث می‌شود راهیاب قاب‌های بیش‌تری را فراپخش نماید که در نتیجه‌ی آن مهاجم اقدام به بوئیدن بسته‌ها می‌کند. از آن‌جائیکه این گونه تهاجم پیچیدگی فراوانی را برای مهاجم به همراه دارد راهیاب‌ها به مراتب کم‌تر از شبکه‌های بی‌سیم LAN (WiFi) و شبکه‌های برمبنای هاب در معرض تهدید می‌باشند.

۵-۹ مقایسه‌ی راهیاب‌ها و مسیریاب‌ها

در فصل ۴ آموختیم مسیریاب‌ها راهیابی بسته‌ها از طریق روش ذخیره و گسیل جلو انجام می دهند به عبارت دیگر آن‌ها راهیاب‌های بسته هستند که داده گرام‌ها را با بهره‌گیری از آدرس‌های لایه‌ی شبکه گسیل می‌نماید. اگرچه یک راهیاب نیز شامل روتر از این لحاظ می‌باشد اما از نقطه نظری بنیادی کاملاً از مسیریاب متفاوت است زیرا از آدرس‌های فیزیکی MAC در لایه‌ی پیوند جهت انتقال قاب‌های داده بهره‌برداری می‌کند. در حالیکه مسیریاب یک راهیاب بسته‌ی لایه‌ی سوم است اما سوئیچ یک راهیاب بسته‌ی لایه‌ی دوم به‌شمار می‌آید.

اگرچه راهیاب‌ها و مسیریاب‌ها از لحاظ بنیادی متفاوت می‌باشند، اما مدیران شبکه در اغلب موارد مجبورند بین گزینش آن‌ها جهت نصب و ارتباط بین تجهیزات تصمیم‌گیری نمایند. به عنوان مثال در شبکه‌ی شکل ۵-۲۶ مدیر شبکه می‌تواند به‌آسانی از یک روتر به جای سوئیچ استفاده نماید و از طریق آن قطعه LAN های دپارتمان‌های مختلف نهاد، سرورها و مسیریاب دریچه اینترنت را به یکدیگر اتصال دهد. یقیناً یک مسیریاب امکان ارتباط بین بخش‌های متفاوت را بدون ایجاد برخورد اطلاعات با یکدیگر در وجود خود نهفته دارد. با توجه به این نکته که هر دو راهیاب‌ها و مسیریاب‌ها می‌توانند کاندیدهای مفید ارتباط بین تجهیزات باشند، آن‌گاه نقاط قوت و ضعف هریک از این دو رویکرد چه می‌باشد؟



شکل ۵-۲۹ ♦ پردازش بسته در راهیاب‌ها، مسیریاب‌ها و میزبان‌ها

1- Switch poisening

ابتدا نقاط قوت و ضعف راهیاب‌ها بررسی می‌شود. همان‌گونه که در بالا اشاره شده است کار با راهیاب بسیار ساده و از این لحاظ طرفداران پرشماری نیز دارد. راهیاب‌ها قادر به تأمین فیلترسازی و گسیل جلو با سرعت نسبتاً بالا می‌باشند. شکل ۵-۲۹ این موضوع را نشان می‌دهد که راهیاب‌ها بایستی پردازش قاب‌ها را حداکثر تا لایه‌ی دوم انجام دهند، در حالی‌که مسیریاب‌ها این پردازش را برای داده گرام‌ها و حداکثر تا لایه‌ی سوم انجام می‌دهند. به عبارتی جهت ممانعت از دوره (حلقوی) شدن قاب‌های فراپخش، توپولوژی فعال یک شبکه‌ی راهیابی شده محدود به یک درخت پوشا می‌شود زیرا در چنین درختی وقوع دوره یا حلقه امکان‌پذیر نمی‌باشد. از طرف دیگر یک شبکه‌ی راهیابی شده‌ی عظیم نیازمند جداول ARP پرحجم در گره‌ها نیز می‌باشد که نتیجه‌ی آن تولید ترافیک ARP قابل ملاحظه و پردازش وسیع می‌باشد. علاوه بر همه‌ی این موارد راهیاب‌ها هیچ گونه حفاظتی را در برابر طوفان‌های فراپخش تأمین نمی‌کنند واگر یک میزبان از مسیر عملکرد عادی خود خارج شود و یک جریان بی‌پایان از قاب‌های فراپخش اترنت را ایجاد نماید آن‌گاه راهیاب‌ها همه‌ی این قاب‌ها را گسیل می‌نمایند و سرانجام سبب فروپاشی سراسری و کامل شبکه می‌شوند.

اکنون نقاط قوت و ضعف مسیریاب‌ها را تحلیل می‌کنیم. از آن‌جائیکه آدرس‌دهی شبکه در اغلب موارد سلسله مراتبی است و مانند آدرس‌دهی MAC یکنواخت نمی‌باشد، بسته‌ها معمولاً دچار حلقه در درون مسیریاب‌ها نمی‌شوند حتی اگر شبکه دارای مسیرهای آزاد باشد. بدیهی است بسته‌ها می‌توانند دچار دوره و یا حلقه شوند چنانچه پیکربندی جداول روترها صحیح انجام نگیرد، اما براساس مطالب فصل ۴ پروتکل IP از یک دامنه‌ی سرپیام ویژه در داده گرام جهت محدودسازی تعداد دفعه‌های تکرار حلقه بهره می‌گیرد. بنابراین بسته‌ها محدود به یک درخت پوشا نمی‌شوند و نمی‌توانند از بهترین مسیر بین مبدأ و مقصد بهره گیرند. از آن‌جائیکه مسیریاب‌ها محدودیت در درخت پوشا ندارند، این امکان را نیز به شبکه‌ی جهانی اینترنت می‌دهند که نسبت به ساختن یک توپولوژی قوی و غنی که شامل پیوندهای فعال چندگانه به عنوان مثال، بین اروپا و آمریکای شمالی است، اقدام نماید. ویژگی دیگر مسیریاب‌ها در آن است که آن‌ها حفاظت به‌گونه‌ی دیواره‌ی آتش را در برابر طوفان‌های فراپخش لایه‌ی دوم فراهم می‌آورند. شاید یکی از چشم‌گیرترین نقاط ضعف مسیریاب‌ها این واقعیت است که آن‌ها بر خلاف راهیاب‌ها انعطاف‌پذیر و تا حدود زیادی دارای استقلال رای نمی‌باشند. این به مفهوم آن است که آن‌ها و میزبان‌های متصل به آن‌ها نیازمند پیکربندی آدرس‌های IP خود می‌باشند. هم‌چنین مسیریاب‌ها اغلب زمان پردازش طولانی‌تری را برای هر بسته در مقایسه با راهیاب‌ها نیازمند می‌باشند، زیرا آن‌ها بایستی حداکثر تا دامنه‌های لایه‌ی سوم را پردازش نمایند.

جدول ۵-۱ ♦ مقایسه‌ی ویژگی تجهیزات ارتباطی شبکه

	هاب (Hub)	مسیریاب‌ها (Routers)	راهیاب‌ها (Switches)
جداسازی ترافیک	خیر	آری	آری
استقلال و انعطاف‌پذیری در پیکربندی	آری	خیر	آری
مسیریابی بهینه	خیر	آری	خیر

با توجه به نقاط قوت وضعف راهیاب‌ها و مسیریاب‌ها که در جدول ۵-۱ خلاصه شده است. چه هنگام شبکه‌ی یک نهاد مانند یک مجتمع دانشگاهی و یا شرکت تجاری بزرگ بایستی از راهیاب‌ها بهره گیرد و چه هنگام از مسیریاب‌ها؟ معمولاً شبکه‌های کوچک شامل چند صد میزبان و تعداد اندکی

قطعه‌های LAN می‌باشند. برای چنین شبکه‌های کوچک که ترافیک ناشی از آن‌ها محلی است و نهفتگی انباشتی بالاتری را بدون نیاز به هیچ گونه پیکربندی آدرس IP دارند، بهره گیری از راهیاب‌ها کافی به نظر می‌رسد. از طرف دیگر شبکه‌های بزرگ‌تر شامل هزاران میزبان از جمله مسیریاب‌ها و نیز راهیاب‌ها در بین شبکه می‌باشند. در این حالت روترها کنترل بیش‌تری را بر روی طوفان‌های فراپخش، جداسازی به مراتب قوی‌تر ترافیک و کاربرد هوشمندانه‌تر از مسیرها در بین میزبان‌های شبکه را تأمین می‌کنند.

۵-۱۰ شبکه‌های محلی مجازی (VLANS)

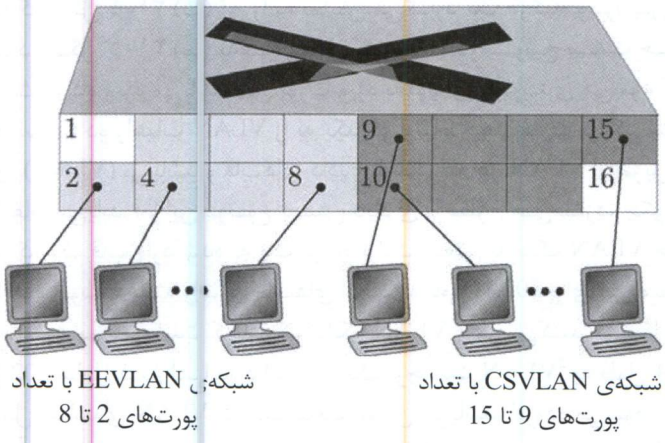
در بحث پیشین شکل ۵-۲۶ ملاحظه گردید شبکه‌های LAN متعلق به نهادها و سازمان‌ها اغلب به‌گونه‌ی سلسله مراتبی پیکربندی می‌شوند، که در آن هر گروه کاری مانند یک دپارتمان با شبکه‌ی LAN راهیابی شده‌ی ویژه‌ی خود متصل به شبکه‌های راهیابی شده‌ی دیگری است که یک راهیاب سلسله مراتبی دارد. اگرچه چنین پیکربندی در یک شرایط مطلوب به خوبی عمل می‌کند اما واقعیت آن است که محیط واقعی شبکه ایده‌آل نمی‌باشد. در پیکربندی شکل ۵-۲۶ سه نقطه ضعف را می‌توان مشاهده نمود:

- **فقدان جداسازی (انزوایی) ترافیک.** اگرچه رویکرد سلسله مراتبی ترافیک گروهی را در درون یک سوئیچ محلی نگهداری می‌کند اما ترافیک فراپخش مانند قاب‌های حامل پیام‌های ARP و DHCP و یا قاب‌هایی که مقصدآن‌ها از لحاظ آدرس از سوی یک راهیاب خودآموز تعریف نشده‌اند، هنوز هم بایستی تمامی شبکه‌ی نهاد را پویش نمایند. محدود نمودن دامنه‌ی چنین ترافیک فراپخش سبب بهینه‌سازی کارایی LAN می‌شود. شاید از همه مهم‌تر مطلوب آن باشد که ترافیک فراپخش LAN را به دلایل حفظ حریم خصوصی و یا امنیتی محدود نمود. به عنوان مثال اگر یک گروه شامل اطلاعات مربوط به تیم مدیران اجرایی رده بالای شرکت باشد و گروه دیگر شامل کارکنان رده‌های پایین‌تر که بسته‌های استشمام گر Wireshark را اجرا می‌کنند، آن‌گاه مدیر شبکه ممکن است ترجیح دهد که ترافیک متعلق به مدیران اجرایی حتی از نزدیکی میزبان‌های کارکنان شرکت هم عبور نکند. این گونه مجزاسازی و انزواگرایی را می توان از طریق جایگزینی راهیاب مرکزی شکل ۵-۲۶ با یک روتر انجام داد. در ادامه ملاحظه می‌شود که این جداسازی از طریق راهیابی لایه‌ی دوم نیز امکان‌پذیر است.
- **بهره‌گیری ناکارآمد از راهیاب‌ها.** اگر به جای سه گروه، نهادی شامل 10 گروه باشد، آن‌گاه تعداد 10 راهیاب سطح – نخست نیاز می‌باشد. اگر هر گروه کوچک باشند به عنوان مثال کم‌تر از 10 نفر، آن‌گاه یک سوئیچ تنها با تعداد 96 درگاه (پورت) به اندازه‌ی کافی بزرگ می‌باشد که هر کاربری را از طریق خویش پشتیبانی نماید. اما این راهیاب تک امکانات انزوای ترافیک را فراهم نمی‌کند.
- **مدیریت کاربران.** اگر یکی از کارکنان (کاربران) بین گروه‌ها در حرکت باشد، کابل‌بندی فیزیکی بایستی تغییر نماید تا قادر به اتصال آن کارمند به یک راهیاب دیگر در شکل ۵-۲۶ باشد. کارکنان متعلق به دو گروه این چالش را پیچیده‌تر و سخت‌تر می‌کنند.

سرانجام، هریک از این دشواری‌ها را می‌توان از طریق یک راهیاب که قادر به پشتیبانی **شبکه**

محلی مجازی (VLAN)[1] باشد، برطرف نمود. همان‌گونه که از عنوان این شبکه می‌توان استنباط کرد، یک راهیاب که پروتکل VLAN را پشتیبانی می‌کند این امکان را به وجود می‌آورد که چندین شبکه‌ی محلی مجازی را بتوان از طریق فقط یک زیرساخت فیزیکی شبکه‌ی محلی (LAN) تعریف نمود. میزبان‌های درون یک VLAN ارتباط خویش را با یکدیگر همانند آن که فقط آن‌ها و نه میزبان‌های دیگر متصل به سوئیچ می‌باشند، انجام می‌دهند. در شبکه‌ی VLAN برمبنای درگاه، پورت‌های راهیاب (واسط‌ها) از طریق مدیر شبکه به گروه‌ها تقسیم می‌شوند. هر گروه شامل یک VLAN با پورت‌هایی در هریک از شبکه‌های محلی مجازی است که تشکیل یک دامنه‌ی فراپخش را می‌دهند. به عبارت دیگر ترافیک فراپخش از یک درگاه فقط می‌تواند به دیگر پورت‌ها در گروه انتشار یابد.

شکل ۵-۳۰ یک راهیاب تک را نشان می‌دهد که دارای 16 پورت است. درگاه 2 به 8 متعلق به EEVLAN است در حالیکه پورت 9 به 15 متعلق به CSVLAN می‌باشد. بدیهی است در این شکل درگاه‌های 1 و 16 اختصاص نیافته‌اند. این شبکه‌ی VLAN همه‌ی دشواری‌هایی که در زیربخش‌های فصل و در بالا به آن اشاره رفته است را برطرف می‌کنند. قاب‌های داده‌ی متعلق به EEVLAN و CSVLAN کاملاً از یکدیگر در انزوا قرار می‌گیرند. در شکل ۵-۳۰ دو راهیاب جایگزین با فقط یک سوئیچ می شوند و اگر کاربری در درگاه شماره‌ی 8 راهیاب به دپارتمان CS متصل گردد، راهبر شبکه به سادگی نرم‌افزار VLAN را مجدداً به گونه‌ای پیکربندی می‌کند که درگاه 8 وابسته به CSVLAN گردد. به‌سادگی می‌توان تصور نمود چگونه راهیاب VLAN پیکربندی و عملیاتی می‌شود. مدیر شبکه یک درگاه را به یک VLAN اختصاص می‌دهد. پورت‌هایی که تاکنون اختصاص نیافته‌اند به یک VLAN پیش‌فرض اختصاص می‌یابند. اختصاص درگاه از طریق نرم‌افزار مدیر VLAN انجام می‌شود که عبارت است از یک جدول نگاشت درگاه به VLAN که در درون سوئیچ مورد بهره‌برداری قرار می‌گیرد و سخت‌افزار راهیاب فقط قاب‌ها را بین پورت‌های متعلق به همان VLAN جابه‌جا می‌کند.



شکل ۵-۳۰ ♦ یک راهیاب تک با دو VLAN پیکربندی شده

بدیهی است از طریق انزوای کامل دو VLAN یک اشکال عمده‌ی دیگر ظاهر میشود! چگونه ترافیک حاصل از بخش EE می‌تواند به دپارتمان CS گسیل گردد؟ یک روش انجام این مهم ارتباط یک درگاه راهیاب VLAN مانند پورت 1 در شکل ۵-۳۰ به یک مسیریاب خارجی می‌باشد، که

1- Virtual area networks

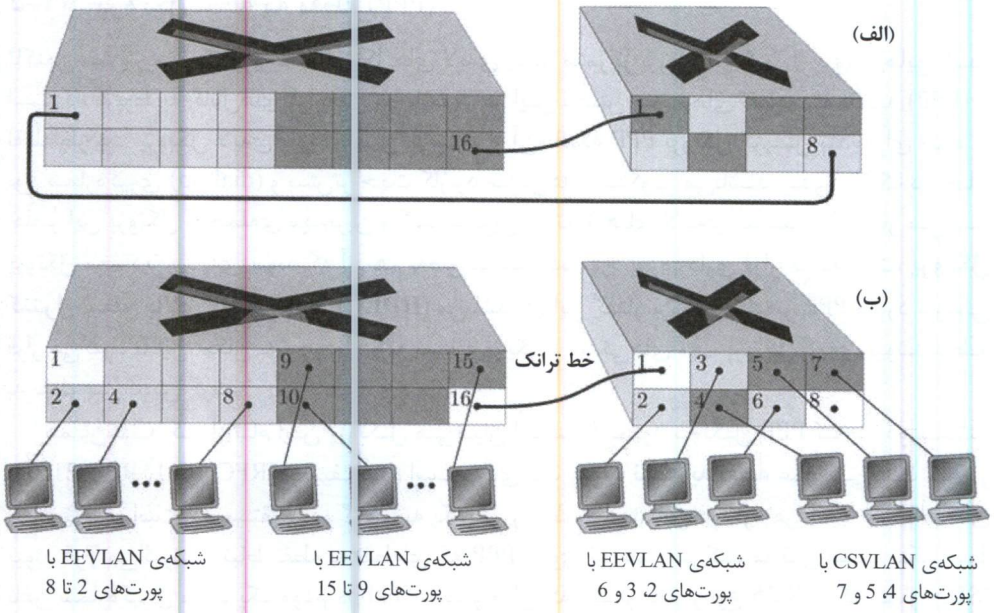
پیکربندی آن درگاه به‌گونه‌ای که تعلق به هر دو بخش EEVLAN و CSVLAN پیدا کند، صورت می‌پذیرد. در این حالت اگرچه دو بخش EE و CS از راهیاب فیزیکی مشابهی بهره می‌گیرند اما پیکربندی منطقی آن‌ها به گونه‌ای عمل می‌کند که به نظر می‌رسد هر دو دپارتمان EE به CS از طریق راهیاب‌های مجزا به مسیریاب متصل شده‌اند. یک داده گرام IP که از بخش EE به CS انتقال می‌یابد ابتدا از EEVLAN جهت رسیدن به مسیریاب عبور می‌کند و سپس گسیل جلو از طریق مسیریاب به سوی CSVLAN می‌شود و سرانجام به میزبان CS وارد می‌گردد. تولیدکنندگان راهیاب انجام چنین پیکربندی‌هایی را برای مدیر شبکه از طریق طراحی تجهیزات درون سوئیچ که هر دو نقش راهیاب VLAN و یک روتر را بازی می‌کنند، آسان نموده‌اند. بنابراین ملاحظه می‌شود نیازی به یک مسیریاب مجزا و خارجی نیز نمی‌باشد.

با بازگشت به شکل ۵-۲۶ فرض کنید به جای داشتن دو بخش مجزای EE و CS برخی از کاربران هر دو دپارتمان در یک مکان مجزا که نیازمند دستیابی به شبکه می‌باشند، ساکن شده‌اند و بدیهی است که آن‌ها مایل هستند که بخشی از شبکه VLAN دپارتمان خود نیز باشند. شکل ۵-۳۱ یک راهیاب 8 پورت دیگر را نشان می‌دهد که درگاه‌های سوئیچی تعریف شده‌اند که بنابر نیاز متعلق به شبکه‌ی مجازی EE و یا CS باشند. آیا این دو راهیاب بایستی به یکدیگر متصل شوند؟ یک راهکار ساده آن است که یک درگاه را اختصاص به CSVLAN برروی هر راهیاب نموده و مشابه آن برای EEVLAN را نیز تعریف نموده و این پورت‌ها را به یکدیگر ارتباط داده، براساس آن‌چه شکل ۵-۳۱ (الف) نشان داده است، این راهکار مقیاس‌پذیر نمی‌باشد زیرا به تعداد N شبکه مجازی (VLAN) برای تعدادN درگاه برروی هر راهیاب نیاز می‌باشد تا بتوان دو سوئیچ را به راحتی به یکدیگر ارتباط داد.

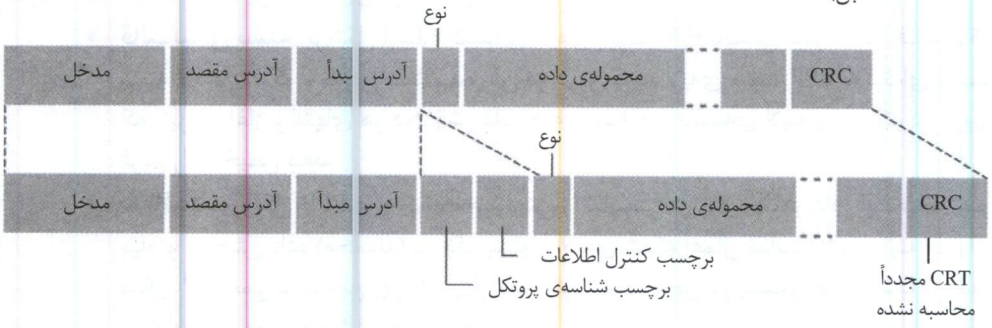
یک رویکرد مقیاس‌پذیر دیگر، ایجاد ارتباط بین راهیاب‌های VLAN می‌باشد که به این ارتباط اصطلاحاً **رابط ترانک شبکه‌ی مجازی**[1] **(VLAN-T)** گفته می‌شود. در رویکرد شبکه‌ی مجازی رابط ترانک (مادرسیم) که شکل ۵-۳۱ (ب) آن را به نمایش می‌گذارد یک درگاه ویژه برروی هر راهیاب اختصاص می‌یابد. در شکل ۵-۳۱ (ب) به‌عنوان مثال پورت 16 در سوئیچ سمت چپ و درگاه 1 در راهیاب سمت راست درگاه ویژه می‌باشد. این پورت ویژه به‌گونه‌ای پیکربندی می‌شود که به عنوان یک درگاه ترانک (مادر سیم) دو راهیاب VLAN را به یکدیگر ارتباط دهد. پورت ترانک متعلق به همه‌ی شبکه‌های مجازی (VLAN) می‌باشد و قاب‌های داده‌ی ارسالی به هر VLAN از طریق پیوند ترانک به دیگر راهیاب‌ها انتقال می‌یابد. اما این موضوع پرسش دیگری را مطرح می‌سازد، چگونه یک راهیاب تشخیص می‌دهد که یک قاب وارد شده به یک درگاه ترانک متعلق به یک VLAN خاص می‌باشد؟ نهاد IEEE یک الگوی توسعه یافته را برای قاب‌های اترنت به نام 802.1Q برای این منظور تعریف نموده است. این قالب جهت فریم‌هایی است که از یک ترانک VLAN عبور می‌کنند. از شکل ۵-۳۲ مشاهده می‌شود قاب 802.1Q شامل فریم استاندارد اترنت با یک **برچست VLAN**[2] به طول 4 بایت می باشد که به سرپیام حامل شناسه‌ی VLAN که قاب متعلق به آن می‌باشد، افزوده می‌شود. برچسب VLAN به یک قاب از سوی راهیاب در سمت ارسال یک ترانک VLAN افزوده می‌شود. پس از آن این برچسب تجزیه و سپس از سوی سوئیچ در سمت دریافت ترانک حذف می‌گردد. برچسب VLAN خود شامل یک دامنه‌ی **شناسه‌گر برچسب پروتکل**[3] **(TPID)** به طول 2 بایت با یک ارزش ثابت هگزا برابر

1- VLAN Trunking
2- VLAN Tag
3- Tag protocol identifier

(81-00)، یک دامنه‌ی **برچسب کنترل اطلاعات (TCI)** به طول 2 بایت که شامل یک دامنه‌ی شناسه‌گر VLAN به طول 12 بیت، و یک دامنه‌ی **اولویت** به طول 3 بیت می‌باشد که از لحاظ کاربرد مشابه دامنه‌ی TOS در داده گرام IP است.



شکل ۵-۳۱ ♦ اتصال دو راهیاب شبکه‌ی مدار مجازی با دو VLAN: (الف) دو کابل (ب) رابط ترانک (مادر کابل)



شکل ۵-۳۲ ♦ فریم اصلی اترنت در بالا، قاب VLAN برمبنای 802.1q در پائین

در این زیربخش به صورت خلاصه شبکه‌های مجازی محلی (VLAN) تحلیل شدند و تمرکز اصلی ما برروی VLAN برمبنای درگاه (پورت) قرار گرفت. بدیهی است شبکه‌های VLAN از چندین روش دیگر نیز قابل تعریف و پیاده‌سازی می‌باشند. در شبکه‌های VLAN برمبنای MAC، مدیر شبکه مجموعه‌ای از آدرس‌های MAC که متعلق به هریک از VLANها می‌باشد را مشخص می‌کند. هرگاه وسیله‌ای به یک درگاه می‌پیوندد، آن پورت به یک شبکه‌ی VLAN مناسب بر مبنای آدرس MAC آن تجهیزات متصل می‌شود. شبکه‌های مجازی محلی (VLAN) را نیز می‌توان برمبنای پروتکل‌های لایه‌ی

شبکه مانند: IPV4، IPV6 و یا Apple Talk و دیگران تعریف نمود. برای ملاحظه‌ی جزئیات بیش‌تر به مجموعه استاندارد [IEEE 802.1q] مراجعه شود.

۵-۱۱ پروتکل نقطه به نقطه (PPP)

تاکنون بیش‌تر مباحث مربوط به پروتکل‌های لایه‌ی پیوند متمرکز برروی آن گروه از پروتکل‌هایی شده است که مربوط به کانال‌های فراپخش می‌باشند. در این بخش پیوندهای نقطه به نقطه (PPP) از نقطه‌نظر یک پروتکل لایه‌ی پیوند تحلیل می‌شود. از آن‌جائیکه PPP پروتکل گزینش شده برای پیوند از نوع شماره‌گیری (dial up) و بیش‌تر جهت کاربرد میزبان‌های مسکونی می‌باشد، بدون شک در حال حاضر این پروتکل از جمله‌ی مهم‌ترین و گسترده‌ترین پروتکل‌های لایه‌ی پیوند به شمار می‌آید. پروتکل مهم دیگر لایه‌ی پیوند که آن هم به‌صورت گسترده مورد بهره‌برداری قرار گرفته است، **پروتکل کنترل سطح بالای پیوند داده[1] (HDLC)** می‌باشد. در این گفتار پروتکل ساده‌تر PPP مورد بررسی قرار می‌گیرد تا این امکان به وجود آید که بسیاری دیگر از ویژگی‌های مهم پروتکل لایه‌ی پیوند نقطه به نقطه نیز کاوش گردد.

همان‌گونه که از نام این پروتکل می‌توان استنباط نمود پروتکل PPP که در دو مستند [RFC 1661, RFC 2153] توصیف شده است برروی یک **پیوند نقطه به نقطه** عمل می‌کند که در واقع پیوندی است که مستقیماً دو گره را به یکدیگر ارتباط می‌دهد. این دو گره هریک در یک انتهای پیوند قرار می‌گیرند. ارتباط نقطه به نقطه‌ای که PPP برروی آن عمل می‌کند ممکن است یک خط تلفن شماره‌گیری مانند یک مودم 56kbps باشد و یا آن که یک ارتباط از نوع X.25، یک مدار ISDN و یا یک پیوند SONET/SDH که در فصل نخست به همه‌ی آن‌ها اشاره رفت. قبل از بررسی جزئیات PPP لازم است نیازهای بنیادی که نهاد IETF جهت طراحی PPP برمبنای مستند [RFC1546] تعریف نموده است، تجزیه و تحلیل گردد. این موارد عبارتند از:

- **قالب‌بندی بسته.** پروتکل ارسال کننده‌ی لایه‌ی پیوند PPP بایستی قادر به دریافت یک بسته لایه‌ی شبکه و سپس بسته‌بندی آن در درون فریم لایه‌ی پیوند PPP به‌گونه‌ای باشد که گیرنده آغاز و انتهای هر دو بخش قاب لایه‌ی پیوند و بسته‌ی لایه‌ی شبکه در درون فریم را تشخیص دهد.
- **شفافیت.** پروتکل PPP هیچ گونه محدودیتی را نبایستی بر روی ظاهر داده از لحاظ سر پیام و یا بخش داده (محموله) در یک بسته‌ی لایه‌ی شبکه اعمال نماید. بنابراین به عنوان مثال، PPP نمی‌تواند مانع بهره‌گیری از یک الگوی بیت خاص در بسته‌ی لایه‌ی شبکه باشد. در ادامه این موضوع بیش‌تر مورد توجه قرار می‌گیرد.
- **پروتکل‌های لایه‌ی شبکه‌ی چندگانه.** پروتکل PPP بایستی قادر به پشتیبانی پروتکل‌های لایه‌ای شبکه‌ی چندگانه باشد. به عنوان مثال پروتکل‌های IP و DECnet که همزمان برروی پیوند فیزیکی مشابه اجرا می‌شوند. این موضوع مشابه پروتکل IP می‌باشد که لازم است برروی پروتکل‌های گوناگون و چندگانه‌ی لایه‌ی انتقال مانند TCP و یا UDP از طریق یک ارتباط انتها به انتها عمل نماید. به عبارت دیگر PPP بایستی قادر به تسهیم پروتکل‌های مختلف لایه‌ی شبکه برروی یک ارتباط نقطه به نقطه باشد. این نیاز به مفهوم

1- High level data link control protocol

آن است که حداقل PPP نیازمند یک دامنه‌ی نوع پروتکل و یا شبیه آن می باشد که PPP سمت دریافت قادر به تسهیم یک قاب دریافتی تا پروتکل مناسب در لایه‌ی شبکه باشد.

- **پیوندهای چندگانه متنوع.** علاوه بر توان‌مندی حمل پروتکل‌های سطح بالاتر چندگانه، پروتکل PPP بایستی قادر به انجام عملیات برروی انواع گسترده‌ای از پیوندهای شامل آن گروه از ارتباط‌ها که به گونه‌ی سریال یا به بیان دیگر یک بیت در هر زمان در یک جهت مستقیم عمل می‌کنند و یا موازی به مفهوم انتشار چندین بیت به صورت موازی و نیز اصطلاحاً **همزمان**[1] در قالب ارسال بیت‌های داده به همراه یک سیگنال ساعت و یا **غیرهمزمان**[2]، کند سرعت، پرسرعت، الکتریکی و یا نوری باشد.
- **آشکارسازی خطا.** یک گیرنده‌ی ppp بایستی قادر به تشخیص بیت‌های خطا در قاب‌های دریافتی باشد.
- **حیات ارتباط.** پروتکل PPP بایستی قادر به تشخیص عیب در سطح لایه‌ی پیوند به گونه‌ای باشد که در صورت عدم توانایی در انتقال داده از سوی پیوند نقطه‌ی ارسال پیوند به بخش دریافت آن وضعیت خطا را به لایه‌ی شبکه ارسال نماید.
- **تعامل‌گرایی در آدرس لایه‌ی شبکه.** پروتکل PPP نیازمند تأمین روشی جهت لایه‌های شبکه است که در تعامل با یکدیگر مانند IP می‌باشند این تعامل به منظور یادگیری و یا پیکربندی آدرس لایه‌ی شبکه یکدیگر می‌باشد.
- **سادگی.** علاوه بر تأمین نیازهای توصیف شده در بالا، PPP بایستی از طراحی ساده‌ای برخوردار باشد، برمبنای مستند [RFC1547] که در جمله‌ای خلاصه می‌گوید «آن‌چه از همه مهم‌تر است سادگی یک پروتکل نقطه به نقطه می‌باشد.» با توجه به موارد بالا رعایت این نیاز آخر اهمیت چشمگیر خود را نشان می‌دهد. در حال حاضر بیش از یکصد مستند RFC در تعریف سادگی PPPپیشنهاد شده است.

اگرچه به نظر می‌رسد بسیاری از این نیازها بر طراحی PPP متمرکز شده است اما وضعیت می‌تواند دشوارتر نیز باشد! مشخصات طراحی PPP به‌گونه‌ی آشکاری شامل عملکرد پروتکلی که PPP الزام به پیاده‌سازی آن‌ها ندارد، نیز می‌شود. این موارد عملکردی پروتکل عبارتند از:

- **تصحیح خطا.** پروتکل PPP تنها بایستی خطای بیت را تشخیص دهد و لازم نیست آن‌ها را تصحیح نماید.
- **کنترل جریان.** از یک گیرنده‌ی PPP انتظار می‌رود که قادر به دریافت فریم‌ها در سرعت کامل لایه‌ی فیزیکی باشد. اگر یک لایه‌ی بالاتر قادر به دریافت بسته‌ها در سرعت کامل نباشد، این برعهده‌ی لایه‌ی بالاتر است که بسته‌ها را حذف و فرستنده را در لایه‌ی بالاتر مجبور به کنترل سرعت نماید. به عبارت دیگر به جای الزام کنترل فرستنده‌ی PPP به کاهش سرعت انتقال، این مسئولیت برعهده‌ی پروتکل سطح بالاتر گذارده می‌شود که سرعت تحویل بسته‌ها به PPP را جهت ارسال کنترل نماید.
- **نوبت‌بندی و ترتیب.** پروتکل PPP الزامی به تحویل قاب‌های داده به پیوند دریافت کننده برمبنای ترتیب ونظمی که آن‌ها از سوی پیوند فرستنده گسیل شده‌اند، ندارد. جالب توجه

1- Synchronous
2- Asynchronous

است اگرچه این انعطاف‌پذیری در عدم تحویل منظم سازگاری با الگوی سرویس IP دارد که اجازه می‌دهد بسته‌های IP به هر ترتیب و قاعده‌ای تحویل انتها به انتها شوند، اما دیگر پروتکل‌های لایه‌ی شبکه که برروی PPP عمل می‌کنند ملزم به تحویل انتها به انتهای بسته‌ها به گونه‌ی منظم و پی‌درپی می‌باشند.

- **پیوندهای چند نقطه.** پروتکل PPP فقط نیازمند عملیات برروی پیوندهایی می‌باشد که فقط یک فرستنده و یک گیرنده دارند. بقیه‌ی پروتکل‌های لایه‌ی پیوند مانند HDLC می‌توانند گیرنده‌های چندگانه (چندتایی) را روی یک پیوند پشتیبانی نمایند. به عبارت دیگر مانند اترنت.

اکنون که اهداف طراحی پروتکل PPP بررسی گردید از لحاظ طراحی PPP جهت دستیابی به این اهداف به ادامه‌ی مطالب مربوط به این زیربخش می‌پردازیم.

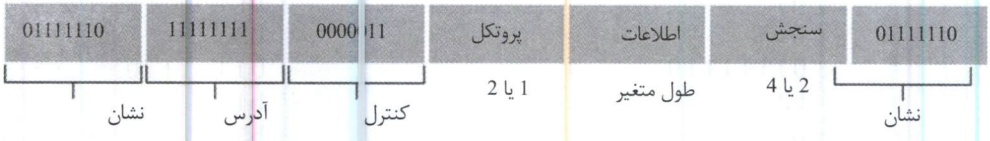
۱- تشکیل قاب داده‌ی PPP

شکل ۵-۳۳ فریم داده‌ی PPP را نشان می‌دهد که از الگوی قاب HDLC توصیف شده در مستند [RFC1662] بهره می‌گیرد. قاب PPP شامل دامنه‌های ذیل است:

- **دامنه‌ی نشان.** هر قاب PPP با یک دامنه‌ی نشان به طول یک بایت و یا یک ارزش ثابت هگزا معادل 7E (دودویی: 1110 0111) آغاز و پایان می‌گیرد.
- **دامنه‌ی آدرس.** تنها ارزش ممکن برای این دامنه برابر هگزا FF یا دودویی 11111111 می‌باشد.
- **دامنه‌ی کنترل.** تنها ارزش ممکن برای این دامنه برابر هگزا 03 یا دودویی 00000011 می‌باشد. زیرا هر دو دامنه‌ی آدرس و کنترل فقط یک ارزش ثابت را می‌توانند اختیار نمایند. ممکن است تعجب کنید چرا این دو دامنه موردنیاز می‌باشند. مشخصات PPP در مستند [RFC1662] عنوان می‌نماید. "دیگر ارزش‌ها می توانند در آینده تعریف شوند"، اگرچه هیچ یک از آن‌ها تا به امروز تعریف نشده‌اند. از آن‌جائیکه این دامنه‌ها ارزش‌های ثابت اختیار می‌کنند، پروتکل PPP این امکان را به فرستنده می‌دهد که بایت‌های کنترل و آدرس را گسیل نکنند، در نتیجه 2 بایت سرآیند در قاب PPP صرفه‌جویی می‌شود.
- **پروتکل.** دامنه‌ی پروتکل به گیرنده‌ی PPP می‌گوید داده‌ی بسته‌بندی شده‌ی دریافتی که محتویات محموله‌ی داده در فریم PPP است متعلق به کدام یک از پروتکل‌های لایه‌ی بالاتر است. هنگام دریافت یک قاب PPP، گیرنده‌ی آن فریم اقدام به سنجش صحت اطلاعات درون آن می‌کند و سپس قاب داده را به سمت پروتکل مناسب آن هدایت می‌کند. دو مستند[RFC 1700, RFC3232] یک رمز پروتکل 16 بیت را جهت بهره‌برداری PPP تعریف می‌کنند. آن‌چه برای ما اهمیت دارد پروتکل IP است که داده‌ی بسته‌بندی شده در قاب PPP می‌باشد. رمز پروتکل IP نیز یک ارزش هگزا معادل 21 یا دودویی 00100001 می باشد. کدهای پروتکل‌های دیگر لایه‌ی شبکه به ترتیب برای Apple Talk برابر (29) و DECnet برابر (27) هگزا می‌باشند.
- **اطلاعات.** این دامنه شامل بسته یا داده‌ی کپسول‌بندی شده است که به سمت یک پروتکل لایه‌ی بالاتر مانند IP از طریق پیوند PPP گسیل می‌شود. حداکثر طول پیش‌فرض

دامنه‌ی اطلاعات برابر 1500 بایت است اگرچه این طول را می‌توان هنگامی که پیوند برای نخستین بار پیکربندی می‌شود نیز تغییر داد.

- **کد مجموع مقابله (Cheksum)**. دامنه‌ی کد مجموع مقابله‌ای که جهت آشکارسازی خطای بیت در یک قاب گسیل شده مورد بهره‌برداری قرار می‌گیرد یک رمز افزونه‌ی دوره‌ای استاندارد 2 و یا 4 بایت HDLC می‌باشد.



شکل ۵-۳۳ ♦ الگوی قاب داده‌ی PPP

۲- تزریق بایت

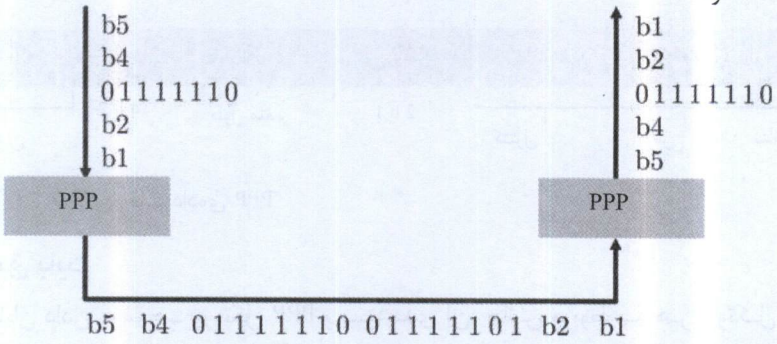
قبل از پایان دادن به مبحث پروتکل PPP و بسته‌بندی آن، چالش مربوط به هر پروتکل که از یک الگوی بیت ویژه در دامنه‌ی نشان خود جهت مشخص نمودن آغاز و پایان یک فریم استفاده می‌کند، تحلیل می‌گردد. این اشکال هنگامی ظهور می‌کند که الگوی نشان در نقاط دیگر بسته مانند محموله‌ی داده‌ی اصلی تکرار شود. به عنوان مثال چه اتفاقی روی می‌دهد اگر نشان 01111110 در ادامه‌ی اطلاعات ظاهر شود؟ آیا گیرنده در انتهای فریم PPP دچار خطا می‌شود؟

یکی از راهکارهای جلوگیری از چنین رویدادی آن است که PPP از گسیل داده‌ی حامل الگوی بیت در دامنه‌ی نشان از سوی پروتکل لایه‌ی بالاتر جلوگیری نماید. ویژگی شفافیت PPP که قبلاً به آن اشاره گردید رفع این نیاز را می‌کند. راهکار جایگزین که در حقیقت PPP و دیگر پروتکل‌ها از آن بهره می‌گیرند معروف به **تزریق بایت**[1] می‌باشد.

پروتکل PPP به تعریف یک **بایت کنترل گریز**[2] به‌گونه‌ی دودویی: 01111101 می‌پردازد. اگر ترتیب بیت‌های نشان که به صورت 01111110 می‌باشد در هر نقطه‌ای از قاب به غیر از نشان تکرار گردد. پروتکل PPP مقدم بر آن بیت‌ها الگوی بایت گریز را قرار می‌دهد. به عبارت دیگر بایت گریز را در جریان داده‌ی در حال انتقال **تزریق** می‌کند و این عمل وارد نمودن بایت گریز را قبل از بایت نشان یعنی 01111110 انجام می‌دهد که برای گیرنده به مفهوم آن باشد که الگوی 01111110 که در پی این بایت گریز آمده است نشان پایان قاب داده نمی‌باشد بلکه داده‌ی واقعی است. گیرنده‌ای که الگوی 01111101 و در پی آن 01111110 را مشاهده نماید ابتدا اقدام به حذف بایت کنترل گریز می‌کند و سپس اطلاعات داده‌ی اصلی را بازیابی می‌نماید. ممکن است این پرسش مطرح شود که اگر بایت کنترل گریز خودبخشی از اطلاعات داده‌ی اصلی باشد آن‌گاه چه اتفاقی روی می‌دهد؟ پاسخ آن است که قبل از این الگو نیز بایستی بایت کنترل گریز تزریق شود. در این حالت هنگامی که گیرنده یک بایت کنترل گریز را مشاهده می‌کند می‌داند که آن بایت به جریان داده تزریق شده است اما هنگامی که دو بایت مشابه بایت کنترل گریز را می‌بیند، متوجه می‌شود یک بایت از آن تزریق و بایت دیگر داده‌ی واقعی در جریان اطلاعات قاب می‌باشد. شکل ۵-۳۴ تزریق بایت PPP را نشان می‌دهد. آن‌چه

1- Byte stuffing
2- Escape control byte

درحقیقت از این شکل می توان استنباط نمود آن است که پروتکل نقطه به نقطه (PPP) بایت داده‌ی گریزان در جریان اطلاعات واقعی قاب را با معادل 20 هگزا XOR می‌کند. یادآوری این نکته ضروری است که PPP دارای یک **پروتکل کنترل پیوند (LCP)**[1] نیز می‌باشد که وظیفه‌ی آن مقداردهی، نگهداری و قطع یک پیوند PPP می‌باشد. پروتکل LCP را می توان با مراجعه به سامانه‌های متعدد اینترنت مطالعه نمود.



شکل ۵-۳۴ ♦ تزریق بایت

۵-۱۲ مجازی سازی پیوند: عملکرد یک شبکه در قالب لایه‌ی پیوند

از آن‌جائیکه تمرکز این فصل پروتکل‌های لایه‌ی پیوند است و با توجه به بررسی همه جانبه‌ی این پروتکل‌ها، جای دارد برروی آن‌چه از واژه‌ی **پیوند** در این فصل آموخته‌ایم تأملی داشته باشیم. مطلب با بازنگری پیوند به عنوان یک رشته‌ی فیزیکی آغاز شد که دو میزبان را به یکدیگر ارتباط می‌دهد. این مهم در شکل ۵-۲ نشان داده شده است در مطالعه‌ی پروتکل‌های دستیابی چندگانه‌ی شکل ۵-۹ ملاحظه گردید، میزبان‌های چندگانه می‌تواند از طریق یک کابل (رشته‌ی) مشترک به یکدیگر متصل شده و آن کابل اتصال دهنده می‌تواند از نوع طیف رادیویی و دیگر رسانه‌ها باشد. این موضوع سبب گردید یک پیوند را در قالب دقیق‌تر آن یعنی یک کانال به جای سیم تحلیل نمائیم. در مطالعه‌ای که پیرامون اترنت LAN برمبنای دو شکل ۵-۲۶ و ۵-۳۱ انجام گرفت، ملاحظه شد که رسانه‌ی ارتباطی بین دو یا چند میزبان می‌تواند زیرساخت راهیابی شده‌ی نسبتاً پیچیده را داشته باشد. در سراسر این بررسی تکاملی میزبان‌ها از طرف دیگر دارای این دیدگاه بودند که رسانه‌ی ارتباطی به شکل ساده یک کانال لایه‌ی پیوند است که دو یا چند میزبان را به یکدیگر ارتباط می‌دهد. به عنوان مثال ملاحظه شد یک میزبان اترنت می‌تواند از عدم آگاهی خود که میزبان‌های LAN دیگری از طریق یک قطعه‌ی کوتاه LAN مانند شکل ۵-۹ و یا به وسیله‌ی شبکه‌ی LAN راهیابی شده‌ای که در یک وسعت جغرافیایی پراکنده قرار دارند مانند شکل ۵-۲۶ و یا سرانجام از طریق یک VLAN مانند شکل ۵-۳۱ متصل شده است، آسوده خاطر باشند.

در بخش ۵-۱۱ ملاحظه شد از پروتکل PPP در اغلب موارد جهت ارتباط از طریق مودم بین دو میزبان استفاده می‌شود. در این مقطع پیوند ارتباط دهنده‌ی دو میزبان در واقع همان شبکه‌ی تلفنی است که عبارت است از یک شبکه‌ی ارتباط جهانی با مجموعه‌ای از راهیاب‌های ویژه‌ی خود، پیوندها و پشته‌های پروتکلی جهت انتقال داده و انجام عملیات سیگنالینگ، در عین حال این شبکه از لحاظ

1- Link control protocol

منطقی نیز مجزا می‌باشد. از طرف دیگر از نقطه نظر لایه‌ی پیوند اینترنت یک ارتباط از نوع اصطلاحاً شماره‌گیری از درون شبکه‌ی تلفنی به صورت یک رشته سیم ساده قلمداد می‌شود. به بیان دیگر اینترنت شبکه‌ی تلفنی را **مجازی‌سازی** می‌کند و تمامیت آن را به عنوان یک فناوری لایه‌ی پیوندکه ارتباط بین دو میزبان اینترنت را برقرار می‌سازد، مشاهده می‌کند.

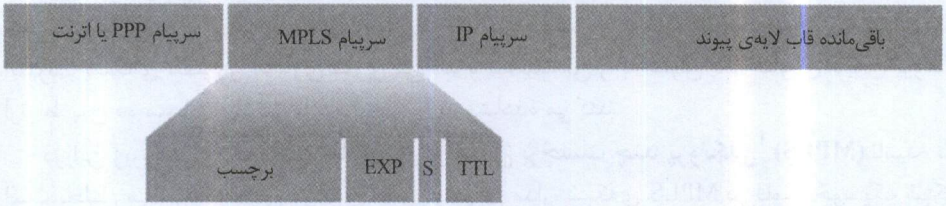
در این زیربخش آن‌چه اصطلاحاً شبکه‌ی **راهیابی برچسب چند پروتکلی**[1] **(MPLS)** نامیده شده است تحلیل می‌شود. برعکس شبکه‌ی تلفنی راهیابی مدار، شبکه‌ی MPLS در قامت خود یک شبکه‌ی راهیابی بسته‌ی مدار مجازی (VCN) می‌باشد. این شبکه قالب بسته‌های ویژه‌ی خود را دارد و از لحاظ گسیل آن‌ها نیز دارای روش خاص خود می‌باشد. بنابراین از نقطه نظر آموزش آکادمیک بحث پیرامون MPLS می‌تواند در چارچوب مطالعه‌ی لایه‌ی شبکه و یا لایه‌ی پیوند قرار گیرد. از نقطه نظر اینترنت MPLS را می‌توان شبیه شبکه‌ی تلفنی به شمار آورد و اترنت راهیابی شده را نیز به عنوان فناوری لایه‌ی پیوند که ارتباط تجهیزات IP را با یکدیگر پشتیبانی می‌کند. بنابراین MPLS را در بحث خود به عنوان لایه‌ی پیوند تحلیل می‌کنیم. شبکه‌های **تکرار قاب**[2] **(FR)** و ATM را نیز می‌توان جهت اتصال بین تجهیزات IP به کار گرفت. اگرچه آن‌ها فناوری قدیمی‌ترند اما به‌کار گرفته شده می‌باشند.در این زیربخش تمرکز ما بیش‌تر برروی چگونگی ارتباط سرورهای MPLS با تجهیزات IP است، اگرچه فناوری‌های مرتبط با آن‌ها را نیز تحلیل خواهیم نمود.

• راهیابی برچسب چند پروتکلی (MPLS)

تکامل MPLS حاصل تلاش‌های متعدد صنعتی از اواسط تا اواخر دهه‌ی 1990 میلادی جهت بهینه‌سازی سرعت گسیل مسیریاب‌های IP از طریق گزینش یک اصل کلیدی در شبکه‌های مدار مجازی بود. این اصل کلیدی یک **برچسب طول ثابت** است. هدف، کنار گذاردن زیرساخت گسیل جلو با بهره‌گیری از آدرس مقصد داده گرام IP نمی‌باشد که مبنای آن برچسب‌های طول ثابت و مدارهای مجازی است، بلکه افزودن به آن با امکان گزینش اختیاری برچسب، گذاری داده گرام‌ها و قادر نمودن روترها در گسیل این داده گرام‌ها براساس برچسب‌های طول ثابت به جای بهره‌گیری از آدرس‌های IP مقصد. از همه مهم‌تر این روش‌ها دوش به دوش IP از آدرس‌دهی و مسیریابی IP بهره می‌گیرند. سازمان IETE به این تلاش‌ها در قالب پروتکل MPLS در مستند [RFC3031, RFC3032] وحدت بخشید و نتیجه‌ی آن آمیزش روش‌های VC با شبکه‌های داده گرام مسیریابی شده می‌باشد.

مطالعه‌ی MPLS را با درنظر گرفتن الگوی یک قاب لایه‌ی پیوند که از طریق یک مسیریاب مجهز به MPLS قابل بهره‌برداری است، آغاز می‌کنیم. شکل ۵-۳۵ نشان می‌دهد که یک قاب لایه‌ی پیوند که روی یک پیوند PPP و یا LAN مانند اترنت انتقال می‌یابد، شامل یک سرپیام کوچک MPLS می‌شود که مابین سرپیام لایه‌ی دوم مثل PPP یا اترنت و سرپیام لایه‌ی سوم مانند IP اضافه می‌گردد. مستند [RFC3032] الگوی سرپیام MPLS را برای چنین پیوندهایی تعریف می‌کند. این سرپیام‌ها برای شبکه‌های ATM، تکرار قاب (FR) و دیگر RFCها نیز تعریف می‌شود. در بین دامنه‌های سرپیام MPLS برچسب از همه بیش‌تر خودنمایی می‌کند زیرا نقش شناسه‌ی مدار مجازی در بخش ۴-۳ را ایفا می‌کند. تعدادی برابر 3 بیت، ویژه‌ی کاربردهای آزمایشی تعریف شده‌اند. از یک بیت S جهت تعریف انتهای مجموعه‌ای از سرپیام‌های پشته شده‌ی MPLS و یک دامنه‌ی TTL استفاده می‌شود.

1- Multiprotocol label switching
2- Framerelay



شکل ۵-۳۵ ♦ سرپیام MPLS، مکان آن در بین سرپیام‌های لایه‌های پیوند و شبکه

از شکل ۵-۳۵ می‌توان ملاحظه نمود که یک قاب گسترش یافته‌ی MPLS فقط می‌تواند بین مسیریاب‌هایی جابه جا شود که هر دوی آن‌ها مجهز به MPLS باشند. زیرا روترهای غیر MPLS هنگامی که سرپیام MPLS را ملاحظه کنند دچار سردرگمی می‌شوند زیرا این گونه مسیریاب‌ها به دنبال سرپیام IP می‌باشند. معمولاً یک مسیریاب مجهز به امکانات MPLS را اصطلاحاً **مسیریاب راهیابی شده‌ی برچسب[1] (LSR)** می‌نامند. زیرا این گونه مسیریاب‌ها، قاب‌های داده‌ی MPLS را از طریق جستجوی برچسب MPLS آن‌ها که در جدول گسیل جلو ذخیره شده است، انتقال می‌دهند. پس از یافتن برچسب MPLS از جدول، بی‌درنگ داده گرام به واسط خروجی مناسب تحویل می‌گردد. در نتیجه یک مسیریاب LSR نیازمند جداسازی آدرس IP مقصد و عملیات جستجوی نگاشت طولانی‌ترین پیشوند در جدول گسیل جلو که در فصل‌های ۳ و ۴ به آن اشاره گردید، نمی‌باشد. چگونه یک مسیریاب LSR از این نکته آگاهی می‌یابد که روتر مجاور آن نیز از نوع LSR می‌باشد؟ علاوه برآن چگونه مسیریاب می‌داند که کدام برچسب را بایستی به یک آدرس IP مقصد داده شده وابسته نماید؟ پاسخ این دو پرسش نیازمند چگونگی تعامل بین یک گروه از مسیریاب‌های MPLS می‌باشد.

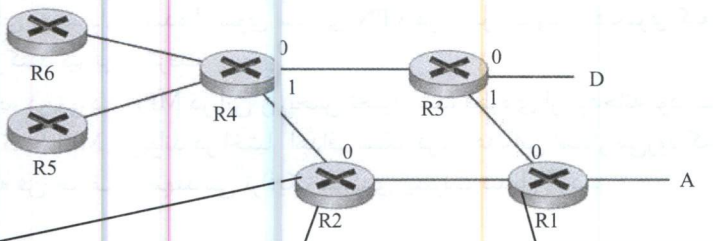
در مثال شکل ۵-۳۶ روترهای به ترتیب R1 تا R4 همگی از نوع LSR یعنی مجهز به امکانات MPLS می‌باشند. مسیریاب‌های R5 و R6 از نوع استاندارد IP درنظر گرفته شده‌اند. فرض کنید R1 به دو مسیریاب R2 و R3 آگهی نموده است که قادر است مسیریابی بسته‌ها را به مقصد A انجام دهد و یک قاب MPLS دریافتی با برچسب 6 را به آن مقصد یعنی A گسیل خواهد نمود. مسیریاب R3 نیز به R4 آگهی داده است که قادر است بسته‌ها را به مقصدهای A و D مسیریابی نماید و علاوه برآن قاب‌های ورودی MPLS با برچسب‌های به ترتیب 10 و 12 را نیز به مقصدهای A و D راهیابی خواهد نمود. روتر R2 نیز به مسیریاب R4 آگهی می‌نماید که امکان دسترسی به مقصد A را دارد و یک قاب MPLS دریافتی با برچسب 8 را به سمت مقصد A گسیل می‌نماید. نکته‌ی قابل توجه آن است که مسیریاب R4 اکنون در موقعیت ویژه‌ای قرار می‌گیرد و آن در اختیار داشتن دو مسیر MPLS به R4 اکنون در موقعیت ویژه‌ای قرار می‌گیرد و آن در اختیار داشتن دو مسیر MPLS به R4 اکنون در موقعیت ویژه‌ای قرار می‌گیرد و آن در اختیار داشتن دو مسیر MPLS به مقصد A می‌باشد. یکی از این مسیرها از طریق واسط 0 با برچسب MPLS با خروجی 10 و دیگری از طریق واسط 1 با برچسب MPLS با خروجی 8 می‌باشد. تصویری گسترده‌تر از شکل ۵-۳۶ این مهم را نشان می‌دهد که تجهیزات IP که عبارتند از R5، R6، A و D همگی از طریق یک زیرساخت MPLS شامل مسیریاب‌های LSR به ترتیب R1، R2، R3، R4 به یکدیگر متصل شده‌اند. اتصال آنها به همان صورتی است که شبکه های LAN راهیابی شده و یا ATM، قادرند تجهیزات IP را به یکدیگر ارتباط دهند. هم مانند یک شبکه ی

1- Local switch router

LAN راهیابی شده و یا ATM، مسیریاب‌های LSR به ترتیب R1 تا R4 نیز همین عمل را **بدون آن که هرگز سرپیام IP یک بسته را لمس نمایند**، انجام می‌دهند.

برچسب ورود	برچسب خروج	مقصد	واسط خروج
10 12	6 9	A D	1 0

برچسب ورود	برچسب خروج	مقصد	واسط خروج
	10 12 8	A D A	0 0 1



برچسب ورود	برچسب خروج	مقصد	واسط خروج
8	6	A	0

برچسب ورود	برچسب خروج	مقصد	واسط خروج
6	-	A	0

شکل ۵-۳۶ ♦ گسیل جلو در MPLS گسترش یافته

شاید تاکنون این نکته را دریافته‌اید که در بحث بالا به پروتکل ویژه‌ای جهت توزیع برچسب‌ها در بین مسیریاب‌های LSR اشاره نشده است. زیرا مطالب مربوط به سیگنالینگ MPLS خارج از چارچوب این اثر می‌باشد. یادآوری این نکته ضروری است که کارگروه MPLS در سازمان IETF در مستند [RFC3468] یک پروتکل توسعه یافته‌ی RSVP که در فصل ۷ آن را مطالعه می‌نمائیم تعریف نموده است که معروف به RSVP-TE [RFC3209] می‌باشد و تمرکز آن بر سیگنالینگ MPLS بنیان نهاده شده است. خوانندگان علاقه‌مند دعوت به مطالعه‌ی مستند [RFC3209] می‌شوند.

تاکنون تاکید مطالب MPLS براین واقعیت بنا نهاده شد، که MPLS عملیات راهیابی را برمبنای برچسب‌ها، بدون نیاز به درنظر گرفتن آدرس IP یک بسته انجام می‌دهد. ویژگی اساسی MPLS و علت علاقه‌مندی فعلی به MPLS به واسطه‌ی افزایش سرعت راهیابی نیست بلکه علت اصلی آن توان‌مندی‌های بالقوه‌ی مدیریت ترافیک در MPLS می‌باشد. همان‌گونه که در بالا ملاحظه شد، مسیریاب R4 از دو مسیر MPLS به مقصد A بهره‌مند است. اگر قرار بود از روش آدرس‌دهی IP استفاده شود، پروتکل‌های مسیریابی IP فقط یک مسیر را به عنوان مسیر حداقل هزینه به مقصد A معرفی می‌نمودند. به عبارت دیگر MPLS این توان‌مندی را دارد که مسیرهای غیرممکن از طریق مسیریابی استاندارد IP برای گسیل داده گرام‌ها را امکان‌پذیر نماید. این یک الگوی ساده از **مهندسی ترافیک** به شمار می‌آید که با بهره‌گیری از MPLS برمبنای مستندهای [Xiaoz000, RFC2702, RFC3272, RFC3246] امکان‌پذیر می شود و در آن یک راهبر شبکه از همه‌ی قواعد مسیریابی IP می‌تواند صرف‌نظر نماید و برخی از ترافیک جاری به سمت یک مقصد را

مجبور به حرکت به سمت یک مسیر ویژه و باقی مانده‌ی ترافیک را به سمت همان مقصد از طریق مسیر دیگری براساس سیاست‌گذاری، کارایی و یا دلایل دیگر جاری نماید.

از MPLS برای اهداف بی‌شمار دیگری نیز می‌توان بهره گرفت. از جمله می‌توان جهت انجام بازیابی سریع مسیر گسیل جلوی MPLS مانند تغییر یک مسیر از پیش محاسبه شده که ممکن است در هنگام انجام عملیات عادی خود دچار عیب شده باشد نیز بهره گرفت. از همه مهم‌تر بهره‌گیری از MPLS جهت پیاده‌سازی شبکه‌هایی است که معروف به **شبکه‌های خصوصی مجازی[1] (VPN)** می‌باشند. در پیاده‌سازی VPN معمولاً یک ISP از شبکه‌ی مجهز به MPLS خود جهت ارتباط شبکه‌های مختلف یک مشتری بهره می‌گیرد. از پروتکل MPLS می‌توان جهت جداسازی منابع و آدرس‌های بهره‌برداری شده از سوی مشتری VPN در مقابل کاربران دیگری که از شبکه‌ی آن ISP عبور می‌کنند نیز بهره گرفت.

آن‌چه در مورد MPLS در این زیربخش تحلیل شد، قطره‌ای از رودخانه بود. با توجه به امکانات فراوانی که MPLS می‌تواند در اختیار اهداف شبکه قرار دهد، این انتظار می‌رود که به سرعت به یک ابزار همه فن حریف در مهندسی ترافیک شبکه‌ی اینترنت تبدیل گردد.

۵-۱۳ یک مثال ساده‌ی وب گردی

با تأمل بر این نکته که پوشش لایه‌ی پیوند در این فصل و لایه‌های شبکه، انتقال و کاربرد در فصل‌های قبل سفر ما را در مسیر پشته‌ی پروتکلی از فراز به فرود تقریباً به پایان می‌رساند! و در آغاز این اثر و در زیربخش ۱-۱ این جمله عنوان شد که **«بسیاری از مطالب این اثر متمرکز بر پروتکل‌های شبکه‌های کامپیوتری است»**.

بدیهی است در پنج فصل نخست این واقعیت به خوبی ملاحظه گردید. قبل از گام نهادن در موضوعات بخش دوم کتاب شاید معقول باشد، پویش در امتداد پشته‌ی پروتکلی را از دیدگاه یکپارچگی پروتکل‌هایی که آن‌ها را آموخته‌ایم تحلیل نمائیم. شاید یک رویکرد برای مشاهده‌ی این تصویر بزرگ، شناسایی بسیاری از پروتکل‌هایی است که در یک موضوع ساده‌ی دانلود برای یک صفحه‌ی وب دخالت دارند. شکل ۵-۳۷ این صحنه را برای ما به نمایش می گذارد دانشجویی به نام باب (B) رایانه‌ی همراه خود را به راهیاب اترنت دانشکده متصل می‌کند و یک صفحه‌ی وب مانند: www.google.com را اخذ می‌کند. همان‌گونه که تاکنون آموخته‌ایم فعالیت‌های پنهان بی‌شماری برای به انجام رسانیدن این درخواست بسیار ساده روی می دهد. هدف ما بررسی تکاملی این فعالیت‌ها تا رسیدن به صفحه‌ی وب موردنظر با توجه به مطالب ۵ فصل اول این اثر می‌باشد.

فرض کنید B رایانه‌ی همراه خود را روشن می‌کند، منتظر فعال شدن کامل سیستم عامل نصب شده در رایانه می‌شود. پس از آن کامپیوتر خود را به یک شبکه‌ی اترنت کابلی که به راهیاب اترنت دانشکده متصل می‌باشد، ارتباط می‌دهد. این شبکه‌ی اترنت کابلی نیز از طرف دیگر به روتر نشان داده شده در شکل ۵-۳۷ متصل شده است. از سوی دیگر مسیریاب دانشکده نیز به یک ISP مرتبط می‌باشد که در این مثال: Comcast.net نام دارد. در این مثال Comcast.net خدمات DNS دانشکده را تأمین می‌کند بنابراین سرور DNS در شبکه‌ی Comcast ساکن است و نه در شبکه‌ی دانشکده. فرض می‌شود سرور DHCP نیز در بین و درون روتر شبکه همان‌گونه که مرسوم است، در حال اجرا می‌باشد.

1- Virtual private networks

هنگامی که باب (B) نخست رایانه‌ی همراه خود را به شبکه ارتباط می‌دهد، بدون در اختیار داشتن آدرس IP قادر به انجام هیچ‌گونه عملیاتی نمی‌باشد. بنابراین نخستین فعالیت مرتبط با شبکه که کامپیوتر B انجام می دهد، پردازش پروتکل DHCP جهت دستیابی به یک آدرس IP مجاز و نیز سایر اطلاعات از سرور محلی DHCP می‌باشد. اکنون ادامه‌ی مراحل را با دنبال نمودن سلسله مراتب فعالیت انجام شده از سوی رایانه‌ی همراه B از گام ۱ تا ۲۴ دنبال می‌کنیم

- **اخذ اطلاعات DHCP، UDP، IP و اترنت**

۱- سیستم عامل B یک **پیام درخواست DHCP** را براساس بخش ۴-۹ تعریف می‌کند و این پیام را در درون یک **قطعه‌ی UDP** براساس بخش ۳-۶ با آدرس درگاه مقصد 67 یعنی سرور DHCP آدرس پورت مبدأ 68 یعنی مشتری DHCP قرار می‌دهد. قطعه‌ی UDP سپس در درون یک **داده گرام IP** براساس بخش ۴-۹ با یک آدرس مقصد فراپخش IP در قالب 255.255.255.255 و نیز یک آدرس مبدأ IP به صورت 0.0.0.0 قرار داده می‌شود. آدرس مبدأ 0.0.0.0 می‌باشد زیرا رایانه‌ی همراه B هنوز مالک یک آدرس IP مجاز نشده است.

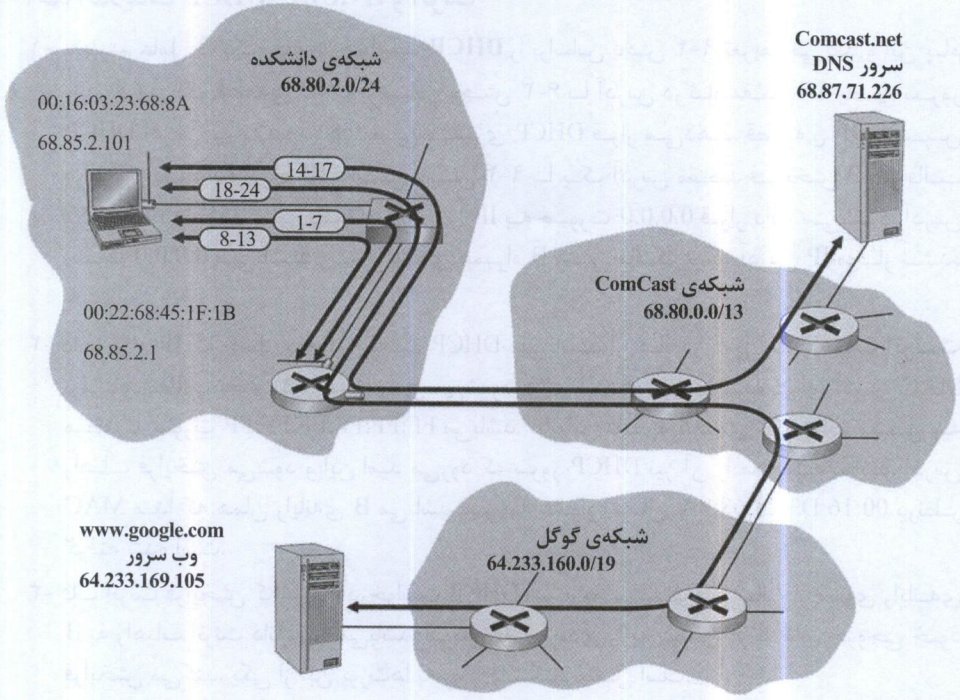
۲- داده گرام IP که حامل پیام درخواست DHCP می‌باشد آن‌گاه در درون یک **قاب اترنت** برمبنای مطالب بخش ۵-۶ بسته‌بندی می‌شود. قاب اترنت اشاره شده دارای آدرس MAC مقصد به صورت FF:FF:FF:FF:FF:FF می‌باشد. بنابراین قاب به همه‌ی تجهیزات متصل به راهیاب فراپخش می‌شود و این امید می‌رود که سرور DHCP نیز آن را ملاحظه نماید. آدرس MAC مبدأ که همان رایانه‌ی B می‌باشد نیز به عنوان مثال: 00:16:D3:23:68:8A درنظر گرفته شده است.

۳- قاب اترنت فراپخش که شامل درخواست DHCP است، نخستین فریم ارسالی از سوی رایانه‌ی B به راهیاب اترنت دانشکده می‌باشد. سوئیچ قاب ورودی را به همه‌ی درگاه‌های خروجی خود فراپخش می‌کند. یکی از این پورت‌ها به روتر دانشکده متصل است.

۴- مسیریاب، قاب اترنت فراپخش را که شامل درخواست DHCP است دریافت می‌دارد. مکان دریافت این فریم بر روی واسط شبکه با آدرس MAC به صورت: 00:22:6B:45:1F:1B می‌باشد. داده گرام IP از قاب جداسازی می‌شود و سپس آدرس فراپخش داده گرام IP نشان می‌دهد که این داده گرام بایستی از سوی پروتکل‌های لایه‌ی بالاتر برروی گره‌ی این واسط پردازش گردد. در نتیجه محموله‌ی داده گرام یعنی قطعه‌ی UDP بر مبنای بخش ۳-۲ **بازگشایی تسهیم** (دی مالتی پلکس) به UDP شده و پیام درخواست DHCP از قطعه‌ی UDP اخذ می‌شود. سرور DHCP اکنون پیام تقاضا را در اختیار دارد.

۵- فرض کنید DHCP سرور در حال اجرا در درون روتر قادر به اختصاص آدرس‌های **سایدر** براساس بخش ۴-۹ می‌باشد و توده آدرس 68.85.2.0/24 را به این فعالیت اختصاص می‌دهد. در این مثال همه‌ی آدرس‌های IP بهره‌برداری شده در درون دانشکده و شبکه‌ی Comcast.net از میان این توده می‌باشند. فرض کنید سرور DHCP آدرس IP به صورت: 68.85.2.101 را به B اختصاص می‌دهد. سرور DHCP یک **پیام تأئید DHCPACK** برمبنای بخش ۴-۹ را تعریف می‌کند که این پیام علاوه بر آدرس IP: 68.85.0.10 آدرس IP سرور DNS

(68.87.71.226)، آدرس IP مسیریاب دریچه‌ی پیش‌فرض یعنی 68.85.2.1 و توده آدرس IP متعلق به زیرشبکه‌ی 68.85.2.0/24 که معادل پوشش شبکه است را نیز شامل می شود. پیام DHCP در درون یک قطعه‌ی UDP که در یک داده گرام IP است، قرار می‌گیرد. این مجموعه نهایتاً در درون قاب اترنت بسته‌بندی می‌شوند. فریم اترنت یک آدرس MAC مبدأ از واسط مسیریاب به شبکه‌ی مسکونی 00:22:68:45:1F:1B و یک آدرس MAC مقصد از رایانه‌ی B یعنی: 00:16:D3:23:68:8A را در اختیار می‌گیرد.



شکل ۵-۳۷ ♦ کاوش در وب

۶- قاب اترنت حامل DHCPACK به صورت تک پخش از طریق مسیریاب به راهیاب گسیل می‌شود. از آن‌جائیکه سوئیچ براساس مطالب بخش ۵-۸ حالت **خودیادگیری** دارد و قبلاً نیز یک قاب اترنت شامل پیام درخواست DHCP را از B دریافت نموده است، این آگاهی را در اختیار دارد که بایستی یک قاب را به آدرس 00:16:D3:23:68:8A گسیل نماید. این آدرس فقط به درگاه خروجی مربوط می‌شود که رایانه همراه B را شامل می‌شود.

۷- رایانه‌ی سیار B قاب اترنتی را دریافت می‌کند که حامل پیام DHCPACK است. داده گرام IP را از آن قاب خارج می‌کند و قطعه‌ی UDP را در نتیجه‌ی این عمل از داده گرام جدا می‌کند و سرانجام نیز پیام DHCPACK را از قطعه‌ی UDP اخذ می‌کند. برنامه‌ی مشتری DHCP در رایانه‌ی B آدرس‌های IP و سرور DNS خود را ثبت می‌کند. این برنامه همچنین آدرس مسیریاب دریچه‌ی پیش فرض را نیز در **جدول گسیل جلوی IP** براساس بخش ۴-۱ نصب می‌کند. رایانه‌ی همراه B همه‌ی داده گرام‌های با آدرس‌های مقصد خارج از زیر شبکه‌اش را که توده آدرس IP: 68.85.2.0/24 می‌باشند به مسیریاب دریچه‌ی پیش فرض گسیل می‌کند. در

این مقطع کامپیوتر B عملیات مقداردهی اولیه‌ی عناصر شبکه‌ی خود را انجام داده و آماده‌ی آغاز فرآیند واکشی صفحه‌ی وب مورد نظر می‌شود. بدیهی است فقط دو مرحله ی آخر DHCP از میان چهار مرحله‌ای که در فصل ۴ تشریح شدند، لازم می‌باشد.

- **به کارگیری پروتکل‌های DNS و ARP**

هنگامی که B آدرس URL مورد نظرش یعنی : www.google.com را در مرورگر وب کامپیوتر خود وارد می‌کند، زنجیره‌ای از رخدادها که سرانجام آن صفحه‌ی وب خانه‌ی Google می‌باشد از سوی مرورگر وب B به نمایش درمی‌آید. مرورگر وب B آغاز این فرآیند را از طریق تعریف یک **سوکت TCP** براساس بخش ۲-۲۱ کلید می زند که از این سوکت جهت ارسال پیام **درخواست HTTP** به آدرس URL اشاره شده استفاده می‌شود (بخش ۲-۶). جهت تعریف و ایجاد سوکت، کامپیوتر همراه B نیازمند آگاهی از آدرس IP برای آدرس اینترنتی www.google.com می‌باشد. در بخش ۲-۱۳ آموختیم که از **پروتکل DNS** جهت انجام ترجمه‌ی نام میزبان به آدرس IP استفاده می‌شود.

۸- سیستم عامل نصب شده در رایانه‌ی B بنابراین یک **پیام درخواست DNS** را تعریف نموده و رشته‌ی www.google.com را در بخش پرسش پیام DNS وارد می‌کند. این پیام DNS سپس در درون یک قطعه‌ی UDP با درگاه مقصد شماره‌ی 53 یعنی سرور DNS قرار می‌گیرد. قطعه‌ی UDP سپس در داخل یک داده گرام IP با آدرس IP مقصد به صورت: 68.87.71.226 که همان آدرس سرورDNS بازگشتی در گام پنجم DHCPACK می‌باشد، جایگذاری می‌شود. آدرس IP مبدأ داده گرام نیز: 68.85.2.101 می‌باشد.

۹- کامپیوتر B سپس داده گرام حامل پیام تقاضای DNS را در یک قاب اترنت قرار می‌دهد. این فریم از طریق هدایت لایه‌ی پیوند به مسیریاب دریچه در شبکه‌ی دانشکده‌ی B گسیل می‌شود. اگرچه رایانه‌ی B از آدرس IP مسیریاب دریچه‌ی دانشکده به آدرس: 68.85.2.1 از طریق پیام DHCPACK در گام ۵ مراحل بالا آگاهی دارد دریچه کامپیوتر B نیازمند بهره‌گیری از **پروتکل ARP** است که در بخش ۵-۵ تحلیل آن انجام شده است.

۱۰- کامپیوتر همراه B یک پیام **تقاضای ARP** با آدرس مقصد IP به صورت: 68.85.2.1 که مسیریاب دریچه پیش فرض است تعریف می‌کند، پیام ARP را در درون یک قاب اترنت جایگذاری می‌کند و آن قاب را حامل آدرس فراپخش مقصد به صورت: FF:FF:FF:FF:FF:FF می‌کند. فریم اترنت به راهیاب گسیل می‌شود و قاب به همه‌ی تجهیزات متصل شامل مسیریاب دریچه تحویل می‌گردد.

۱۱- مسیریاب دریچه، قاب حامل پیام تقاضای ARP را از واسط شبکه‌ی دانشکده دریافت می‌کند و از نگاشت آدرس IP مقصد 68.85.2.1 در پیام ARP با آدرس IP واسط خود آگاه می‌شود. روتر دریچه یک پیام **پاسخ ARP** را آماده می‌کند و اعلام می دارد آدرس MAC آن با مشخصات: 00:22:6B:45:1F:1B مربوط به آدرس IP: 68.85.2.1 می‌باشد. پس از آن پیام پاسخ ARP را در یک قاب اترنت جایگذاری و آدرس مقصد آن نیز به صورت: 00:16:D3:23:68:8A (رایانه‌ی B) تعریف می‌شود و قاب را به راهیاب شبکه‌ی دانشکده گسیل می‌کند و آن نیز قاب را تحویل کامپیوتر همراه B می‌دهد.

۱۲- رایانه‌ی B فریم حامل پیام پاسخ ARP را دریافت می‌کند و آدرس MAC مسیریاب دریچه

یعنی: 00:22:6B:45:1F:1B را از پیام پاسخ ARP جدا می‌کند.

۱۳- سرانجام رایانه‌ی B می‌تواند قاب اترنت حامل درخواست DNS را به آدرس MAC مسیریاب دریچه هدایت نماید. لازم به یادآوری است که داده گرام IP در این قاب حامل یک آدرس مقصد IP به صورت 68.87.71.226 است که همان سرور DNS است. قاب داده که حامل آن داده گرام است نیز آدرس مقصد 00:22:6B:45:1F:1B را که همان مسیریاب دریچه است حمل می‌کند. رایانه‌ی همراه B این قاب را به سوئیچ گسیل می‌کند و آن نیز فریم را به مسیریاب دریچه تحویل می‌دهد.

- **مسیریابی بین قلمرو تا سرور DNS**

۱۴- مسیریاب دریچه پس از دریافت و جداسازی داده گرام IP که حامل تقاضای DNS است به جستجوی آدرس مقصد این داده گرام (68.87.71.226) می‌پردازد. از اطلاعات جدول گسیل جلو آگاه می‌شود که داده گرام بایستی به روتر سمت چپ در شبکه‌ی Comcast در شکل ۵-۳۷ ارسال گردد. داده گرام IP در درون یک قاب لایه‌ی پیوند مناسبی جایگذاری می‌شود که وابسته به پیوندی است که مسیریاب دانشکده را به روتر سمت چپ Comcast متصل می‌کند. آن‌گاه این قاب از طریق این پیوند گسیل می‌شود.

۱۵- مسیریاب طرف چپ در شبکه‌ی Comcast قاب را دریافت می‌دارد، داده گرام IP را از آن جدا می‌کند، آدرس مقصد داده گرام یعنی 68.87.71.226 را تحلیل می‌کند واسط خروجی جهت ارسال داده گرام به سرور DNS را از جدول گسیل جلوی خود که اطلاعات آن از سوی یک پروتکل مسیریابی بین قلمرو مانند **OSPF, RIP** و یا **IS-IS** و نیز **پروتکل بین قلمرو و اینترنت BGP** در شبکه Comcast فراهم آمده است، را برمبنای بخش ۴-۱۹ شناسایی می‌کند.

۱۶- سرانجام داده گرام IP که حامل تقاضای DNS است وارد سرور DNS می‌شود. پیام تقاضا از طریق سرور DNS بازیابی می‌شود و آن‌گاه به جستجوی نام www.google.com در بانک اطلاعات DNS خود می‌پردازد و برمبنای مطالب بخش ۲-۱۳ **سابقه‌ی منبع DNS** را شناسایی می‌کند که حامل آدرس IP برابر 65.232.169.105 برای نام www.google.com است. بدیهی است این بازیابی درصورتی انجام می‌شود که این آدرس در حال حاضر در حافظه‌ی نهان سرور DNS موجود می‌باشد. بدیهی است منبع این اطلاعات نهان شده **سرور خودمختار پیام پاسخ DNS** را تشکیل می‌دهد که در برگیرنده‌ی نگاشت نام میزبان به آدرس IP می‌باشد. این پیام پاسخ در قطعه‌ی UDP قرار می‌گیرد و قطعه پس از بسته‌بندی به صورت یک داده گرام IP به سمت رایانه B یعنی آدرس 68.85.2.101 هدایت می‌شود. این داده گرام از طریق شبکه‌ی Comcast به مسیریاب دانشکده گسیل و از آن‌جا از طریق راهیاب اترنت به کامپیوتر B تحویل می‌گردد.

۱۷- رایانه‌ی همراه B آدرس IP سامانه‌ی google.com را از پیام پاسخ DNS بازیابی و سرانجام پس از عملیات طاقت فرسای فراوانی کامپیوتر همراه B آماده‌ی تعامل با سرور www.google.com می‌شود!

- **تعامل برنامه‌های مشتری و سرور وب: TCP و HTTP**

۱۸- اکنون که رایانه‌ی همراه B آدرس IP شرکت google.com را به‌دست آورده است، می‌تواند **سوکت TCP** را برمبنای بخش ۲-۲۱ تعریف نماید و از آن برای ارسال **HTTP GET** برمبنای بخش ۲-۷ به سامانه‌ی google.com استفاده نماید. هنگامی که B سوکت TCP را تعریف می‌کند، پروتکل TCP در رایانه‌ی B نخست یک عملیات **دست دهی سه جانبه** را برمبنای بخش ۳-۱۷ با برنامه‌ی TCP در google.com انجام می دهد. بنابراین کامپیوتر همراه B نخست یک قطعه‌ی **TCPSYN** به آدرس پورت مقصد 80 برای HTTP تعریف می‌کند و این قطعه‌ی TCP را در درون یک داده گرام IP با آدرس IP مقصد 64.233.169.105 (www.google.com) قرار می‌دهد. این داده گرام سپس در درون یک قاب با آدرس MAC مقصد به صورت: 00:22:6B:45:1F:1B که همان مسیریاب دریچه است، جایگذاری شده و سپس به سوی راهیاب شبکه گسیل می‌شود.

۱۹- مسیریاب‌های شبکه‌های دانشکده، Comcast و google داده گرام حامل TCPSYN را به سوی www.google.com با بهره‌گیری از جدول گسیل جلو در هر مسیریاب براساس مراحل 14 تا 16 در بالا هدایت می‌کنند. یادآوری می‌شود اطلاعات درون جدول گسیل جلو مدیریت گسیل بسته‌ها را برروی پیوند بین قلمرو که مابین شبکه‌های google و Comcast قرار دارد و از طریق پروتکل **BGP** برمبنای بخش ۴-۲۱ محاسبه می‌شود، انجام می‌دهد.

۲۰- سرانجام داده گرام حامل TCPSYN وارد google.com می‌شود. پیام TCPSYN از داده گرام مجزا و از حالت تسهیم خارج و وارد **سوکت خوش‌آمدگو** وابسته به پورت 80 می‌شود. یک سوکت ارتباطی برمبنای مطالب بخش ۲-۲۱ تعریف می‌شود تا ارتباط TCP بین سرور HTTP در گوگل و برنامه مشتری رایانه‌ی همراه B رسمیت خود را به دست آورد. یک پیام TCPSYNACK در قالب یک قطعه براساس مطالب بخش ۳-۱۷ تولید و در درون داده گرام هدایت شده و به سمت B قرار می‌گیرد و سرانجام نتیجه‌ی حاصل از این فرآیند در درون یک قاب داده‌ی لایه‌ی پیوند که مناسب ارتباط با نخستین مسیریاب پرش کوتاه سامانه‌ی google.com می‌باشد، جایگذاری می‌شود.

۲۱- داده گرام حامل قطعه‌ی TCPSYNACK از درون شبکه‌های google، Comcast و دانشکده عبور می‌کند و سرانجام وارد کارت شبکه‌ی اترنت در کامپیوتر همراه B می‌شود. داده گرام درون قاب از طریق سیستم عامل به سوکت TCP ایجاد شده در مرحله‌ی ۱۸ که وارد وضعیت اتصال می‌شود، تحویل می‌گردد.

۲۲- سرانجام سوکت رایانه‌ی B آماده‌ی ارسال بایت‌ها به google.com می‌شود و بنابراین مرورگر B پیام HTTPGET را تعریف می‌کند و در درون آن آدرس URL گوگل را قرار می‌دهد. پیام HTTPGET به درون سوکت نگارش می‌شود و پیام GET به عنوان محموله یک قطعه‌ی TCP تعریف می شود. قطعه‌ی TCP در یک داده گرام جایگذاری و به سوی google.com برمبنای مراحل ۱۸ تا ۲۰ بالا گسیل می‌شود.

۲۳- سرور HTTP در گوگل پیام HTTPGET را از سوکت TCP می‌خواند و یک پیام **واکنش (پاسخ) HTTP** را برمبنای مطالب بخش ۲-۶ خلق می‌کند و صفحه‌ی وب مورد تقاضا را نیز وارد بدنه‌ی این پیام HTTP می‌کند تمام مجموعه را به درون سوکت TCP تزریق می‌کند.

۲۴- داده گرام حامل پیام پاسخ HTTP از درون شبکه‌های گوگل، Comcast و دانشکده عبور کرده و وارد رایانه‌ی همراه B می‌شود. برنامه‌ی مرورگر B پیام HTTP را از سوکت خود خوانده و بخش **html** صفحه‌ی وب مورد نظر را از بدنه‌ی پاسخ HTTP جدا می‌کند و سرانجام آن را به عنوان یک صفحه‌ی وب به نمایش می‌گذارد!

گردش در وب که در بالا مورد اشاره قرار گرفت شامل بسیاری از عملیات یک شبکه‌ی کامپیوتری می‌باشد. اگر شما همه‌ی مطالب بالا و یا بخش وسیعی از آن را متوجه شدید آن‌گاه می‌توانید امیدوار باشید که درک نسبتاً عمیقی از شبکه‌های کامپیوتری را به دست آورده‌اید. خوانندگان مشتاق به مستند [RFC5218] جهت گسترش دانش بیش‌تر در این چارچوب هدایت می‌شوند.

اگرچه سفر از فراز پشته‌ی پروتکلی به فرود آن در این‌جا به پایان خود نزدیک می‌شود اما مطالعه‌ی شبکه‌های کامپیوتری هنوز به اتمام نرسیده است. در چهار فصل آتی شبکه‌های بدون سیم و چندرسانه‌ای و نیز امنیت و مدیریت شبکه‌های کامپیوتری تحلیل می‌شوند. این چهار عنوان را نمی‌توان در هیچ یک از لایه‌ها قرار داد. بدیهی است هریک از این چهارعنوان خود در برگیرنده تعداد قابل توجهی لایه می‌باشند، بنابراین درک کامل این عناوین نیازمند آگاهی عمیق از چگونگی ساختار و عملکرد لایه‌های شبکه می‌باشد که با مطالعه‌ی فصل ۵ و لایه‌ی پیوند داده‌ها این مهم نیز به سرانجام خود می‌رسد.

۱۴-۵ شبکه‌های مراکز داده

در سال‌های اخیر شرکت‌های اینترنتی مانند گوگل، مایکروسافت، فیس‌بوک و آمازون به همراه رقبای آسیایی و اروپایی خود اقدام به ساخت و گسترش مراکز داده در مقیاس و ظرفیت‌های فوق‌العاده عظیم نموده‌اند. هر یک از این مراکز داده ده‌ها و صدها هزار میزبان را فضاسازی کرده‌اند و به صورت همزمان بسیاری از کاربردهای ابری متمایز از یکدیگر مانند جست و جو، ایمیل، شبکه‌های اجتماعی و تجارت الکترونیک را پشتیبانی می‌کنند. هر مرکز داده بهره‌مند از **شبکه‌ی مرکز داده‌ی**[1] مستقل خود می‌باشد که میزبان‌های خویش را با یکدیگر و مرکز داده را با اینترنت ارتباط می‌دهند. در این زیربخش تحلیل مختصری از مراکز داده به منظور بهره‌گیری در کاربردهای ابری ارایه می‌شود.

مخارج ایجاد و نگهداری یک مرکز داده معمولاً بیش از 12 میلیون دلار آمریکا در هر ماه برای تعداد یکصدهزار میزبان در مرکز داده است. از میان این هزینه‌ی بسیار گزاف نزدیک به 45 درصد آن مربوط به میزبان‌ها می‌شود که هر 3 یا 4 سال بایستی با فناوری جدید جایگزین شوند و 25 درصد به زیر ساخت این مراکز مربوط می‌شود که شامل تجهیزات تامین نیرو شامل UPS، مبدل‌ها، مولدهای اضطراری و سیستم‌های سرمایشی می‌شود. نزدیک به 15 درصد این مخارج هزینه پرداخت الکتریسیته‌ی مصرفی است و 15 درصد باقی‌مانده نیز متعلق به تجهیزات شبکه شامل راهیاب‌ها، مسیریاب‌ها و پیوندهای ارتباطی خارجی و نیز هزینه‌ی پرداختی بابت ترافیک ترانزیت این مراکز است اگرچه در این میان شبکه‌سازی بالاترین هزینه را شامل نمی‌شود اما نوآوری‌های شبکه عامل بسیار مهمی در کاهش کلی هزینه‌ها و نیز افزایش قابل توجه کارآیی آنها می‌تواند به شمار آید.

زنبورهای کارگر در یک مرکز داده میزبان‌ها هستند. آنها خدمات محتوایی مانند صفحات وب و فایل‌های صوتی و تصویری را خدمات رسانی می‌کنند. مستندات و ایمیل‌ها را ذخیره می‌کنند و به صورت

1 - Data center network

جمعی پردازش‌های توزیع شده‌ی گسترده‌ای مانند محاسبات توزیع شاخص برای موتورهای جست و جوی شبکه را انجام می‌دهند. میزبان‌های واقع در مراکز داده اصطلاحاً **تیغه‌ها**[1] نامیده می‌شوند و معمولاً یادآور جعبه‌های پیتزا هستند چرا که آنها را می‌توان میزبان‌های کالا مانند CPU، حافظه، دیسک و غیره در نظر گرفت. میزبان‌ها معمولاً در قفسه‌ها پشته می‌شوند و هر قفسه معمولاً شامل 20 تا 40 تیغه است. در قسمت فوقانی هر قفسه یا راک یک راهیاب (سوئیچ) وجود دارد که اصطلاحاً به آن **راهیاب فوقانی قفسه**[2] (TORS) گفته می‌شود و میزبان‌های درون قفسه‌ها را به یکدیگر و به راهیاب‌های دیگر در مرکز داده پیوند می‌دهد. به‌گونه‌ای خاص هر میزبان در قفسه دارای یک کارت واسط شبکه است که آن را به TORS متصل می‌نماید و هر TORS نیز بهره‌مند از پورت‌های اضافه‌تر به منظور ارتباط با راهیاب‌های دیگر یا همان TORS می‌باشد. اگرچه میزبان‌ها در حال حاضر بهره‌مند از ارتباط‌های اترنت 1Gbps به راهیاب‌های خود می‌باشند اما ارتباط‌های با سرعت 10 Gbps در حال گسترش هستند. معمولاً به هر میزبان آدرس IP داخلی مرکز داده نیز اختصاص می‌یابد.

شبکه‌ی مرکز داده دو نوع ترافیک را پشتیبانی می‌کند: (۱) ترافیک جاری بین مشتریان خارجی و میزبان‌های داخلی و (۲) ترافیک جاری بین میزبان‌های داخلی. به منظور تحقق جریان‌ها بین مشتریان خارجی و میزبان‌های داخلی، شبکه‌ی مرکز داده (DCN) معمولاً شامل یک یا چندین **مسیریاب مرزی**[3] است و DNC را به شبکه‌ی اینترنت عمومی متصل می‌کند. بنابراین شبکه‌ی DCN ارتباط بین قفسه‌ها را برقرار می‌نماید و پس از آن قفسه‌ها را به روترهای مرزی متصل می‌کند. شکل ۵-۳۸ نمایی از یک شبکه‌ی DCN را نشان می‌دهد. **طراحی شبکه‌ی DCN** هنر ارتباط بین شبکه و پروتکل‌هایی است که قفسه‌ها را به یکدیگر و به مسیریاب‌های مرزی متصل می‌کنند. این شاخه‌ی طراحی در حال حاضر یکی از مهم‌ترین موضوعات پژوهشی در عرصه‌ی شبکه‌های کامپیوتری به شمار می‌آید.

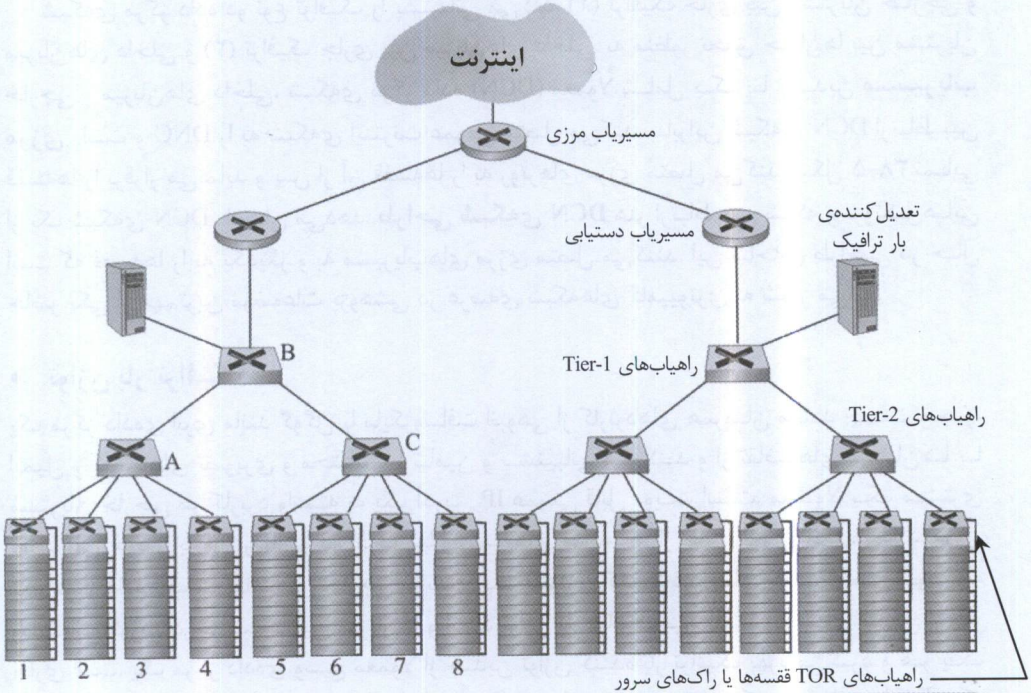
- **توازن بار ترافیک**

یک مرکز داده‌ی ابری مانند گوگل یا مایکروسافت انبوهی از کاربردهای همزمان مانند جست و جو، ایمیل و کاربردهای تصویری و محتوایی را تامین و پشتیبانی می‌کند و از تقاضاهای میزبان‌ها یا مشتریان خارجی هر کاربرد وابسته به یک آدرس IP عمومی قابل رویت است. معمولاً یک مشتری تقاضاهای خود را به آن ارسال و در طرف مقابل پاسخ‌های مناسب را از جانب آن دریافت می‌دارد. درون یک مرکز داده نخست تقاضاهای خارجی به سوی یک **توازن کننده‌ی بار**[4] هدایت می‌شود که وظیفه‌اش توزیع درخواست‌ها بین میزبان‌ها و ایجاد توازن بار ترافیک بین آنها بر مبنای تابعی از بار جاری است. یک مرکز داده‌ی وسیع معمولاً از چندین توازن کننده بار ترافیک بهره می‌گیرد و هر یک از آنها اختصاص به مجموعه‌ی ویژه‌ای از کاربردهای ابری دارند. چنین توازن کننده‌های بار ترافیک معمولاً یک **راهیاب لایه‌ی 4**[5] نامیده می‌شوند زیرا تصمیم‌گیری خود را بر مبنای شماره پورت مقصد که اطلاعات لایه‌ی چهارم است و نیز آدرس IP مقصد در بسته انجام می‌دهند. هنگام دریافت یک تقاضا برای یک کاربرد خاص توازن کننده‌ی بار ترافیک، آن درخواست را به یکی از میزبان‌های توانمند

---

1 - Blades
2 - Top of Rack Switch
3 - Border routers
4 - Load balance
5 - Switch layer 4

در پردازش آن کاربرد گسیل می‌نماید. یک میزبان در این شرایط ممکن است خدمات قابل ارایه از سوی دیگر میزبان‌ها را به منظور تسهیل در پردازش تقاضای اشاره شده، درخواست نماید. هنگامی که میزبان پردازش تقاضا را کامل نماید، پاسخ آن را به سوی توازن‌کننده‌ی بار ترافیک گسیل می‌کند و آن نیز پاسخ تامین شده را به سوی مشتری خارجی گسیل می‌دارد. توازن‌کننده‌ی بار ترافیک نه تنها بار کاری بین میزبان‌های یک DCN را متوازن می‌کند بلکه یک عملکرد شبیه ترجمه‌ی آدرس شبکه (NAT) را نیز فراهم می‌آورد که وظیفه‌ی آن ترجمه‌ی آدرس عمومی IP خارجی به آدرس IP داخلی در یک میزبان DCN و سپس برگردان مجدد آدرس برای بسته‌هایی است که در سمت مخالف به طرف مشتری در جریان هستند. این موضوع سبب می‌شود تا از تماس مستقیم مشتریان با میزبان‌های DCN ممانعت به عمل آید و در نتیجه DCN از امنیت قابل قبول در برابر تهاجم‌های احتمالی بهره‌مند گردد. این امنیت به مفهوم آن است که ساختار داخلی شبکه پنهان می‌ماند و مشتریان نمی‌توانند تعامل مستقیم با میزبان‌های DCN برقرار نمایند.



شکل ۵-۳۸ ♦ یک شبکه‌ی DCN با ساختار سلسله‌مراتبی.

• معماری سلسله مراتبی DCN

برای یک مرکز DCN کوچک که فقط شامل چند هزار میزبان است یک شبکه‌ی ساده شامل یک مسیریاب مرزی، یک توازن‌کننده‌ی بار ترافیک و ده‌ها قفسه یا راک است که همگی آنها به یک راهیاب اترنت متصل شده‌اند. برای گسترش به مقیاس ده‌ها تا صدها هزار میزبان یک DCN در اغلب مواقع از یک **سلسله مراتب روترها و راهیاب‌ها** بهره می‌گیرد. چنین جانمایی در شکل ۵-۳۸ نشان داده شده است. در بالای سلسله مراتب، مسیریاب مرزی به روترهای دستیابی متصل می‌شود که در شکل ۵-۳۸ تنها دو

فقره از آنها نشان داده شده است در صورتی که در عمل تعداد آنها به مراتب بیشتر است. پایین هر روتر دستیابی تعداد سه طبقه راهیاب وجود دارد. هر مسیریاب دستیابی متصل به یک سوئیچ طبقه‌ی بالاتر است و هر راهیاب طبقه‌ی بالاتر به چندین سوئیچ در طبقه‌ی دوم و نیز به یک توازن‌کننده‌ی بار ترافیک متصل می‌شود. هر راهیاب طبقه‌ی دوم خود به چندین قفسه از طریق TORS متصل می‌شود که راهیاب‌های طبقه‌ی سوم نامیده می‌شوند. همه‌ی پیوندها به‌گونه‌ی خاص از اترنت به منظور پروتکل‌های لایه‌های پیوند و فیزیکی بهره می‌گیرند که آمیزشی از کابل‌بندی نوری و فلزی است. با چنین طرح سلسله‌مراتبی امکان مقیاس‌پذیری یک DCN به صدها هزار میزبان فراهم می‌آید.

از آنجا که برای یک تامین‌کننده‌ی کاربردهای ابری بسیار مهم است که به صورت مستمر کاربردهای مورد نیاز متقاضیان با ضریب دسترسی بسیار بالا را تامین نماید بنابراین مراکز داده نیز شامل تجهیزات و پیوندهای افزونه در طراحی خود هستند که شکل ۵-۳۸ آنها را نشان نمی‌دهد. به عنوان مثال هر راهیاب TORS قادر است به تعداد 2 راهیاب طبقه 2 متصل شده و نیز هر روتر دستیابی، سوئیچ طبقه‌ی 1 و راهیاب طبقه‌ی 2 نیز افزونه‌ی خود را به صورت ذخیره دارند که در طرح کلی تجمیع می‌شوند. در طراحی سلسله مراتبی مانند شکل ۵-۳۸ ملاحظه‌ی این نکته ضروری است که میزبان‌های ذیل هر مسیریاب دستیابی تشکیل یک زیر شبکه را می‌دهند. به منظور موضعی‌سازی ترافیک فراپخش ARP، هر یک از این زیرشبکه‌ها خود به زیرشبکه‌های کوچک‌تر از نوع VLAN تقسیم‌بندی می‌شوند و هر کدام از آنها شامل چند صد میزبان هستند.

اگرچه معماری سلسله مراتبی سنتی توصیف شده در بالا چالش مقیاس‌پذیری را برطرف می‌سازد اما از یک اثر جانبی به صورت **محدودیت ظرفیت میزبان به میزبان** رنج می‌برد. به منظور درک این محدودیت مجدداً شکل ۵-۳۸ را در نظر گرفته و فرض کنید هر میزبان متصل به راهیاب TORS از طریق یک پیوند 1 Gbps است و این در شرایطی است که پیوندهای اترنت بین راهیاب‌ها هر کدام 10 Gbps هستند. بنابراین دو میزبان در یک قفسه‌ی مشابه همیشه قادر است با سرعت کامل 1 Gbps ارتباط برقرار نماید و تنها محدودیت آن سرعت کارت‌های واسط شبکه‌ی هر میزبان است. اگر تعداد انبوهی از جریان‌های داده‌ی همزمان در شبکه‌ی DCN وجود داشته باشند آنگاه حداکثر سرعت بین دو میزبان در قفسه‌های متفاوت می‌تواند به مراتب کمتر باشد. به منظور درک بهتر این موضوع یک الگوی ترافیک شامل 40 جریان داده‌ی همزمان بین تعداد 40 جفت از میزبان‌ها در قفسه‌های مختلف را در نظر بگیرید. به صورت خاص فرض کنید هر 10 میزبان در قفسه‌ی شماره‌ی یک در شکل ۵-۳۸ جریانی را به میزبانی در راک 5 گسیل می‌کند. علاوه بر آن فرض کنید تعداد 10 جریان داده‌ی همزمان بین جفت‌هایی از میزبان‌ها در قفسه‌ی 2 و 6 و نیز 10 جریان همزمان نیز بین قفسه‌های 3 و 7 و 10 جریان همزمان دیگر بین راک‌های 4 و 8 وجود دارند. اگر هر جریان به صورت یکسان از ظرفیت پیوندها مشترکاً با دیگر جریان‌های در حال عبور از آن پیوند استفاده نماید، آنگاه تعداد 40 جریان داده از طریق پیوند 10 Gbps از A به B و نیز پیوند با سرعت 10 Gbps از طریق B به C امکان‌پذیر می‌شود و هر کدام از آنها فقط می‌تواند به اندازه‌ی: 10 Gbps/40 = 250 Mbps از ظرفیت پیوند را دریافت نماید. در نتیجه به گونه‌ی چشمگیر به مراتب کمتر از سرعت کارت واسط شبکه‌ی هر میزبان است که معادل 1 Gbps می‌باشد. این دشواری برای جریان‌های بین میزبان‌هایی که نیاز به ارتباط با سلسله مراتب بالاتر دارند به مراتب دشوارتر می‌شود. یک راهکار برای این محدودیت بهره‌گیری از سوئیچ‌ها و مسیریاب‌های با سرعت بالاتر است. اما این راه‌حل سبب افزایش چشمگیر در هزینه‌های DCN خواهد شد. قیمت تمام شده‌ی چنین تجهیزاتی بسیار بالا است.

پشتیبانی ارتباط‌های میزبان به میزبان با سرعت بالا نیز بسیار اهمیت دارد زیرا یکی از نیازهای کلیدی هر مرکز داده انعطاف‌پذیری در تامین پردازش و خدمات مورد نیاز است. به عنوان مثال یک موتور جست و جوی اینترنتی پرظرفیت ممکن است روی هزاران میزبان گسترده شده در چندین قفسه با نیازهای پهنای باند قابل توجه بین همه‌ی جفت میزبان‌ها در حال اجرا باشد. به صورت مشابه یک خدمات رایانش ابری مانند EC2 ممکن است نیاز به تعمیم چندین ماشین مجازی شامل خدمات مشتری روی میزبان‌های فیزیکی با حداکثر بهره‌وری از پهنای باند بدون توجه به موقعیت مکانی آنها در DCN داشته باشد. اگر چنین میزبان‌های فیزیکی در سراسر چندین قفسه توزیع شده باشند آنگاه تنگناهای توصیف شده در بالا باعث کارایی بسیار ضعیف DCN خواهد شد.

- **جهت‌گیری مطلوب در DCN**

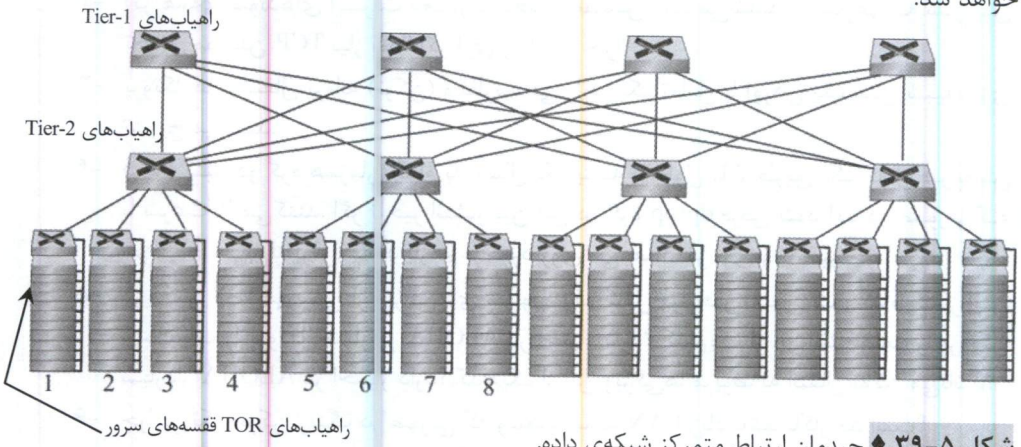
به منظور کاهش مخارج مراکز داده و نیز بهینه‌سازی همزمان تاخیر و نهفتگی کارایی شرکت‌های بزرگ اینترنتی خدمات ابری مانند گوگل، فیس بوک، آمازون و مایکروسافت همواره در پی طرح‌های جدید پیاده‌سازی مراکز داده هستند. اگرچه همه‌ی این طراحی‌ها خصوصی و در مالکیت آنها می‌باشد اما در درون آنها جهت‌گیری‌های انجام شده قابل مشاهده است.

یکی از این جهت‌گیری‌ها بهره‌گیری از معماری‌های جدید ارتباطی و پروتکل‌های شبکه است که نقاط ضعف طرح‌های سلسله مراتبی سنتی را برطرف می‌سازد. یک رویکرد جایگزین، سلسله مراتبی از سوئیچ‌ها و مسیریاب‌ها با **چیدمان‌های کاملاً متصل**[1] به یکدیگر است. این توپولوژی در شکل ۵-۳۹ نشان داده شده است. در این طراحی هر راهیاب رده‌ی نخست به همه‌ی سوئیچ‌های طبقه‌ی 2 متصل می‌شود به گونه‌ای که: (۱) ترافیک میزبان به میزبان هرگز بیشتر از ظرفیت راهیاب‌های هر طبقه نمی‌شود و (۲) با وجود n راهیاب در طبقه‌ی نخست بین هر دو سوئیچ طبقه‌ی 2 تعداد n مسیر تفکیک شده وجود دارد. چنین طراحی می‌تواند ظرفیت و سرعت میزبان به میزبان را به صورت چشمگیری افزایش دهد. به منظور مشاهده‌ی این تاثیر مجدداً مثال 40 جریان داده را در نظر بگیرید. جانمایی شکل ۵-۳۹ به آسانی می‌تواند نیازهای چنین الگوی جریان داده را تامین نماید زیرا تعداد 4 مسیر متمایز بین نخستین راهیاب طبقه 2 و دومین سوئیچ در همان رده وجود دارد که همراه یکدیگر ظرفیت انباشتی 40 Gbps را بین دو راهیاب نخست در طبقه‌ی 2 فراهم می‌آورند. چنین طراحی نه تنها محدودیت‌های سرعت میزبان به میزبان را کاهش می‌دهد بلکه یک محیط انعطاف‌پذیر خدمات و پردازش را فراهم می‌آورد که ارتباط بین هر دو قفسه یا راک که متصل به یک راهیاب مشابه نیستند را از نقطه نظر منطقی بدون توجه به موقعیت‌های مکانی مراکز داده یکسان می‌سازد.

رویکرد مهم دیگر بهره‌گیری از روش انباشت کانتینرهای حمل و نقل دریایی است که اصطلاحاً به آن **مراکز داده‌ی مدولار (MDC)**[2] گفته می‌شود. شبکه‌ی یوتیوب نمونه‌ای از این رویکرد است. در یک MDC سازندگان، یک کانتینر استاندارد 12 متری را در قالب یک مرکز داده‌ی کوچک جانمایی نموده و آن را به مکانی که برای نصب در نظر گرفته شده است ارسال می‌کنند. هر کانتینر شامل چندهزار میزبان است که در قالب ده‌ها قفسه پشته شده است و ارتباط فیزیکی نزدیک با یکدیگر دارند. در یک مکان مرکز داده چندین کانتینر مشابه هم به یکدیگر و در نهایت به شبکه‌ی اینترنت متصل می‌شوند. هنگامی که یک کانتینر از پیش طراحی و تولید شده در یک مرکز داده مورد استفاده قرار

1 - Fully connected topology
2 - Modular data centers

می‌گیرد خدمات نگهداری آن دشوار خواهد شد بنابراین هر کانتینر به گونه‌ای طراحی می‌شود که خدمات‌رسانی آن هرگز متوقف نگردد اگرچه کارایی آن به شدت کاهش یابد. در صورت ظهور عیب در قطعه‌هایی مانند سرورها و راهیاب‌ها کانتینر مرکز داده به عملیات خود با کارایی کمتر ادامه می‌دهد. هنگامی که تعداد بخش‌های معیوب به صورت قابل ملاحظه‌ای افزایش و کارایی آن از آستانه‌ی قابل قبول تنزل پیدا کند آنگاه تمامی کانتینر از مدار خارج و جایگزین آن یک مرکز داده‌ی کاملاً جدید خواهد شد.



شکل ۵-۳۹ ♦ چیدمان ارتباط متمرکز شبکه‌ی داده.

ساختن یک مرکز داده با بهره‌گیری از روش کانتینر چالش‌های جدیدی را برای موضوعات شبکه‌سازی کامپیوتری ایجاد می‌کند. با یک MDC دو نوع ساختار شبکه را می‌توان تعریف کرد: (۱) شبکه‌های داخلی کانتینر به گونه‌ای طراحی شود که در درون آن هسته‌ی شبکه، هر کانتینر را به دیگری متصل می‌کند و (۲) در درون هر کانتینر به ظرفیت چند هزار میزبان امکان تعریف یک شبکه‌ی کاملاً متصل با استفاده از راهیاب‌های اینترنت تجاری ارزان قیمت دارای ظرفیت Gbps امکان‌پذیر باشد. در هر صورت طراحی هسته‌ی شبکه نیازمند ارتباط با صدها و یا هزاران کانتینر است و این در شرایطی است که پهنای باند وسیع میزبان به میزبان بین کانتینرها هنوز هم یک چالش به شمار می‌آید. بر مبنای این دشواری در سال 2010 میلادی معماری یک راهیاب ترکیبی نوری – الکتریکی پیشنهاد شد که ارتباط بین MDC ها را امکان‌پذیر سازد.

هنگامی که از توپولوژی بهره‌مند از ارتباط‌های وسیع بهره‌گیری می‌شود یکی از موضوعات مهم طراحی الگوریتم‌های مسیریابی مابین راهیاب‌ها می‌باشد. یک راهکار استفاده از نوعی مسیریابی تصادفی است. امکان دیگر به کارگیری چندین کارت واسط شبکه در هر میزبان و اتصال هر میزبان به چندین راهیاب ارزان قیمت است که به میزبان‌ها اجازه‌ی مسیریابی هوشمند ترافیک بین سوئیچ‌ها را می‌دهد. اعمال تغییرات و یا توسعه‌ی هر یک از این رویکردها در مراکز داده‌ی فعلی در حال انجام است. نوآوری و تحولات خیره‌کننده‌ی روزمره در رابطه با موارد اظهار شده دور از انتظار نیست و هر روز بایستی در انتظار تحولی نو در این عرصه بود.

تمرین‌های فصل پنجم

۱- خدماتی که پروتکل‌های لایه‌ی پیوند برای پروتکل‌های لایه‌ی شبکه فراهم می‌آورند کدام است؟ کدام‌یک از این خدمات لایه‌ی پیوند، سرویس‌های مرتبط به خود را در IP و TCP دارند؟

۲- اگر همه‌ی پیوندهای اینترنت مجبور به تحویل مطمئن داده می‌شدند آیا بازهم به سرویس تحویل مطمئن TCP نیاز بود؟ اگر آری و یا خیر چرا؟

۳- پروتکل‌های انتقال پروانه (توکن) و نظرسنجی را در یک تعامل با آوردن یک مثال شبیه‌سازی تشریح می‌کنید.

۴- فرض کنید دو گره همزمان اقدام به ارسال یک بسته به طول L از طریق یک کانال فراپخش با سرعت R می‌کنند. اگر تأخیر انتشار بین این دو گره dprop فرض شود آیا در صورتی‌که شرط $d_{prop} < \frac{L}{R}$ برقرار باشد بازهم برخورد روی می‌دهد؟ اگر آری و یا خیر چرا؟

۵- در بخش ۵-۴ فهرستی از ویژگی‌های مطلوب یک کانال فراپخش عنوان شد. کدامیک از این مشخصات را پروتکل شیاردار ALOHA در اختیار دارد؟ کدام‌یک از این مشخصات را پروتکل شیاردار ALOHA در اختیار دارد؟ کدام‌یک از این ویژگی‌ها مربوط به انتقال توکن می‌باشد؟

۶- چرا پروتکل توکن رینگ در صورتی که وسعت شبکه LAN زیاد باشد ناکارآمد است؟

۷- برای شبکه‌ی شکل ۵-۱۹ مسیریاب دارای دو مدول ARP هریک با جدول ARP ویژه‌ی خود می‌باشند. آیا این امکان وجود دارد که یک آدرس MAC مشابه در هر دو جدول ظاهر شود؟

۸- طول و فضای هریک از آدرس‌های ذیل چه مقدار است؟

الف) آدرس MAC

ب) آدرس IPV4

ج) آدرس IPV6

۹- چرا درون یک فریم فراپخش ارسالی یک تقاضای ARP وجود دارد؟ علت پاسخ واکنش ARP که در درون یک قاب با آدرس MAC مقصد مشخصی ارسال می‌گردد، چیست؟ توضیح دهید.

۱۰- فرض کنید سه گره‌ی A، B و C از طریق کارت‌های شبکه‌ی خود به یک شبکه‌ی LAN فراپخش مشابه متصل شده‌اند. اگر A هزاران داده گرام IP به سوی B گسیل نماید که هریک از قاب‌های داده‌ی تشکیل شده از آن‌ها آدرس MAC مقصد B را در درون خود دارند. آیا کارت شبکه‌ی C، این قاب‌ها را پردازش می‌کند؟ اگر آری آیا C داده گرام‌های IP در درون این قاب‌ها را به لایه‌ی شبکه‌ی C تحویل می‌دهد؟ پاسخ شما در صورتیکه A قاب‌ها را با آدرس MAC فراپخش گسیل نماید چه خواهد بود؟

۱۱- در پروتکل CSMA/CD پس از برخورد پنجم (5 مرتبه برخورد) احتمال آن که یک گره K=4 را گزینش نماید چقدر است؟ نتیجه‌ی K=4 مربوط به چه مقدار تأخیر زمانی برروی یک اترنت با سرعت 10Mbps است؟

۱۲- مقایسه‌ای را بین ساختار قاب‌های 10BASE-T, 100BASE-T و اترنت گیگابیت انجام دهید. چه تفاوتی با یکدیگر دارند؟

۱۳- فرض کنید یک کارت شبکه‌ی با سرعت 10Mbps جریان بی‌نهایت زیادی از ارزش‌های 1 را با بهره‌گیری از رمزگذاری منچستر به درون یک کانال گسیل نماید. سیگنال خارج شده از کارت شبکه دارای چه تعداد انتقال در هر ثانیه است؟

۱۴- شکل ۵-۲۶ را درنظر بگیرید. چه تعداد زیرشبکه برمبنای آدرس‌دهی به روش بخش ۴-۸ وجود دارد؟

۱۵- فرض کنید بخش اطلاعات درون یک بسته به صورت الگوی: 1010101010101011 باشد. اگر از روش توازن زوج استفاده شود. ارزش دامنه‌ای که بیت های توازن را در حالت روش دوبعدی در خود نگهداری می‌کند چقدر است؟ با در نظر گرفتن این نکته که یک دامنه با حداقل طول کد مجموع مقابله استفاده شده است، پاسخ ارایه گردد.

۱۶- فرض کنید بخش اطلاعات یک بسته مانند D در شکل ۵-۴ شامل 10 بایت به گونه‌ی 8 بیت بدون علامت باینری (دودویی) برای ارزش‌های عددی از صفر تا 9 باشد. کد مجموع مقابله‌ی اینترنت این داده را محاسبه کنید؟

۱۷- با درنظر گرفتن پرسش ۱۶ اگر به جای ارزش‌های عددی از صفر تا 9 این 10 بایت شامل موارد ذیل باشند:

الف) بیان باینری اعداد از 1 تا 10

ب) حروف بزرگ از A تا J به صورت ASCII

ج) حروف کوچک از a تا j به صورت ASCII

مطلوبست محاسبه‌ی کد مجموع مقابله‌ی اینترنت هر کدام؟

۱۸- مثالی را غیر از شکل ۵-۶ نشان دهید که سنجش توازن دوبعدی می‌تواند خطای یک بیت را آشکارسازی و تصحیح نماید. مثالی را نشان دهید که خطای دو بیت را شناسایی کند اما تصحیح نکند.

۱۹- سه گره‌ی فعال A، B و C را که در حال رقابت با یکدیگر جهت دستیابی به یک کانال با استفاده از پروتکل شیاردار ALOHA می‌باشند در نظر بگیرید. اگر هر گره تعداد بی‌شماری از بسته‌ها را جهت ارسال داشته باشد و نیز هر گره تلاش نماید انتقال در هر حفره را با احتمال p انجام دهد، و شیارها به ترتیب از 1 به بعد شماره‌گذاری شده باشند (شیار نخست 1، شیار دوم 2، و غیره) آن‌گاه:

الف) احتمال آن که گره‌ی A نخستین مرتبه در شیار 4 موفق شود چقدر است؟

ب) احتمال آن که برخی از گره‌ها A، B و یا C در شیار 2 موفق شوند چقدر است؟

ج) احتمال آن که نخستین موفقیت در شیار 4 باشد، چیست؟

د) کارایی این سیستم سه گره‌ای چقدر است؟

۲۰- نمودار تابع p جهت تحلیل کارایی پروتکل‌های شیاردار ALOHA و خالص ALOHA را برای ارزش‌های مختلف N رسم نمائید:

الف) N 10

ب) N 25

ج) N 50

۲۱- یک کد مولد G به طول 4 بیت را در شکل ۵-۸ درنظر بگیرید. فرض کنید D دارای ارزش به

صورت: 10101010 می‌باشد. ارزش R چقدر است؟

۲۲- با توجه به پرسش ۲۱ مقدار R را برای هریک از مقادیر D به دست آورید؟

الف) D=100 10001

ب) D=10100011

ج) D=01010101

۲۳- یک کانال فراپخش با تعداد N گره و سرعت Rbps را درنظر بگیرید. فرض کنید کانال فراپخش از روش نظرسنجی (با یک گره‌ی اضافی نظرخواهی) جهت دستیابی چندگانه بهره می‌گیرد. فرض کنید مقدار زمان لازم از هنگامی که یک گره انتقال خود را کامل می‌کند تا آن که گره‌ی بعدی اجازه انتقال را به دست آورد dpoll باشد. فرض کنید در بین حلقه‌ی نظرسنجی یک گره مجاز به ارسال حداکثر Q بیت می‌باشد. حداکثر نهفتگی کانال فراپخش چقدر است؟

۲۴- یک شبکه‌ی اترنت 100BASE-T با سرعت 100Mbps که همه‌ی گره‌های آن مستقیماً به یک هاب متصل شده‌اند را درنظر بگیرید. برای دستیابی به کارایی 0.5 حداکثر فاصله‌ی بین هر گره و هاب چقدر است؟

اگر فرض شود طول قاب 64 بایت باشد و تکرار کننده‌ای نیز وجود ندارد، آیا این حداکثر فاصله این اطمینان را ایجاد می‌کند که یک گره‌ی گسیل کننده‌ی A قادر به تشخیص این که دیگر گره‌ها نیز انتقالی را همزمان با A انجام داده‌اند یا خیر، باشند؟ اگر آری و یا خیر چرا؟ حداکثر فاصله‌ی خود را با استاندارد واقعی 100Mbps چگونه مقایسه می‌کنید؟

۲۵- با یادآوری این نکته که در پروتکل CSMA/CD کارت شبکه به اندازه‌ی 512*K بیت زمان پس از یک برخورد منتظر می‌شود که ارزشی برای K تصادفی گزینش شود. برای K=100 چه مدت زمان کارت شبکه در انتظار بازگشت به مرحله 2 به ترتیب برای شبکه‌های اترنت با سرعت‌های 10Mbps و 100Mbps قرار می‌گیرد؟

۲۶- در بخش ۵-۴ توصیف کارایی شیاردار ALOHA آورده شده است. هدف این پرسش کامل نمودن این محاسبه‌ی کارایی است.

الف) با یادآوری این نکته که هرگاه N گره‌ی فعال وجود دارد، کارایی شیاردار ALOHA برابر $N.p(1-p)^{N-1}$ می‌باشد. ارزشی را برای p محاسبه کنید که عبارت را به حداکثر مقدار خود نزدیک کند.

ب) با بهره‌گیری از ارزش p در بخش (الف) کارایی شیاردار ALOHA را برای N از صفر تا بی‌نهایت به دست آورید. یادآوری می‌شود رابطه‌ی $(1-\frac{1}{N})^N$ هنگامی که N به سمت بی‌نهایت میل نماید برابر $\frac{1}{e}$ می‌باشد.

۲۷- نشان دهید حداکثر کارایی خالص ALOHA برابر $\frac{1}{2e}$ می‌باشد؟ این پرسش را با توجه به پرسش ۲۶ پاسخگویی نمائید.

۲۸- فرض کنید گره‌های A و B هر دو برروی یک شبکه‌ی اترنت گذرگاه 10Mbps قرار دارند و تأخیر انتشار بین دو گره نیز 325 بیت زمانی می‌باشد. اگر فرض شود گره‌ی A آغاز به انتشار یک قاب نماید و قبل از آن که عملیات انتقال خاتمه یابد گره‌ی B شروع به انتقال قاب نماید. آیا گره‌ی A می‌تواند انتقال خود را قبل از تشخیص انتقال B کامل نماید؟ اگر آری و یا خیر چرا؟

۲۹- در پرسش ۲۸ اگر پاسخ آری باشد آن‌گاه A به اشتباه باور کرده است که قاب داده‌ی او با موفقیت و بدون برخورد منتقل شده است. اگر فرض شود در زمان t=0 بیت، A آغاز به انتقال یک قاب نماید. در بدترین حالت، A قابی را انتقال می‌دهد که حداقل اندازه‌ی آن برابر $(64+512)$ بیت باشد. بنابراین A می تواند انتقال فریم را در فاصله‌ی $t=512+64$ بیت انجام دهد و درنتیجه پاسخ خیر است، اگر سیگنال B به A قبل از زمان معادل $t=512+64$ بیت برسد آنگاه در بدترین شرایط چه هنگام سیگنال B به A می‌رسد؟

۳۰- فرض کنید دو گره‌ی A و B در دو نقطه‌ی انتهایی یک کابل 900 متری می‌باشند و هریک نیز یک قاب 1000 بیت شامل دو بخش سرپیام و سرآغاز را به یکدیگر گسیل می‌کنند. هر دو گره تلاش در ارسال در زمان t=0 می‌کنند. اگر فرض شود تعداد 4 تکرار کننده بین A و B وجود دارد که هرکدام یک تأخیر معادل 20 بیت را تزریق می‌کنند، و سرعت انتقال نیز 10Mbps است و از پروتکل CSMA/CD با فواصل عقب‌نشینی مضارب 512 بیت استفاده شده است. پس از برخورد اول گره‌ی A K=0 و B مقدار K=1 را در پروتکل عقب‌نشینی تصادفی اختیار می‌کند. اگر از سیگنال انسداد و تأخیر 96 بیت زمانی صرف‌نظر شود، مطلوبست:

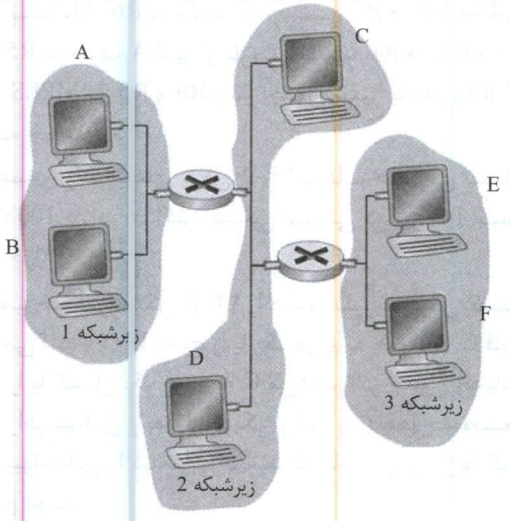
الف) تأخیر انتشار یک طرفه که شامل تأخیر تکرار کننده‌ها نیز می‌شود، بین A و B چقدر است؟ فرض کنید سرعت انتشار سیگنال $2\times10^8\,m/s$ است.

ب) در چه مدت زمان بسته‌ی A کاملاً تحویل B می‌شود؟

ج) اگر فرض شود فقط A مایل به ارسال بسته باشد و راهیاب‌ها جایگزین تکرارکننده‌ها شوند و هر راهیاب نیز تأخیر پردازشی معادل 20 بیت زمانی را علاوه بر زمان تأخیر ذخیره‌سازی و گسیل جلو داشته باشد. آن‌گاه چه مدت زمان لازم است تا بسته‌ی A به B تحویل گردد؟

۳۱- با توجه به شکل ۵-۲۶ فرض کنید که همه‌ی پیوندها 100Mbps می‌باشند. حداکثر نهفتگی انباشتی که می‌توان در بین این 14 سیستم انتهایی در این شبکه حاصل نمودچقدر است؟ چرا؟

۳۲- سه شبکه‌ی LAN مرتبط به یکدیگر از طریق دو مسیریاب بر مبنای شکل ۵-۴۰ درنظر بگیرید.



شکل ۵-۴۰ ♦ سه زیر شبکه‌ی متصل از طریق مسیریاب‌ها

الف) شکل را به گونه‌ای رسم نمائید که شامل کارت‌های شبکه گردد.

ب) آدرس‌های IP را به همه‌ی واسط‌های شبکه براساس این موارد: برای زیرشبکه‌ی شماره یک از آدرس‌های در قالب 111.111.111.xxx و برای زیرشبکه 2 از قالب 122.222.222.yyy و برای زیرشبکه‌ی 3 از آدرس‌های در قالب 133.133.133.zzz اختصاص دهید.

ج) آدرس‌های MAC همه‌ی کارت‌های شبکه را مشخص کنید.

د) حالتی را درنظر بگیرید که یک داده گرام IP از میزبان A به میزبان F گسیل می‌شود. فرض کنید همه‌ی جداول ARP به روزرسانی شده‌اند. شماره گذاری همه‌ی مراحل انجام شده در مثال بخش ۵-۵ که از یک روتر تک بهره می‌گرفت را مجدداً انجام دهید.

و) مرحله (د) را با این فرض تکرار نمائید که جدول ARP در میزبان فرستنده تهی است و بقیه‌ی جداول به روزرسانی شده‌اند.

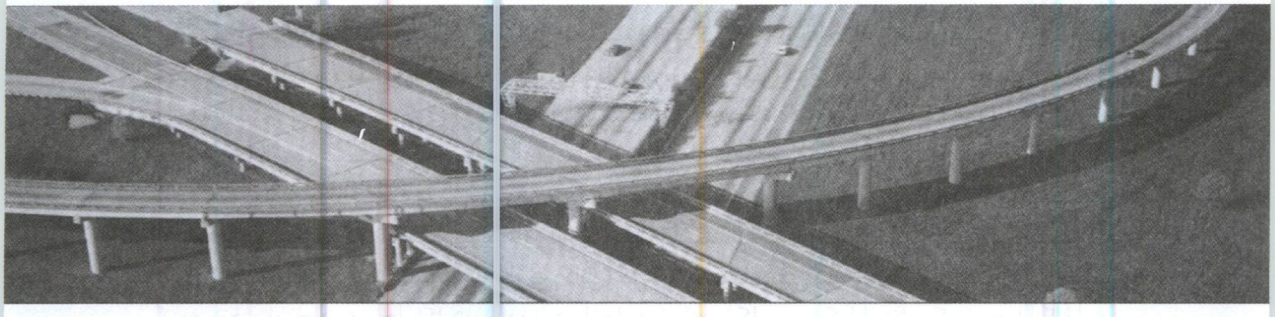
۳۳- مجدداً شکل ۵-۴۰ را در پرسش ۳۲ درنظر بگیرید. آدرس‌های IP و MAC را برای واسط‌های میزبان A، هر دو مسیریاب و میزبان F اختصاص دهید. فرض کنید میزبان A داده گرامی را به میزبان F گسیل می‌کند. مطلوب است آدرس‌های MAC مبدأ و مقصد در قاب داده‌ای که این داده گرام IP را بسته‌بندی می‌کند هنگامی که فریم ارسالی: (۱) از A به روتر چپ، (۲) از مسیریاب چپ به روتر سمت راست و (۳) از مسیریاب سمت راست به F منتقل می‌شود؟ هم‌چنین مطلوب است آدرس‌های IP مبدأ و مقصد در داده گرام IP بسته‌بندی شده در درون قاب در هریک از این نقاط در هر لحظه‌ی زمانی؟

۳۴- فرض کنید سه راهیاب شکل ۵-۲۶ با هاب جایگزین شوند. اگر همه‌ی پیوندها سرعت 100Mbps اختیار نمایند. حداکثر نهفتگی انباشتی که در بین این 14 سیستم انتهایی از این شبکه می‌توان نتیجه گرفت چقدر است؟ چرا؟

۳۵- شبکه‌ی MPLS شکل ۵-۳۷ را درنظر بگیرید و فرض کنید که مسیریاب‌های R5 و R6 از نوع مجهز به MPLS می‌باشند. اگر مایل به انجام مهندسی ترافیک در آن‌ها به‌گونه‌ای می‌باشید که بسته‌ها از R6 به مقصد A از مسیر $R6 \rightarrow R4 \rightarrow R3 \rightarrow R1$ راهیابی شوند و بسته‌های از R5 به طرف A نیز از طریق مسیر $R5 \rightarrow R4 \rightarrow R2 \rightarrow R1$ ، آن‌گاه نشان دهید جدول MPLS در R5 و R6 و نیز جدول تغییر یافته‌ی R4 که این موضوع را امکان‌پذیر می‌کند چگونه خواهد بود؟

۳۶- فرض کنید همه‌ی راهیاب‌های شکل ۵-۲۶ با هاب جایگزین شده‌اند و همه‌ی پیوندها نیز سرعت 100Mbps دارند. حداکثر نهفتگی انباشتی در بین 14 سیستم انتهایی در این حالت چقدر است؟ چرا؟

عملیات راهیاب خودآموز شکل ۵-۲۴ را درنظر بگیرید. فرض کنید که (۱) گره‌ی A قابی را به D گسیل می‌کند، (۲) D پاسخ را با یک فریم به A و (۳) C قابی را به D ارسال می‌کند و (۴) D پاسخ را با گسیل یک فریم به C می‌دهد. اگر جدول راهیاب در آغاز تهی باشد. نشان دهید جدول راهیاب قبل و بعد از هریک از این رخدادها چه وضعیتی دارد؟ برای هریک از این حالت‌ها پیوندهایی را شناسایی نمائید که قاب برروی آن‌ها گسیل می‌شود. علت پاسخ خود را توضیح دهید.



فصل ۶

شبکه‌های سیّار و بی‌سیم

در دنیای ارتباطات تلفنی، پانزده سال گذشته دوران طلایی ارتباط تلفنی سلولی یا سیار بوده است. تعداد مشترکین سلولی سیار در جهان از رقم 34 میلیون در سال 1993 میلادی به رقم 5.5 میلیارد مشترک در انتهای سال 2008 میلادی رسید. این ارقام نشان می‌دهد بر اساس آمار گزارش شده از سوی نهاد ITU در سال 2012 میلادی تعداد مشترکین سلولی سیار در حال حاضر گوی سبقت را از آنچه اصطلاحاً خطوط تلفن ثابت نامیده می‌شود، ربوده است و این سیر فزاینده ادامه دارد. برجستگی‌های بی‌شماری را می‌توان برای ارتباطات سلولی سیار برشمرد. در هر نقطه و هر زمان تقریباً بدون هیچ‌گونه مانعی دستیابی به شبکه‌ی ارتباطات تلفنی جهانی از طریق تجهیزات کم‌وزن، کوچک و بسیار ساده از لحاظ حمل و نقل امکان‌پذیر می‌باشد. با شتاب در نوآوری کامپیوترهای همراه، تبلت‌ها، Palmtops و امکان دستیابی هرکجا، هر لحظه و تقریباً بدون مانع آن‌ها به شبکه‌ی جهانی اینترنت انفجار مشابهی نیز در بهره‌وری از تجهیزات سیار بدون سیم اینترنت در سراسر گیتی دور از انتظار نمی‌باشد.

بدون توجه به رشد آتی تجهیزات بدون سیم اینترنت، از هم اکنون آشکار است که شبکه‌های بی‌سیم و خدمات سیار وابسته به آن‌ها آمده‌اند که برای مدت درازی باقی بمانند. از نقطه‌نظر شبکه‌سازی چالش پیش روی از جانب این شبکه‌ها مربوط به لایه‌های شبکه و پیوند داده می‌باشد. زیرا از این لحاظ شبکه‌های بی‌سیم کامپیوتری کاملاً متفاوت از شبکه‌های به اصطلاح ثابت کامپیوتری می‌باشند.

آغاز مطالب این فصل بررسی کاربران سیار، پیوندهای بی‌سیم، شبکه‌ها و نیز ارتباط آن‌ها با شبکه‌های بزرگ‌تر و گسترده‌تر ثابت کامپیوتری است که این‌گونه کاربران سرانجام بایستی به آن‌ها متصل گردند. آنچه حائز اهمیت است تفاوت و تمایز اساسی بین چالش‌های پیش‌رو از سوی طبیعت پیوندهای ارتباطی **بی‌سیم (wireless)** در چنین شبکه‌هایی و قدرت **تحرک (mobility)** پیوندهای

ارتباطی آن‌ها می‌باشد. این تمایز مهم بین بی‌سیم و تحرک به ما این اجازه را می‌دهد که جداسازی بهینه، شناسایی و مهارت را از اصول کلیدی موارد مربوط به این موضوعات در هر بخش در اختیار داشته باشیم. یقیناً در بسیاری از محیط‌های شبکه‌سازی شده، گره‌های شبکه در آن‌ها بی‌سیم می‌باشند اما سیّار نیستند. از جمله‌ی این موارد می‌توان از شبکه‌های خانگی و یا اداری نام برد که تجهیزات کاری و یا صفحه‌های نمایشگر بزرگ ثابت در اختیار دارند. بدیهی است الگوهای محدودی از تحرک وجود دارند که نیازمند پیوندهای بی‌سیم نمی‌باشند. از جمله‌ی این موارد کاربری است که از یک کامپیوتر همراه ثابت (متصل از طریق کابل) در خانه استفاده می‌کند و ممکن است آن را خاموش کرده و همراه خود به محل کارش برده و در آنجا به شبکه‌ی کامپیوتر کابلی شرکت محل کار خود مجدداً متصل نماید. البته بسیاری از شبکه‌های کامپیوتری در حال حاضر محیطی را فراهم می‌آورند که در آن کاربران هر دو حالت **بی‌سیم** و **متحرک** را می‌توانند اختیار کنند. از جمله مواردی که می‌توان در این رابطه در ذهن مجسم نمود، هنگامی است که یک کاربر سیّار (مانند شخصی که در صندلی عقب خودرو نشسته است) در حال انجام یک مکالمه‌ی تلفنی از طریق IP یا همان ارتباط VOIP می‌باشد که شامل چندین ارتباط TCP خروجی است در حالی که با سرعت 160 کیلومتر در ساعت از درون آزادراهی در حرکت است. در این نقطه که محل تقاطع بی‌سیم و تحرک (بدون سیم و مرتباً از نقطه‌ای به نقطه‌ی دیگر در فواصل کوتاه زمانی در حرکت بودن) است، چالش‌های بزرگ فنی و سیاست‌گذاری علمی بسیار هیجان‌انگیز پدید می‌آید!

ابتدا زمینه‌ای را که در آن ارتباطات بی‌سیم و متحرک موجود می‌باشد بررسی می‌کنیم. یک شبکه که در آن کاربران بی‌سیم و احتمالاً به صورت متحرک به یک زیرساخت شبکه‌ی بزرگ‌تر از طریق یک پیوند بی‌سیم در لبه یا نبش شبکه متصل شده‌اند. سپس مشخصات این پیوند بی‌سیم بررسی می‌شود و آنگاه به اختصار دست‌یابی چندگانه‌ی تقسیم رمز (CDMA) معرفی می‌شود. این پروتکل دست‌یابی، یک رسانه‌ی دست‌یابی مشترک که اغلب در شبکه‌های بی‌سیم مورد بهره‌برداری قرار می‌گیرد را معرفی می‌کند. در بخش‌های بعدی فصل موضوعات مربوط به پیوندهای ارتباطی استاندارد شبکه‌های LAN بی‌سیم (IEEE 802.11) بررسی عمیق می‌شوند و علاوه بر آن دو فناوری دیگر بلوتوث و WIMAX تحلیل می‌شوند. دست‌یابی اینترنت سلولی و فناوری در حال ظهور سلولی 3G و 4G که هر دو امکان صوت و دست‌یابی سریع اینترنت را فراهم می‌آورند، مورد بررسی و تحلیل قرار می‌گیرد. در بخش‌های انتهایی فصل تمرکز بر روی موضوعات مربوط به تحرک و به ویژه چالش مکان‌یابی یک کاربر متحرک، مسیریابی یک کاربر سیار و مدیریت کاربر خواهد بود. چگونگی پیاده‌سازی خدمات سیار استاندارد IP و GSM از جمله‌ی موارد مهم دیگری است که تحلیل می‌شوند. سرانجام تأثیر پیوندهای سیار و عامل تحرک بر روی پروتکل‌های لایه‌ی انتقال و کاربردهای شبکه بررسی می‌شوند.

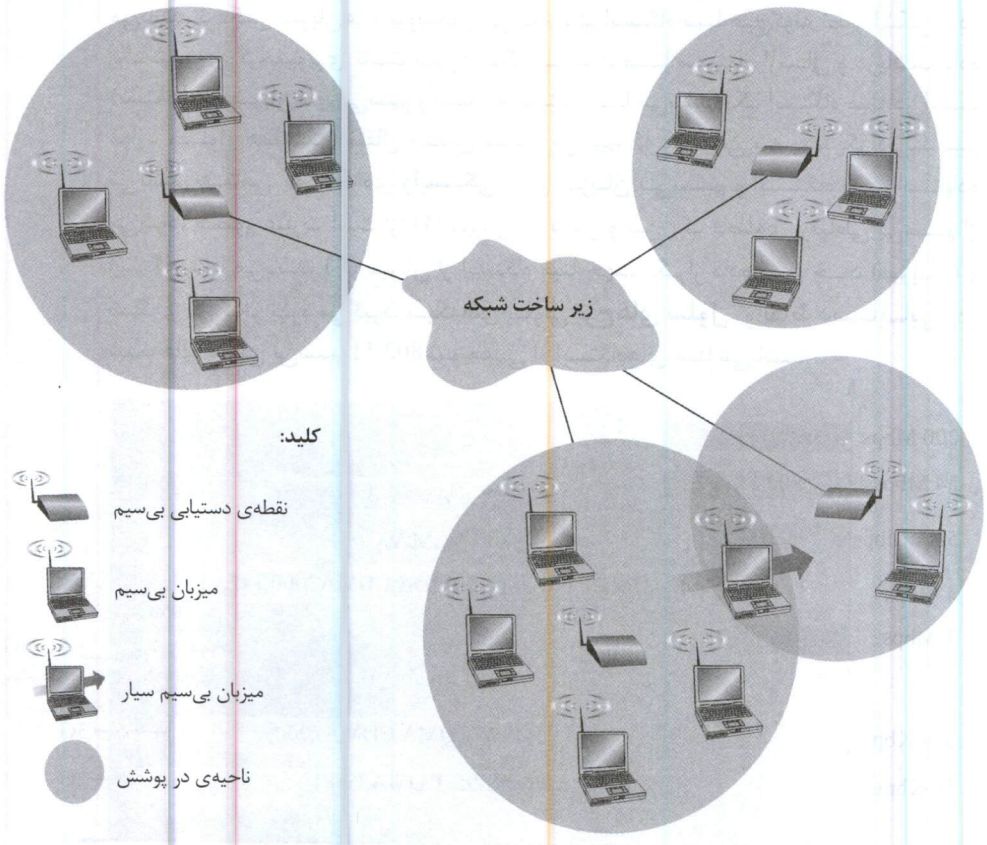
۶-۱ ارتباطات بی‌سیم و تحرک

شکل ۶-۱ زمینه را برای بررسی موضوع انتقال داده‌ی بی‌سیم و تحرک آماده می‌کند. در آغاز بحث در اطراف این موضوع، در قالب کلی آن چه پوشش گسترده‌ای از شبکه‌های مختلف شامل LAN بی‌سیم مانند IEEE 802.11 و سلولی مانند 3G می‌باشد، را دنبال می‌کنیم. در بخش‌های بعدی کاوش نسبتاً عمیق معماری بی‌سیم انجام خواهد شد. اما قبل از هر مطلبی ابتدا به شناسایی عناصر یک بی‌سیم در ذیل می‌پردازیم:

- **میزبان‌های بی‌سیم.** همانند حالت شبکه‌های کامپیوتری ثابت، میزبان‌ها تجهیزات

سیستم‌های انتهایی می‌باشند که برنامه‌های کاربردی را به اجرا درمی‌آورند. یک **میزبان بی‌سیم** ممکن است یک لپ‌تاپ، پالم‌تاپ، تبلت‌ها، گوشی تلفن و کامپیوتر PC باشد. میزبان‌ها به خودی خود ممکن است ثابت و یا سیار باشند.

- **پیوندهای بی‌سیم.** یک میزبان به ایستگاه مبنای خود که در ادامه تعریف می‌شود و یا به یک میزبان بی‌سیم دیگر از طریق یک **پیوند ارتباطی بی‌سیم** متصل می‌شود. فناوری‌های گوناگون پیوند بی‌سیم دارای سرعت‌های مختلف انتقال می‌باشند و بُرد آن‌ها نیز در فواصل گوناگون انجام می‌شود.

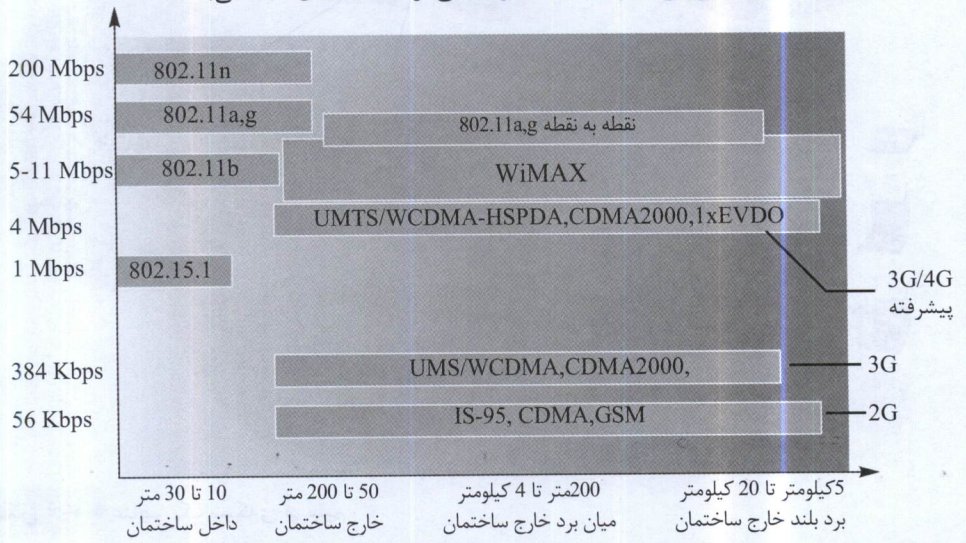


شکل ۶-۱ ♦ عناصر یک شبکه‌ی بی‌سیم

شکل ۶-۲ دو مشخصه‌ی کلیدی **ناحیه‌ی در پوشش** و **سرعت پیوند** را برای استانداردهای شبکه‌های بی‌سیم پرطرفدار نشان می‌دهد. این شکل فقط جهت بیان کلی این مشخصات می‌باشد. به عنوان مثال برخی از انواع شبکه‌ها در حال حاضر مورد بهره‌برداری قرار می‌گیرند و بعضی از سرعت‌های مربوط به پیوند نیز می‌تواند در آن‌ها افزایش و یا کاهش مقداری خارج از ارقام نشان داده شده را بستگی به مسافت، شرایط کانال و تعداد کاربران موجود در شبکه‌ی بی‌سیم اختیار نماید. این استانداردها در نیمه اول فصل ۷ تحلیل خواهند شد. علاوه بر این‌ها مشخصات دیگر پیوندهای بی‌سیم مانند نرخ خطای بیت و علل وقوع خطای بیت نیز در

بخش ۶-۲ تحلیل و بررسی می‌شوند. در شکل ۶-۱ پیوندهای بی‌سیم ارتباط میزبان‌های بی‌سیم ساکن در لبه‌ی شبکه را با زیرساخت یک شبکه‌ی بزرگ‌تر برقرار می‌کنند. یادآوری این نکته نیز دور از انتظار نیست که پیوندهای بی‌سیم بعضی از اوقات در درون یک شبکه جهت ارتباط روترها راهیاب‌ها و دیگر تجهیزات ارتباطی شبکه نیز به کار گرفته می‌شوند. در هر حال تمرکز در این فصل بر کاربرد ارتباط‌های بی‌سیم در اطراف لبه‌های شبکه می‌باشد زیرا چالش‌های عمده‌ی فنی بیشتر در این نقاط خودنمایی می‌کنند و علاوه بر آن بیشترین توسعه و گسترش شبکه‌های بی‌سیم نیز در این نقاط قرار دارد.

- **ایستگاه مبنا.** یکی از بخش‌های کلیدی زیرساخت شبکه‌ی بی‌سیم **ایستگاه مبنا (BS)** می‌باشد. برعکس میزبان‌ها و پیوندهای بی‌سیم یک ایستگاه مبنا هیچ‌گونه نظیر آشکاری در یک شبکه‌ی کامپیوتری ثابت ندارد. یک ایستگاه مبنا مسئول ارسال و دریافت داده (بسته‌ها) به میزبان‌های بی‌سیم وابسته به ایستگاه مبنا می‌باشد. یک ایستگاه مبنا در اغلب موارد مسئول هماهنگی انتقال چندین میزبان بی‌سیم که وابستگی به یکدیگر دارند نیز می‌باشد. هنگامی که از واژه‌ی **وابستگی** بین میزبان بی‌سیم و ایستگاه مبنا استفاده می‌شود، منظور عبارت است از: **(۱)** میزبان در درون و بین یک فاصله‌ی ارتباطی بی‌سیم از ایستگاه مبنا می‌باشد، **(۲)** میزبان از ایستگاه مبنا جهت تکرار داده بین خود (میزبان) و شبکه‌ی بزرگ‌تر بهره می‌گیرد. شبکه‌های سلولی **برج‌های سلول**[1] و **نقاط دست‌یابی**[2] در شبکه‌های محلی بی‌سیم 802.11 نمونه‌هایی از ایستگاه‌های مبنا می‌باشند.



شکل ۶-۲ ♦ مشخصات پیوند برخی از استانداردهای شبکه‌ی بی‌سیم

در شکل ۶-۱ ایستگاه مبنا متصل به شبکه‌ای بزرگ‌تر مانند اینترنت، شرکت، شبکه‌ی تلفنی و یا شبکه‌ی خانگی می‌باشد. بنابراین به عنوان عامل تکرار لایه‌ی پیوند بین میزبان بی‌سیم و مابقی دنیایی

1- Cell towers
2- Access points

که میزبان با آن ارتباط برقرار می‌کند، می‌باشد. میزبان‌های وابسته به ایستگاه مبنا در اغلب موارد گفته می‌شود اصطلاحاً به **روش زیرساخت** عمل می‌کنند، زیرا همه‌ی خدمات شبکه‌های سنتی مانند اختصاص آدرس و مسیریابی از جانب شبکه‌ای که میزبان به آن از طریق ایستگاه مبنا متصل شده است را تامین می‌کنند. در غیبت چنین زیرساختی، میزبان‌ها بایستی خدمات لازم مانند مسیریابی، اختصاص آدرس، ترجمه‌ی نام DNS و غیره را نیز فراهم آورند.

هنگامی که یک میزبان سیّار از محدوده‌ی یک ایستگاه مبنا خارج می‌شود و وارد محدوده‌ی ایستگاه دیگری می‌گردد، آن میزبان نقطه‌ی وابستگی خود را به شبکه‌ی بزرگ‌تر تغییر می‌دهد. این فرآیند اصطلاحاً **جابه‌جایی**[1] نامیده می‌شود. چنین تحرک بسیاری از پرسش‌های چالش برانگیزی را مطرح می‌نماید که از جمله‌ی آن‌ها می‌توان به موارد ذیل اشاره نمود:

۱- اگر یک میزبان قادر به جابه‌جایی باشد چگونه می‌توان مکان جاری میزبان سیار در شبکه را به گونه‌ای شناسایی نمود تا داده‌ی ارسالی به آن میزبان سیار تحویل گردد؟

۲- آدرس‌دهی چگونه انجام می‌شود، با توجه به این نکته که یک میزبان می‌تواند در بسیاری از مکان‌های ممکن باشد؟

۳- اگر میزبان در هنگام انجام یک ارتباط TCP و مکالمه‌ی تلفنی حرکت نماید و تغییر مکان دهد چگونه داده‌ها مسیریابی می‌شوند تا وقفه‌ای در ارتباط حاصل نگردد؟

- **زیرساخت شبکه.** این یک شبکه‌ی بزرگ‌تر را که یک میزبان بی‌سیم مایل به ارتباط با آن می‌باشد معرفی می‌کند. با توجه به توضیحات ارایه شده پیرامون عناصر یک شبکه‌ی بی‌سیم، ملاحظه می‌شود این قطعه‌ها را می‌توان به صورت‌های گوناگونی از شبکه‌های بی‌سیم ترکیب نمود. در بالاترین سطح تقسیم‌بندی، شبکه‌های بی‌سیم را می‌توان بر مبنای دو فرضیه در نظر گرفت. (۱) نخست آن که یک بسته در شبکه‌ی بی‌سیم دقیقاً از یک یا چندین نقطه‌ی پرش کوتاه (hop) عبور می‌کند و (۲) وجود یک زیرساخت مانند یک ایستگاه مبنا در شبکه. این موارد دوگانه را می‌توان چنین توصیف نمود:
- **زیرساخت برمبنای یک نقطه‌ی پرش کوتاه (hop).** این شبکه‌ها دارای یک ایستگاه مبنا می‌باشند که به یک شبکه‌ی بزرگ‌تر کامپیوتری ثابت مانند اینترنت متصل شده‌اند. علاوه بر آن همه‌ی ارتباط‌ها بین ایستگاه مبنا و یک میزبان بی‌سیم فقط از طریق یک نقطه پرش کوتاه می‌باشد. شبکه‌های 802.11 در کارگاه‌ها، کتابخانه‌ها، شبکه‌های تلفنی سلولی و شبکه‌های WIMAX 802.11 همگی در این گروه از زیرساخت قرار می‌گیرند.
- **بدون زیرساخت برمبنای یک نقطه‌ی پرش کوتاه (hop).** در این شبکه‌ها، ایستگاه مبنایی وجود ندارد که به یک شبکه‌ی بی‌سیم متصل باشد. ملاحظه خواهد شد یکی از گره‌ها در این شبکه‌ی تک‌هاپ هماهنگی انتقال دیگر گره‌ها را بر عهده می‌گیرد. شبکه‌های بلوتوث و 802.11 که اصطلاحاً **روش ویژه‌ی روزمره (ad – hoc)** نامیده می‌شوند از این نوع شبکه‌ها به شمار می‌آیند.
- **زیرساخت برمبنای چند پرش کوتاه (multihop).** در این شبکه‌ها، یک ایستگاه مبنا وجود دارد که به یک شبکه‌ی بزرگ‌تر به صورت کابلی متصل می‌باشد. برخی از گره‌های

1- Hand off

بی‌سیم، ممکن است مجبور باشند ارتباط‌های خود با یکدیگر را از درون گره‌های بی‌سیم دیگری با بهره‌گیری از ایستگاه مبنا انجام دهند. برخی **از شبکه‌های بی‌سیم حسگر**[1] و آنچه که به **شبکه‌های بی‌سیم راه‌یابی مستقل**[2] معروف می‌باشند از این جمله موارد می‌باشند.

- **بدون زیرساخت چندنقطه‌ی پرش کوتاه (multi hop)**. در این نوع شبکه‌ها ایستگاه مبنا وجود ندارد و گره‌ها ممکن است مجبور باشند پیام‌ها را بین چندین گره‌ی دیگر تکرار نمایند تا سرانجام پیام به مقصد مورد نظر تحویل گردد. گره‌ها ممکن است سیار و ارتباط بین گره‌ها نیز متغیر باشد. یک گروه از شبکه‌ها که اصطلاحاً به آن‌ها **شبکه‌های سیار مستقل**[3] **(MANET)** گفته می‌شود از این جمله می‌باشند. اگر گره‌های سیار در خودروها می‌باشند آنگاه شبکه را اصطلاحاً **شبکه‌ی مستقل خودرویی**[4] **(VANET)** می‌نامند. بدیهی است توسعه و تکمیل پروتکل‌های چنین شبکه‌هایی چالش برانگیز است و نیازمند پژوهش‌های فراوانی نیز می‌باشد.

در این فصل تمرکز بر روی شبکه‌های تک پرش‌کوتاه می‌باشد و پس از آن شبکه‌های بر مبنای زیرساخت تحلیل می‌شوند. در ادامه پیوندهای انحصاری بی‌سیم تجزیه و تحلیل می‌شوند.

۶-۲ ویژگی‌های پیوندهای بی‌سیم شبکه

گام نخست در بررسی مطالب مربوط به این زیربخش را با تحلیل یک شبکه‌ی کابلی ساده، مانند یک شبکه‌ی خانگی به همراه یک راه یاب اترنت ثابت که از طریق کابل به میزبان‌ها متصل شده است، آغاز می‌کنیم. اگر این شبکه‌ی کابلی اترنت را با رقیب بدون کابل (بی‌سیم) آن یعنی یک شبکه‌ی 802.11 جایگزین نمائیم. آنگاه یک کارت شبکه‌ی بی‌سیم نیز جایگزین کارت‌های اترنت در میزبان‌ها می‌شود و جای راه‌یاب را نیز یک نقطه‌ی دست‌یابی بی‌سیم خواهد گرفت. اما نکته‌ی جالب توجه این حقیقت است که تقریباً نیازی به هیچ‌گونه تغییری در لایه‌های شبکه و بالاتر از آن نمی‌باشد. این نکته به مفهوم آن است که فقط نیازمند تمرکز بر روی لایه‌ی پیوند می‌باشیم در آن هنگام که به جستجوی تفاوت‌های بین شبکه‌های باسیم و بدون سیم هستیم، یقیناً چندین تفاوت مهم بین پیوندهای کابلی و اصطلاحاً رادیویی یا بی‌سیم را می‌توان شناسایی نمود:

- **کاهش قدرت سیگنال**. امواج و پرتوهای الکترومغناطیسی به هنگام عبور از ماده دچار تضعیف می‌شوند. به عنوان مثال یک سیگنال رادیویی در قالب یک پرتو الکترو مغناطیس هنگام عبور از دیوار دچار تضعیف می‌شود. حتی در فضای آزاد یا خلاء، سیگنال‌های الکترومغناطیسی دچار پراکندگی و در نتیجه کاهش قدرت سیگنال نیز می‌شوند. تضعیف سیگنال در اثر پراکنش را اصطلاحاً **افت مسیر**[5] می‌نامند و مقدار تضعیف قدرت سیگنال بستگی بسیار نزدیک و مستقیم به فاصله‌ی بین فرستنده و گیرنده دارد.

1- Wireless sensor networks
2-Wireless mesh networks
3- Mobile ad hoc networks
4- Vehicular ad hoc networks
5- Path loss

- **تأثیر تداخل منابع دیگر**. منابع رادیویی گوناگون دیگری در یک پهنای باند فرکانس مشابه در حال انتقال سیگنال‌های اطلاعاتی می‌باشند، همگی آن‌ها سبب تاثیرگذاری بر یکدیگر می‌شوند. به عنوان مثال یک گوشی تلفن بی‌سیم با پهنای باند 2.4 GHz بر روی سیگنال‌های رادیویی یک شبکه‌ی LAN بی‌سیم با مشخصات 802.11b تاثیر می‌گذارد زیرا پهنای باند این شبکه نیز 2.4 GHz می‌باشد. بنابراین دور از انتظار نیست هنگامی که یک کاربر شبکه 802.11b در حال مکالمه با یک گوشی بی‌سیم 2.4 GHz است، آگاه شود که هر دو امکانات او یعنی شبکه و تلفن عملکرد مفید همگشی خود را ندارند. علاوه بر تداخل ایجاد شده از سوی منابع رادیویی در حال انتقال، اصطلاحاً نویز الکترومغناطیسی حاصل از محیط مانند یک موتور در حال کار، یک وسیله‌ی الکتریکی و یا یک وسیله‌ی مایکروویو نیز می‌تواند باعث ایجاد اختلال در سیگنال‌ها شود.
- **انتشار چند مسیری سیگنال**. آنچه در واژگان شبکه پراکندگی سیگنال و یا **انتشار چندمسیری**[1] نامیده شده است، هنگامی به وقوع می‌پیوندد که بخشی از پرتو الکترومغناطیسی از طریق تماس با اشیاء و یا سطح زمین منعکس شده و مسیرهای گوناگونی را از لحاظ طول یا مسافت بین فرستنده و گیرنده اختیار نماید. این سبب تضعیف و به اصطلاح تارشدن سیگنال ورودی به گیرنده می‌شود. جابه‌جایی اشیاء بین فرستنده و گیرنده سبب ایجاد انتشار چند مسیری می‌شود، که در گذر زمان حالت متغیر نیز به خود می‌گیرد.

آنچه از مطالب بالا می‌توان نتیجه‌گیری نمود، این حقیقت است که در اثر این کنش و واکنش‌ها خطای بیت در اطلاعات بر روی پیوندهای بی‌سیم به مراتب چشمگیرتر از پیوندهای کابلی است. به همین علت است که پروتکل‌های پیوندهای بی‌سیم مانند 802.11 نه تنها از کدهای آشکارسازی بسیار قدرتمند از نوع CRC بهره می‌گیرند بلکه از پروتکل‌های انتقال داده‌ی مطمئن در سطح لایه پیوند نیز جهت انتقال مجدد قاب‌های آلوده شده، استفاده می‌کنند.

پس از بررسی اختصاری اختلال‌های به وجود آمده بر روی یک کانال بی‌سیم یا رادیویی، اکنون توجه خود را به سوی میزبان دریافت کننده‌ی سیگنال بی‌سیم معطوف می‌کنیم. یک میزبان بی‌سیم گیرنده اطلاعات، سیگنال الکترومغناطیسی را دریافت می‌دارد که الگوی تضعیف شده‌ای از سیگنال اصلی ارسالی از سوی فرستنده است. تضعیف به واسطه‌ی عوامل گوناگون و نیز تاثیر انتشار چندمسیری است که به آن نویز حاصل از عوامل موجود در محیط نیز اضافه شده است. معمولاً برای نشان دادن این تاثیرها بر روی عملکرد سیگنال واقعی اصطلاحاً از **نسبت قدرت سیگنال به نویز**[2] **(SNR)** استفاده می‌شود. این ارزش یک اندازه‌گیری نسبی از قدرت سیگنال دریافتی یا همان اطلاعات به قدرت نویز می‌باشد. معمولاً SNR اندازه‌گیری شده با یکای دسی‌بل **(dB)** بیان می‌شود. ارزش SNR اندازه‌گیری شده در مقیاس dB عبارت است از 20 برابر لگاریتم در مبنای 10 نسبت دامنه‌ی سیگنال دریافتی به دامنه‌ی نویز. اگر فرض شود **دامنه‌های** سیگنال و نویز به ترتیب S_{amp} و N_{amp} باشند آنگاه SNR اندازه‌گیری شده از طریق رابطه‌ی (۶-۱) محاسبه می‌شود.

1- Multipath propagation
2- Signal to noise ratio

$$SNR = 20\log_{10}\frac{S_{amp}}{N_{amp}} \qquad (۱-۶)$$

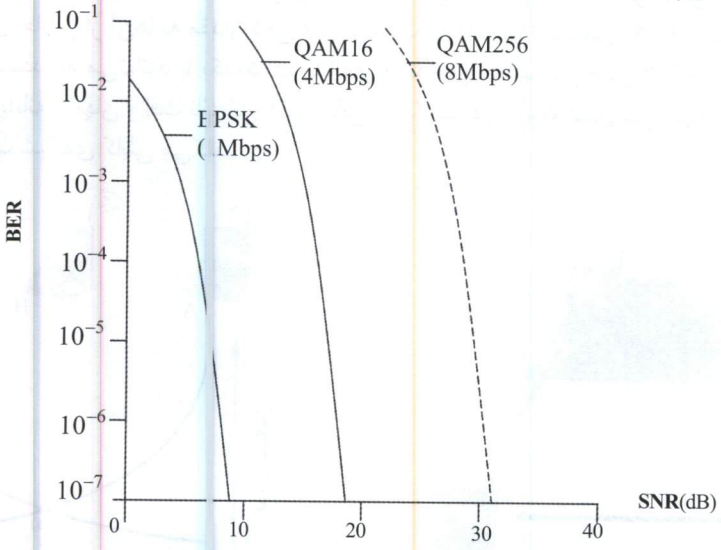
از نقطه نظر هدفی که دنبال می‌کنیم، آنچه مهم است این نکته می‌باشد که هر چه مقدار SNR بیشتر و بزرگ‌تر باشد مفهوم آن سهولت در بازیابی سیگنال دریافتی از سوی گیرنده در حضور نویز می‌باشد.

شکل ۶-۳ **نرخ خطای بیت**[1] **(BER)** را نشان می‌دهد که عبارت است از احتمال خطای بیت بر روی بیت دریافتی در گیرنده در برابر SNR. ارزش‌هایی که این شکل نشان می‌دهد مربوط به سه روش گوناگون رمزگذاری اطلاعات ارسالی بر روی یک کانال بی‌سیم ایده‌آل است. در این زیربخش به دنبال بررسی نظریه‌های رمزگذاری و مدولاسیون و نیز بازیابی سیگنال و BER نمی‌باشیم. خوانندگان در صورت علاقه‌مندی می‌توانند برای مطالعات جزئیات کامل این موراد به مرجع [Schwartz 1980] مراجعه نمایند. اما در هر حال شکل ۶-۳ چندین مشخصه از لایه‌ی فیزیکی را توصیف می‌کند که جهت درک پروتکل‌های ارتباطی بی‌سیم لایه‌های بالاتر ضروری می‌باشند. این ویژگی‌ها عبارتند از:

- **در یک روش مدولاسیون، نسبت ارزش‌های SNR و BER عکس یکدیگر می‌باشند.** از آنجائیکه فرستنده می‌تواند با تقویت قدرت انتقال خود ارزش SNR را افزایش دهد، در نتیجه فرستنده احتمال خطای بیت در یک قاب داده‌ی دریافتی را از طریق تقویت قدرت انتقال کاهش می‌دهد. بدیهی است در عمل بهره‌ی چندانی از افزایش قدرت انتقال از سوی یک فرستنده فراتر از یک آستانه‌ی ویژه حاصل نمی‌شود. به عنوان مثال آنچه ممکن است در نهایت از این افزایش حاصل شود، کاهش احتمال وقوع خطای بیت (BER) از مقدار برابر 10^{-12} به 10^{-13} می‌باشد. یادآوری این نکته ضروری است که افزایش قدرت انتقال به مفهوم مصرف انرژی بیشتر از سوی فرستنده است و این مهم به ویژه در ابزار کاربردی سیار که از نیروی باطری تغذیه می‌کنند بسیار با اهمیت و نگران‌کننده می‌باشد. علاوه بر آن انتقال سیگنال‌های فرستنده با قدرت افزونتر سبب ایجاد اختلال در دیگران خواهد شد. شکل ۶-۴ (ب) اهمیت این موضوع را به تصویر کشیده است.
- **مدولاسیون با سرعت انتقال بیت بالاتر برای یک SNR ویژه معادل است با افزایش نرخ BER (چه در حالت خطا و یا غیر آن).** به عنوان مثال در شکل ۶-۳ اگر SNR=10dB با روش مدولاسیون **(BPSK)**[2] و سرعت انتقال 1Mbps در نظر گرفته شود ملاحظه می‌شود که نرخ BER در این حالت برابر است با 10^{-7}، در حالی که با روش مدولاسیون **(QAM16)**[3] و نرخ انتقال 4 Mbps نرخ BER برابر است با 10^{-1}، رقمی به مراتب بالاتر که عملاً پذیرش آن غیرممکن است. اما SNR=20dB و روش مدولاسیون QAM16 با سرعت 4 Mbps نرخ BER برابر 10^{-7}را عرضه می‌دارد در عین حالی که روش BPSK با نرخ 1 Mbps نرخ BER بسیار پائین را نتیجه می‌دهد که در نمودار نمی‌توان نقطه‌ای را برای آن در نظر گرفت. اگر نرخ BER برابر 10^{-7} قابل تحمل باشد آنگاه سرعت

1- Bit error rate
2- Binary phase shift keying
3- Quadrature amplitude modulation

انتقال به مراتب بالاتر QAM16 در این شرایط، روش مدولاسیون ترجیحی خواهد بود. رعایت این مهم ما را به مشخصه‌ی بعدی هدایت می‌کند.



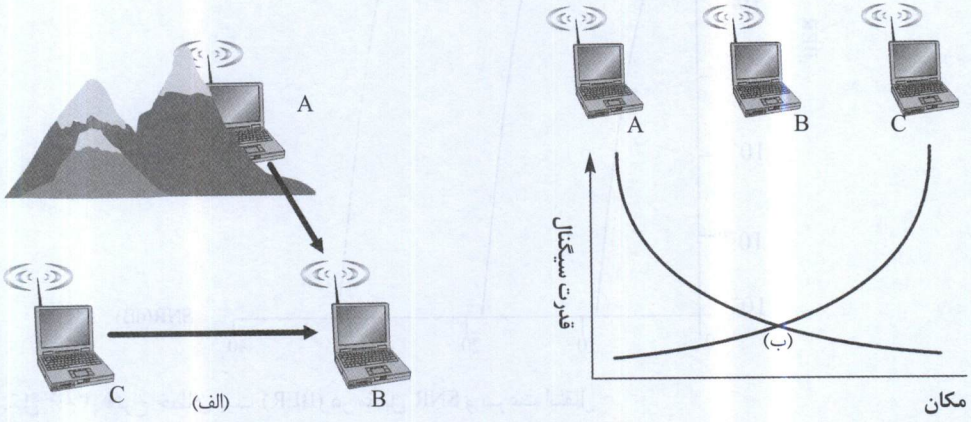
شکل ۶-۳ ♦ نرخ خطای بیت (BER) در مقابل SNR و سرعت انتقال

- **گزینش پویای روش مدولاسیون لایه‌ی فیزیکی ممکن است باعث سازگاری آن با شرایط کانال شود.** ارزش‌های SNR و بنابراین BER ممکن است در اثر جابه‌جایی و یا تغییرات محیط عملکرد کاربران تغییر نماید. از روش‌های رمزگذاری و مدولاسیون اصطلاحاً **خود سازگار**[1] در سیستم‌های داده‌ی سلولی و نیز در شبکه‌های WIMAX 802.16 و WiFi 802.11 استفاده می‌شود که آن‌ها در بخش ۶-۴ مطالعه خواهند شد. این نوع روش‌ها این امکان را فراهم می‌کنند که از طریق آن‌ها بتوان یک روش مدولاسیون با بالاترین سرعت انتقال را در شرایط وجود محدودیت BER برای یک کانال با مشخصات ویژه تامین نمود.

افزایش خطای بیت و متغیر شدن این خطا به نسبت تغییرات زمانی تنها تفاوت بین یک پیوند باسیم و بی‌سیم نمی‌باشد. بار دیگر این نکته یادآوری می‌شود که در یک پیوند فراپخش کابلی همه‌ی گره‌ها اقدام به دریافت اطلاعات از دیگر گره‌ها می‌نمودند. در چنین حالتی پیوندهای بی‌سیم همانگونه که شکل ۶-۴ نشان می‌دهد. به این سادگی عمل نمی‌کنند. اگر فرض شود پایانه‌ی A در حال انتقال داده به پایانه‌ی B می‌باشد و نیز اگر فرض شود پایانه‌ی C در حال انتقال به پایانه‌ی B می‌باشد آنگاه با آنچه اصطلاحاً در واژگان شبکه‌های کامپیوتری بی‌سیم به **چالش پایانه‌ی نهان**[2] معروف شده است، یک مانع فیزیکی در محیط مانند یک کوه و یا یک ساختمان بلند ممکن است از شنود انتقال بین A و C جلوگیری نماید به عبارت دیگر هیچ یک از مبادله‌ی دیگری آگاه نشود، حتی اگر انتقال‌های A و C در مقصد B با یکدیگر تداخل داشته باشند. این حقیقت به وسیله‌ی شکل ۶-۴ (الف) نشان داده شده است. حالت دوم دیگری که نتیجه‌ی آن عدم آشکارسازی برخورد در گیرنده می‌باشد، مربوط به

1- Adaptive modulation
2- Hidden terminal problem

وضعیتی است که اصطلاحاً به آن **افت ناگهانی**[1] قدرت سیگنال به هنگام انتشار از درون رسانه‌ی بی‌سیم گفته می‌شود. شکل ۶-۴ (ب) حالتی را به نمایش می‌گذارد که A و C در نقاطی قرار گرفته‌اند که سیگنال‌های حاصل از آن‌ها به مقدار کافی قوی نیستند که قادر به تشخیص یکدیگر باشند، اما به میزانی قوی هستند که می‌توانند با یکدیگر در پایانه‌ی B تداخل نمایند. در بخش ۶-۴ ملاحظه خواهد شد که چالش پایانه‌ی نهان و افت ناگهانی دست‌یابی چندگانه در شبکه‌های بی‌سیم را به مراتب پیچیده‌تر از یک شبکه‌ی کابلی می‌کند.



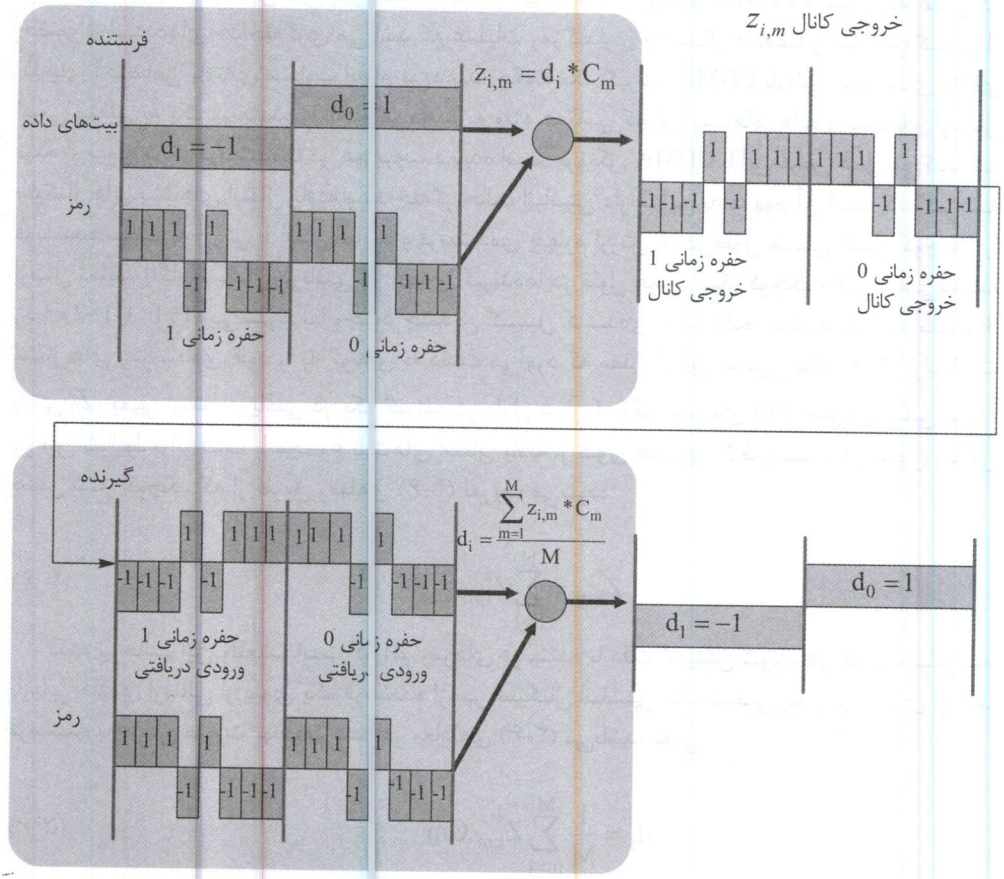
شکل ۶-۴ رخداد چالش پایانه‌ی نهان (الف) در اثر مانع و (ب) در اثر افت ناگهانی

۶-۳ دست‌یابی چندگانه‌ی تقسیم رمز (CDMA)

در فصل ۵ ملاحظه شد، هنگامی که چندین میزبان از طریق یک رسانه‌ی مشترک با یکدیگر ارتباط برقرار می‌نمودند، به منظور جلوگیری از تداخل سیگنال‌های ارسالی از سوی چندین فرستنده بهره‌گیری از یک پروتکل درگیرنده‌ها بسیار ضروری می‌نمود. در فصل ۵ سه گروه پروتکل رسانه‌ی دست‌یابی تعریف شد. این سه عبارت بودند از: پروتکل تقسیم و جداسازی کانال، دست‌یابی تصادفی و رعایت نوبت. **دست‌یابی چندگانه‌ی تقسیم رمز**[2] **(CDMA)** متعلق به خانواده‌ی پروتکل‌های تقسیم و جداسازی کانال می‌باشد. این پروتکل حاکم در فناوری‌های سلولی و LAN بی‌سیم می‌باشد. به دلیل اهمیت ویژه‌ی CDMA در دنیای شبکه‌های کامپیوتری بی‌سیم تحلیل مستقلی از آن در این زیربخش قبل از فناوری‌های دیگر دست‌یابی بی‌سیم انجام می‌شود. در یک پروتکل CDMA هر بیت ارسالی از طریق ضرب سیگنال بیت در یک سیگنال دیگر **(رمز)** که نرخ تغییر آن در هر ثانیه به مراتب بالاتر از سرعت بیت‌های داده‌ی ارسالی است و به آن اصطلاحاً **نرخ سنگ‌ریزه**[3] گفته می‌شود، رمز یا کُد می‌شود. شکل ۶-۵ یک روش ساده‌ی رمزگذاری و رمزگشایی CDMA را نشان می‌دهد. فرض کنید سرعتی که در آن بیت‌های داده‌ی اصلی به مقصد رمزگذار CDMA وارد می‌شود به صورت یکایی از زمان تعریف شود که در آن جهت انتقال هر بیت از داده‌ی اصلی نیاز به یک شیار یا حفره‌ی زمانی به

1- Fading
2- Code division multiplex access
3- Chipping rate

طول یک بیت می‌باشد. اگر d_i ارزش بیت داده برای شماره بیت i باشد. برای انجام عملیات ساده‌ی ریاضی هر بیت داده با ارزش 0 را به صورت 1- نمایش می‌دهیم. هر بیت در شیار خود به تعداد M شیار کوچکتر مطابق شکل ۶-۵ تقسیم می‌شود و همان‌گونه که شکل نشان می‌دهد در اینجا M=8 می‌باشد. بدیهی است در عمل M به مراتب ارزش بزرگ‌تری را اختیار می‌کند. رمز CDMA که از سوی فرستنده مورد بهره‌برداری قرار می‌گیرد شامل زنجیره‌ای ار ارزش‌های M به نام Cm است که m=1,...,M می‌باشد. ارزش هر Cm برابر = 1+ و یا 1- می‌باشد. در مثال شکل ۶-۵ تعداد M بیت رمز CDMA که از طریق کاربر مورد استفاده قرار می‌گیرد به گونه‌ی: (1,1,1,-1,1,-1,-1,-1) می‌باشد.



شکل ۶-۵ ♦ مثال ساده‌ای از یک **CDMA**، رمزگذاری فرستنده، رمزگشایی گیرنده

برای آنکه نشان دهیم CDMA چگونه عمل می‌کند، نقطه‌ی تمرکز بحث را بر روی بیت داده‌ی i یعنی d_i قرار می‌دهیم. شیار کوچکتر شماره‌ی m در زمان انتقال بیت d_i خارج شده از رمزگذار CDMA برابر m و Z_i در نظر گرفته می‌شود و ارزش آن برابر است با مقدار d_i ضرب شده در Cm که بیت شماره‌ی m در رمز اختصاص یافته‌ی CDMA می‌باشد. به عبارت دیگر رابطه‌ی (۶-۲):

$$Z_{i,m} = d_i.Cm \qquad (۶-۲)$$

در شرایط ساده و ایده‌آل که در آن فرستنده‌ها با یکدیگر در تداخل نمی‌باشند، گیرنده بیت‌های رمز شده‌ی $Z_{i,m}$ را دریافت و نسبت به بازیابی بیت داده‌ی اصلی d_i از طریق رابطه‌ی (۶-۳) اقدام می‌کند.

$$d_i = \frac{1}{M}\sum_{m=1}^{M} Z_{i,m}.Cm \qquad (۶-۳)$$

از رابطه‌ی (۶-۳) می‌توان جهت صحت عملکرد بازیابی صحیح بیت‌های داده در شکل ۶-۵ بهره گرفت.

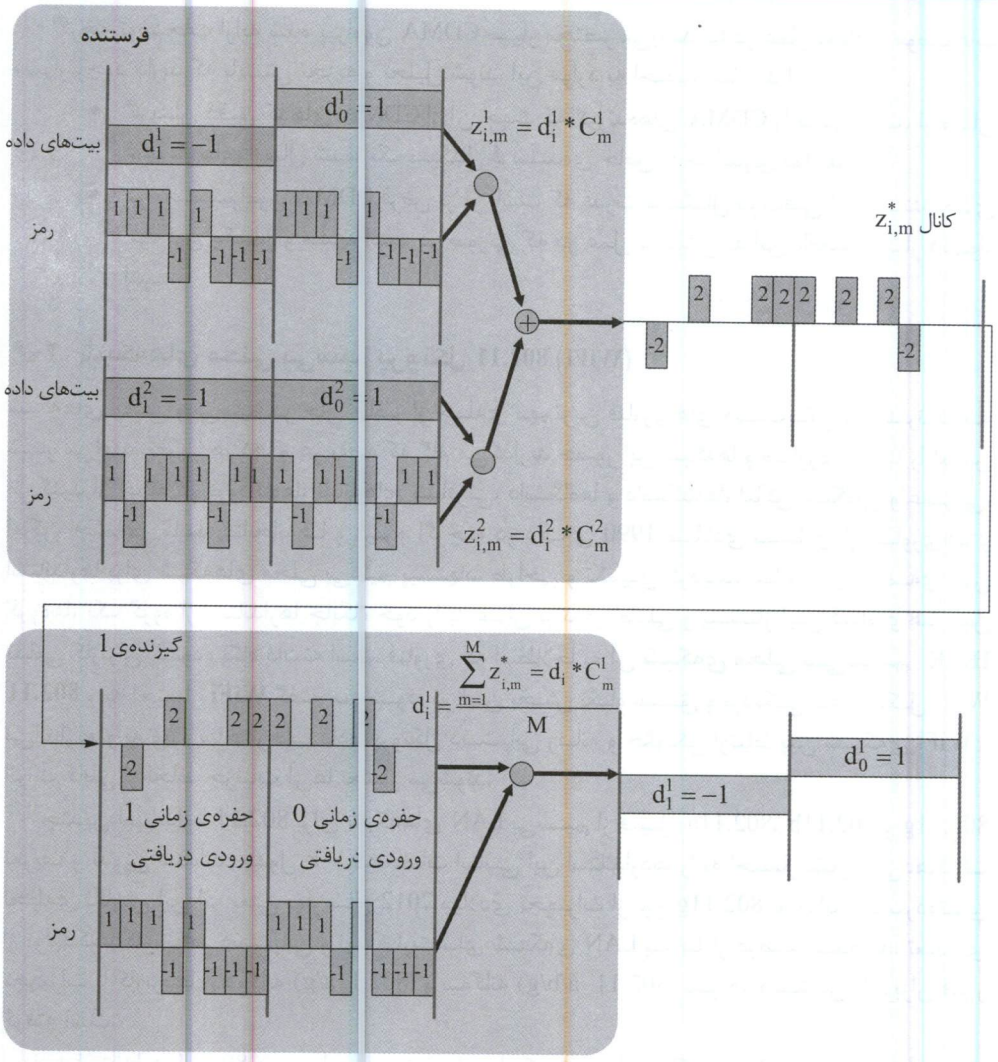
اما از آنجائیکه در عمل هیچ‌چیز ساده و ایده‌آل نمی‌باشد، پروتکل CDMA مجبور به عمل در حضور فرستنده‌های مداخله‌گری می‌باشد که عملیات رمزگذاری و انتقال داده‌ها را با بهره‌گیری از کدهای اختصاص یافته‌ی متفاوت انجام می‌دهند. چگونه یک گیرنده CDMA بازیابی بیت‌های داده‌ی اصلی متعلق به یک فرستنده را انجام می‌دهد؟ به ویژه هنگامی که این بیت‌های داده با بیت‌های ارسال شده از سوی دیگر فرستنده‌ها در هم پیچیده شده است. پروتکل CDMA با این فرض عمل می‌کند که سیگنال‌های بیت‌های انتقال یافته و مداخله‌گر حالت انباشتی دارند. این به مفهوم آن است که اگر سه فرستنده نسبت به ارزش 1 اقدام نمایند و فرستنده‌ی چهارم ارزش 1- در طول همان شیار کوچک را ارسال نماید، آنگاه سیگنال دریافتی در همه‌ی گیرنده‌ها در طول همان شیار کوچک برابر 2 می‌باشد، زیرا: (2=1-1+1+1) می‌شود. با وجود چندین گسیل کننده‌ی بیت داده، یک فرستنده مانند S انتقال‌های رمز شده‌ی خود را به گونه‌ای به دست می‌آورد که مقدار آن بر اساس رابطه (۶-۲) برابر است با: $Z^s_{i,m}$. یعنی ارزش دریافتی در یک گیرنده در طول شیار کوچک شماره‌ی m از حفره‌ی زمانی بیت i در این شرایط برابر است با مجموع بیت‌های انتقال یافته از سوی همه‌ی N فرستنده در طول زمانی همان شیار کوچک که از طریق رابطه‌ی (۶-۴) تعریف می‌شود:

$$Z^*_{i,m} = \sum_{S=1}^{N} Z^s_{i,m} \qquad (۶-۴)$$

نکته‌ی جالب این واقعیت است که اگر رمزهای فرستنده با دقت گزینش شوند، هر گیرنده قادر به بازیابی داده‌ی ارسالی از سوی یک فرستنده از بین سیگنال انباشتی یا تجمعی با بهره‌گیری از کُد فرستنده به همان صورت توصیف شده در معادله‌ی (۶-۳) می‌باشد، یعنی:

$$d_i = \frac{1}{M}\sum_{m=1}^{M} Z^*_{i,m}.Cm \qquad (۶-۵)$$

همان‌گونه که در شکل ۶-۶ نشان داده شده است برای یک مثال CDMA با دو فرستنده، تعداد M بیت کد CDMA به کار گرفته شده از سوی فرستنده‌ی بالاتر به صورت : (1,1,1,-1,1,-1,-1,-1) می‌باشد، درحالیکه رمزCDMA استفاده شده ازسوی فرستنده‌ی پائین‌تر در قالب:(1,-1,1,1,1,-1,1,1) می‌باشد، شکل ۶-۶ چگونگی بازیابی بیت‌های داده‌ی اصلی ارسالی از فرستنده‌ی بالاتر را به وسیله‌ی یک گیرنده نشان می‌دهد. بدیهی است که گیرنده قادر به جداسازی داده از فرستنده 1 می‌باشد علی‌رغم تداخلی که از سوی انتقال فرستنده‌ی 2 صورت می‌پذیرد.



شکل ۶-۶ ♦ نمونه‌ای از **CDMA** با دو فرستنده

بار دیگر مثال کلاس درس و دانشجویان فصل ۵ را در ذهن خود تصور نمائید. یک پروتکل CDMA شبیه به دانشجویانی در کلاس می‌باشد که به زبان‌های مختلف محاوره می‌کنند. بدیهی است در چنین شرایطی انسان‌ها با زیرکی ویژه‌ی خود عمل می‌نمایند یعنی حس شنوایی خود را متمرکز به محاوره‌ای می‌کنند که از لحاظ زبانی برای آن‌ها قابل فهم است، در حالیکه بقیه‌ی گفت‌وگوها به زبان‌های دیگر را فیلتر می‌کنند. بنابراین پروتکل CDMA از آن جهت یک پروتکل تقسیم‌بندی و جداسازی می‌باشد که در آن بخش‌های جدا شده (پارتیشن‌ها) **فضای رمز**[1] می‌باشند که هر گره را به یک قطعه‌ی انحصاری از فضای رمز اختصاص می‌دهند. در روش تقسیم‌بندی که در فصل ۵ به آن اشاره شد زمان و یا فرکانس (پهنا باند بسامد) به داده و نهایتاً به گره‌ها اختصاص می‌یافت.

1- Code space

اگرچه توضیحات ارایه شده پیرامون CDMA بسیار مختصر می‌باشد اما در عمل تعدادی موضوعات دشوار وجود دارند که بایستی تجزیه و تحلیل شوند. این موارد به اختصار عبارتند از:

- گزینش دقیق کدهای CDMA از آن جهت که گیرنده‌های CDMA را قادر نماید از میان سیگنال‌های ارسال شده، یک سیگنال فرستنده‌ی خاص را جداسازی نمایند.
- در بحث پیرامون CDMA فرض بر آن است که قدرت سیگنال دریافتی از فرستنده‌های گوناگون یکسان و مشابه است در صورتی که در عمل دستیابی به این واقعیت بسیار دشوار است.

۶-۴ شبکه‌های محلی بی‌سیم پروتکل 802.11 (WiFi)

شبکه‌های محلی و بی‌سیم در حال حاضر از جمله‌ی مهم‌ترین فناوری‌های دست‌یابی در اینترنت به شمار می‌آیند. تقریباً هرکجا و هر نقطه که گام می‌گذارید حضور این شبکه‌ها و فناوری آن‌ها را لمس می‌کنید از جمله در فرودگاه‌ها، پایانه‌های مسافرتی، دانشگاه‌ها و دانشکده‌ها، اماکن مسکونی و عمومی، اماکن اجتماعی مانند کتابخانه‌ها و غیره. اگرچه در دهه‌ی 1990 میلادی بسیاری از فناوری‌ها و استانداردها برای شبکه‌های محلی بی‌سیم پیشنهاد، طراحی و تکمیل گردید، اما در بین همه‌ی این گروه‌ها، یک گروه از استانداردها جایگاه خود را به عنوان برنده‌ی اصلی و پیشتاز این فناوری‌ها بین تمامی کاربران گشوده نگاه داشته است. فناوری که اصطلاحاً به آن **شبکه‌ی محلی بی‌سیم IEEE 802.11** و به اختصار **WiFi** گفته می‌شود. در این بخش نگاه عمیق و نزدیکی به پروتکل WiFi می‌اندازیم و به ویژه ساختار قاب داده، پروتکل دست‌یابی رسانه و چگونگی ارتباط بین شبکه‌ی WiFi و اترنت کابلی از لحاظ جزئیات آن‌ها تحلیل می‌شوند.

چندین استاندارد 802.11 برای شبکه‌های LAN بی‌سیم از قبیل 802.11a، 802.11b و 802.11g تعریف و تدوین شده‌اند. جدول ۶-۱ مشخصات اساسی این استانداردها را به اختصار نشان می‌دهد. تا لحظه‌ی نگارش این اثر یعنی بهار سال 2012 میلادی تجهیزات از نوع 802.11g به مراتب گسترده‌تری از تولیدکنندگان نقاط دست‌یابی و نیز کارت‌های شبکه‌ی LAN به بازار عرضه شده‌اند. تعدادی تجهیزات با کاربردهای دوگانه (802.11a/g) و سه‌گانه (802.11 a/b/g) نیز در دسترس کاربران قرار گرفته است.

سه استاندارد 802.11 در بسیاری از مشخصات با یکدیگر در اشتراک می‌باشند. همه‌ی آن‌ها از پروتکل دست‌یابی رسانه‌ی مشابه یعنی CSMA/CD که در ادامه تحلیل می‌شود، بهره می‌گیرند. هر سه استاندارد از ساختار قاب مشابه برای تعریف فریم‌های لایه‌ی پیوند خود استفاده می‌کنند و هر سه پروتکل قادر به کاهش سرعت انتقال خویش جهت دسترسی به مسافت‌های طولانی‌تر می‌باشند و هر سه استاندارد امکان کاربرد روش‌های اصطلاحاً **روزمره و ویژه (ad hoc)** و **زیرساخت** را فراهم می‌آورند. در عین حال همان‌گونه که جدول ۶-۱ نشان می‌دهد هر سه استاندارد از لحاظ لایه‌ی فیزیکی دارای تفاوت‌های عمده و بنیادی نیز می‌باشند.

از جمله موارد قابل توجهی که می‌توان آن‌ها را برشمرد، عبارت است از آن که استاندارد 802.11b در کاربرد شبکه‌های محلی بی‌سیم از سرعت انتقال 11Mbps و پهنای باند فرکانسی ثبت نشده‌ی GHz 2.485 – 2.4 بهره می‌گیرد که آن را در رقابت با طیف فرکانس 2.4 GHz تلفن‌های بی‌سیم و دستگاه‌های مایکروفر قرار می‌دهد. از طرف دیگر پروتکل 802.11a در شبکه‌های محلی بی‌سیم از سرعت انتقال بالاتری برخوردار است اما در مقابل متحمل هزینه‌ی فرکانس‌های به مراتب بالاتر

می‌باشد. از طریق عملکرد در یک فرکانس بالاتر شبکه‌ی محلی پروتکل 802.11a از طول مسافت انتقال به مراتب کوتاه‌تری برای یک سطح قدرت تعریف شده برخوردار می‌باشد و نیز از ضعف انتشار چندمسیری رنج می‌برد. پروتکل شبکه‌ی بی‌سیم 802.11g در محدوده‌ی فرکانس پائین 802.11b عمل می‌نماید و قابلیت سازگاری با تجهیزات 802.11b را نیز دارد. اما در عین حال از سرعت انتقال بالاتری در مقایسه با 802.11a بهره‌مند می‌باشد.

جدول ۶-۱ ♦ خلاصه‌ای از استانداردهای IEEE 802.11

استاندارد	محدوده‌ی فرکانس(ویژه ایالات متحده آمریکا)	نرخ انتقال داده
802.11b	2.4 –2.485 GHz	حداکثر تا 11 Mbps
802.11a	5.1-5.8 GHz	حداکثر تا 54 Mbps
802.11g	2.4 –2.485 GHz	حداکثر تا 54 Mbps

یک استاندارد جدید WiFi که اخیراً تعریف شده است معروف به [IEEE 802.11n 2012] می‌باشد. پروتکل 802.11n از آنتن‌های اصطلاحاً **چند ورودی – چند خروجی**[1] **(MIMO)** بهره می‌گیرد. به عبارت دیگر دو یا چندین آنتن در سمت ارسال و دو یا چندین آنتن نیز در سمت دریافت که هر یک از آن‌ها می‌توانند سیگنال‌های گوناگونی را ارسال و یا دریافت نمایند. اگرچه هنوز استانداردهای مربوط به این پروتکل نهایی نشده‌اند اما تجهیزات ماقبل استاندارد آن‌ها به بازار کاربران عرضه شده‌اند. سنجش‌های اولیه نشان داده است که برآورد نهفتگی بیش از 200 Mbps در عمل برای آن‌ها امکان‌پذیر می‌باشد. در حال حاضر یک نگرانی از پیش نویس استاندارد جاری چگونگی تعامل تجهیزات 802.11n با تجهیزات موجود 802.11a/b/g می‌باشد.

- **معماری 802.11**

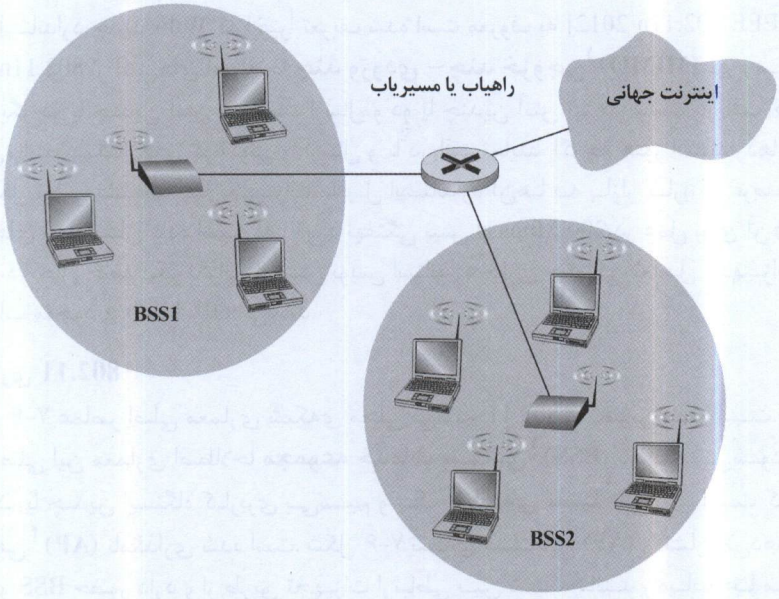
در شکل ۶-۷ عناصر اصلی معماری شبکه‌ی محلی بی‌سیم 802.11 به تصویر آمده است. مجموعه‌ی سازه‌ی اصلی این معماری اصطلاحاً **مجموعه خدمات بنیادی**[2] **(BSS)** نامیده می‌شود. یک BSS شامل یک یا چندین ایستگاه کاربری بی‌سیم و یک **پایانه‌ی بنا**[3] مرکزی است که **نقطه‌ی دست‌یابی**[4] **(AP)** نامگذاری شده است. شکل ۶-۷ نقطه‌ی دست‌یابی (AP) را نشان می‌دهد که در هر یک از دو BSS حضور دارد و از طریق تجهیزات ارتباطی بین شبکه مانند راه‌یاب و یا مسیریاب به یکدیگر متصل و از طریق آن‌ها نهایتاً با دیگر کاربران اینترنت در ارتباط قرار می‌گیرد.

همانند تجهیزات شبکه‌ی اترنت هر یک از پایانه‌های بی‌سیم 802.11 نیز دارای یک آدرس MAC به طول 6 بایت می‌باشند که در تراشه‌ی میان‌افزار کارت شبکه‌ی 802.11 تعریف و ذخیره شده است. هر AP نیز از یک آدرس MAC جهت واسط بی‌سیم خود بهره می‌گیرد. این آدرس‌های MAC مانند شبکه‌ی اترنت از سوی نهاد IEEE مدیریت می‌شوند و منحصر به فرد در سراسر گیتی می‌باشند.

1- Multiple input, multiple output
2- BasicService Set
3- Base Station
4- Access point

در زیر بخش ۶-۱ ملاحظه شد شبکه‌های محلی بی‌سیم که اغلب از نقاط دست‌یابی (AP) بهره می‌گیرند را اصطلاحاً **شبکه‌های LAN بی‌سیم زیرساخت** می‌نامند در این عنوان واژه‌ی **زیرساخت** به مفهوم AP و زیرساخت شبکه‌ی اترنت کابلی است که نقاط دست‌یابی (AP) و یک روتر را به یکدیگر ارتباط می‌دهند.

شکل ۶-۸ نشان می‌دهد که پایانه‌های IEEE 802.11 را می‌توان در قالب گروهی تعریف نمود و از این طریق یک شبکه‌ی **مستقل روزمره** (ad – hoc) تشکیل داد که عبارت است از شبکه‌ای که دارای هیچ نقطه‌ی مرکزی و ارتباطی با **دنیای خارج** از خود نمی‌باشد. در این حالت شبکه را بر مبنای رویکرد اصطلاحاً **در پرواز**[1] می‌نامند که از طریق تجهیزات سیار که هر یک خود را در مجاورت با دیگری مشاهده می‌کند، در ارتباط قرار می‌گیرند. با این ویژگی که نیازمند ارتباط می‌باشند و الزامی به یافتن زیرساخت شبکه‌ی از پیش ایجاد شده در موقعیت مکانی خود ندارند، یک شبکه‌ی مستقل روزمره (ad – hoc) ممکن است هنگامی تشکیل شود که مردمان با رایانه‌های همراه خود در یک سالن گردهمایی،یک ترن و یا یک خودرو قرار دارند و مایل به تبادل اطلاعات با یکدیگر در غیاب یک AP مرکزی می‌باشند.



شکل ۶-۷ ♦ معماری شبکه‌ی محلی **IEEE 802.11**

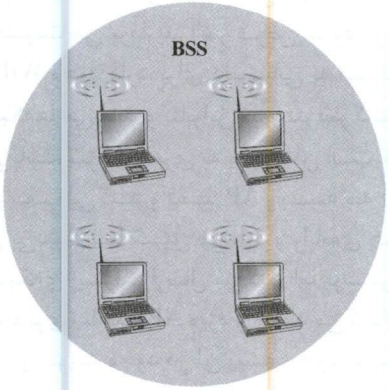
شبکه‌های ad – hoc مورد توجه بسیاری از دست‌اندکاران و پژوهشگران شبکه می‌باشند و علت آن نیز گسترش روزافزون تجهیزات ارتباطی قابل حمل در سراسر گیتی است. بدیهی است در این زیر بخش نقطه‌ی تمرکز بر روی زیرساخت محلی بی‌سیم می‌باشد.

• کانال و وابستگی

در پروتکل 802.11 هر پایانه‌ی بی‌سیم نیازمند وابستگی به یک AP قبل از ارسال و یا دریافت داده‌های

1- On the fly

لایه‌ی شبکه می‌باشد. اگرچه همه‌ی استانداردهای 802.11 از روش وابستگی یا به عبارت دیگر شرکت‌پذیری بهره می‌گیرند اما در این زیربخش توجه ما به این موضوع در چارچوب پروتکل 802.11b/g می‌باشد.



شکل ۶-۸ ♦ یک شبکه‌ی روزمره‌ی متصل (**ad – hoc**) بر مبنای پروتکل **IEEE 802.11**

هنگامی که یک مدیر شبکه یک AP را نصب می‌کند، وی به اختصاص یک یا دو واژه‌ی **شناسه‌گر مجموعه خدمات**[1] **(SSID)** به آن AP می‌پردازد. به عنوان مثال هنگامی که گزینه‌ی **مشاهده‌ی شبکه‌های قابل دسترس** را در ویندوز XP یا 7 مایکروسافت ملاحظه می‌کنید، فهرستی به نمایش در می‌آید که SSID هر AP در محدوده را به شما نشان می‌دهد. مدیر شبکه هم‌چنین بایستی یک **شماره‌ی کانال** را نیز به AP اختصاص دهد. جهت درک بهتر مفهوم شماره کانال این نکته یادآوری می‌شود که 802.11 در محدوده برد فرکانس 2.4 تا 2.485 گیگاهرتز عمل می‌نماید. در بین این محدوده به وسعت 85 MHz پروتکل 802.11 تعداد 11 کانال که به صورت نسبی هم‌پوشانی دارد، را تعریف می‌کند. هر یک از دو کانال اصطلاحاً دچار هم‌پوشانی نخواهند شد اگر فقط از طریق 4 کانال و یا بیشتر از یکدیگر جدا شده باشند. به ویژه مجموعه کانال‌های 1، 6 و 11 تنها مجموعه‌ای از سه کانال بدون هم‌پوشانی به شمار می‌آیند. این به مفهوم آن است که یک مدیر شبکه قادر به تعریف یک شبکه‌ی محلی بی‌سیم با سرعت انتقال انباشتی حداکثر 32 Mbps از طریق نصب سه نقطه‌ی دست‌یابی 802.11 در یک مکان فیزیکی می‌باشد، که در آن کانال‌های 1، 6 و 11 به AP اختصاص یافته‌اند و هر یک از نقاط دست‌یابی AP نیز به یک راه‌یاب متصل می‌باشند.

با این آگاهی بنیادی از کانال‌های 802.11 به توصیف یک حالت جالب و متداول می‌پردازیم و آن اصطلاحاً یک **بیشه‌ی WiFi**[2] می‌باشد. یک بیشه‌ی WiFi عبارت است از هر موقعیت فیزیکی که در آن پایانه‌ی بی‌سیم یک سیگنال به اندازه‌ی کافی قدرتمندی را از دو یا چندین AP دریافت می‌کند. به عنوان مثال در بسیاری از مراکز اجتماعی در شهری مانند نیویورک، یک پایانه‌ی بی‌سیم می‌تواند سیگنالی را از تعداد بی‌شماری AP که در نزدیکی یکدیگر قرار دارند، دریافت نماید. در میان این تعداد بی‌شمار AP یکی از آن‌ها از سوی یک مرکز اجتماعی مانند یک کافه اینترنت مدیریت شده است در حالیکه دیگر APها ممکن است در آپارتمان‌های مسکونی نزدیک آن کافه مستقر باشند. هر یک از این

1- Service set identifier
2- WiFi jungle

نقاط دست‌یابی (AP) احتمالاً ممکن است در زیر شبکه‌های IP مختلف قرار گرفته باشند و به هر یک نیز یک کانال به گونه‌ی مستقل اختصاص یافته باشد.

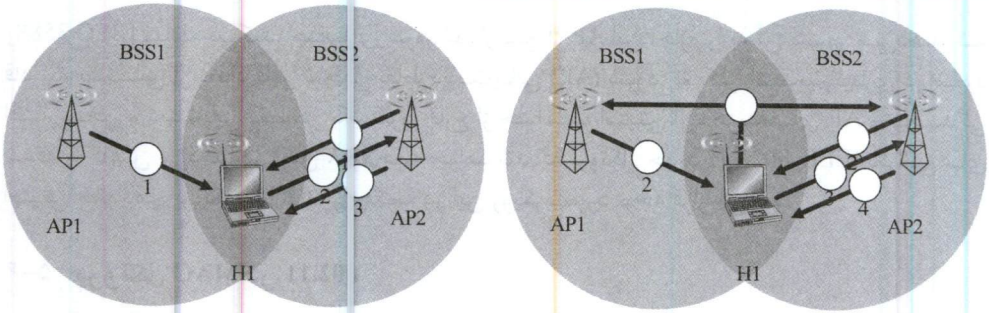
اکنون فرض کنید شما وارد یک بیشه‌ی WiFi می‌شوید و از طریق رایانه‌ی همراه خود به جستجوی یک دست‌یابی به اینترنت می پردازید. اگر فرض شود در این موقعیت تعداد 5 نقطه‌ی دست‌یابی (AP) در بیشه‌ی WiFi وجود دارد. برای دست‌یابی به اینترنت، پایانه‌ی بی‌سیم شما نیازمند پیوستن فقط به یکی از زیرشبکه‌ها می‌باشد و بنابراین نیازمند **شرکت‌پذیری**[1] یا **وابستگی** فقط با یکی از این نقاط دست‌یابی می‌باشد. شرکت‌پذیری به مفهوم آن است که یک پایانه‌ی بی‌سیم یک کابل مجازی را بین خود و AP تعریف می‌کند و فقط AP وابسته به آن، قاب‌های داده را که شامل داده‌گرام‌ها می‌باشد به پایانه‌ی بی‌سیم شما گسیل می‌دارد و رایانه‌ی همراه شما نیز قاب‌های داده را از طریق همان AP وابسته به شبکه‌ی اینترنت ارسال می‌کند. اما پرسش آن است که چگونه پایانه‌ی بی‌سیم شما وابستگی (شرکت‌پذیری) با یک AP ویژه را ایجاد می‌کند؟ و از همه مهم‌تر، چگونه پایانه‌ی بی‌سیم شما می‌داند کدام AP در صورت وجود در آن بیشه حضور دارد؟

استاندارد 802.11 یک AP را مجبور می‌کند که متناوباً **فریم‌های ردیابی**[2] **(بیکن)** گسیل دارد که هر یک از آن‌ها شامل SSID متعلق به AP و آدرس MAC آن می‌باشد. پایانه‌ی بی‌سیم شما با این آگاهی که AP ها قاب‌های ردیابی را ارسال می‌نمایند نسبت به جستجوی 11 کانال جهت شناسایی قاب‌های رهگیری از هر یک از نقاط دست‌یابی (AP) که ممکن است حضور داشته باشند، اقدام می‌کند. بدیهی است برخی از آن‌ها ممکن است این انتقال را بر روی کانال مشابهی انجام دهند. با آگاهی از AP در دسترس که از طریق قاب‌های رهگیری صورت می‌گیرد میزبان بی‌سیم شما یکی از آن نقاط دست‌یابی (AP) را جهت عملیات شرکت‌پذیری گزینش می‌کند.

استاندارد 802.11 به تعریف الگوریتمی از چگونگی گزینش AP در دسترس جهت وابستگی نمی‌پردازد، الگوریتم ویژه‌ی این عملیات در اختیار طراحان نرم‌افزار و میان‌افزار پروتکل 802.11 گذارده شده است تا آن‌ها این مهم را در میزبان بی‌سیم جایگذاری نمایند. آنچه در عمل انجام می شود گزینش یک AP از سوی میزبان است که قاب ردیابی آن بیشترین قدرت سیگنال دریافتی را شامل گردد. اگرچه دریافت یک سیگنال نسبتاً قوی بر اساس شکل ۶-۳ می‌تواند سودمند باشد، اما قدرت سیگنال تنها عامل مشخصات AP نیست که کارایی دریافت سیگنال از سوی میزبان را تعریف می‌کند. به ویژه یک AP گزینش شده ممکن است از لحاظ قدرت سیگنال از مقیاس قابل قبولی بهره‌مند باشد اما از سوی میزبان‌های وابسته‌ی دیگر خود که مایل به اشتراک پهنای باند بی‌سیم در آن AP می‌باشند، دچار سرریزی شده باشد و این در حالی است که یک AP کاملاً آزاد از لحاظ ترافیک به دلیل آنکه قدرت سیگنال آن تا حدودی ضعیف‌تر است، گزینش نشده است. چندین راه‌کار جایگزین به منظور گزینش AP نیز اخیراً پیشنهاد شده است.

فرآیند کاوش کانال‌ها و شنود قاب‌های ردیابی در واژگان شبکه‌های بی‌سیم به **رهگیری خنثی**[3] بر اساس شکل ۶-۹ (الف) معروف می‌باشد. یک میزبان بی‌سیم می‌تواند عملیات رهگیری را **فعال** نیز انجام دهد. این فرآیند از طریق فراپخش یک فریم کاوشگر که از سوی همه‌ی نقاط دست‌یابی در محدوده‌ی میزبان‌های بی‌سیم قابل دریافت است، انجام می‌گیرد. شکل ۶-۹ (ب) این فرآیند را نشان می‌دهد.

1- Association
2- Beacon frames
3- Passive scanning



(ب) رهگیری فعال
۱- فراپخش قاب رهگیری از H1.
۲- قاب پاسخ رهگیری از APها ارسال می‌شود.
۳- ارسال قاب تقاضای وابستگی از H1 به AP گزینش شده
۴-ارسال قاب پاسخ وابستگی از AP گزینش شده به H1

(الف) رهگیری خنثی
۱- قاب‌های رهگیری ازAP ها ارسال می‌شود.
۲- تقاضای وابستگی قاب ارسال می‌شود وH1 به AP گزینش شده گسیل می‌شود.
۳- پاسخ وابستگی از AP گزینش شده به H1 ارسال می‌شود.

شکل ۶-۹ ♦ کاوش فعال و خنثی برای نقاط دست‌یابی (**AP**)

واکنش نقاط دست‌یابی به تقاضای فریم کاوشگر از طریق یک قاب پاسخ به کاوشگر همراه می‌باشد. یک میزبان بی‌سیم پس از آن می‌تواند به گزینش AP مورد نظر جهت وابستگی از بین نقاط دست‌یابی پاسخ‌دهنده، اقدام نماید.

پس از گزینش AP شرکت‌پذیر، میزبان بی‌سیم یک قاب درخواست وابستگی را به نقطه‌ی دست‌یابی (AP) گسیل می‌کند و AP نیز پاسخ میزبان را با بازگشت پاسخ شرکت‌پذیری برای آن ارسال می‌کند. لازم به یادآوری است که این دست‌دهی دوم تقاضا و پاسخ جهت کاوش فعال لازم می‌باشد، زیرا هنگامی که یک نقطه‌ی دست‌یابی در حال نشان دادن واکنش به نخستین فریم تقاضای جستجو می‌باشد، نمی‌داند کدام یک از تعداد بی‌شمار AP در حال پاسخگویی را میزبان بی‌سیم جهت شرکت‌پذیری گزینش می‌کند. این مهم شبیه به عملیات یک برنامه‌ی مشتری DHCP است که می‌تواند از بین چندین سرور DHCP بر مبنای شکل ۴-۲۱ در فصل ۴ یک سرور را گزینش نماید. بی‌درنگ پس از وابستگی به یک AP آنگاه میزبان مایل به پیوستن به زیرشبکه‌ای است که آن AP به آن تعلق دارد به عبارت دیگر مانند آدرس‌دهی IP که در فصل ۴ تشریح شد. در این مقطع میزبان بی‌سیم یک پیام بازیابی DHCP را از طریق AP به زیرشبکه وارد می‌کند تا قادر به دست‌یابی یک آدرس IP از آن زیر شبکه گردد. پس از دریافت آدرس IP همه‌ی کاربران در سراسر گیتی قادر به مشاهده‌ی آن میزبان بی‌سیم به گونه‌ی یک میزبان عادی با یک آدرس IP در آن زیرشبکه می‌باشند.

جهت تعریف و ایجاد یک فرآیند شرکت‌پذیری با یک AP خاص، پایانه‌ی بی‌سیم ممکن است نیازمند اعتبارسنجی خود با AP باشد. پروتکل محلی بی‌سیم 802.11 شماری از روش‌ها را جهت اعتبارسنجی و دست‌یابی در اختیار دارد. یک رویکرد که از سوی شرکت‌های تجاری به صورت گسترده مورد استفاده قرار می‌گیرد اجازه‌ی دست‌یابی به یک شبکه‌ی بی‌سیم بر مبنای آدرس MAC پایانه‌ی کامپیوتر می‌باشد. رویکرد دوم که بسیاری از مراکز اجتماعی مانند رستوران‌ها و غیره از آن بهره می‌گیرند به کارگیری یک نام کاربری و رمز عبور می‌باشد. در هر دو حالت نقطه‌ی دست‌یابی انحصاراً با یک سرور اعتبارسنج ارتباط برقرار می‌کند و اطلاعات خود را بین میزبان انتهایی بی‌سیم و آن سرور اعتبارسنج با بهره‌گیری از یکی از دو پروتکل [RFC 2865] RADIUS و یا DIAMETER

[RFC 3588] مبادله می‌کند. جداسازی سرور اعتبارسنج از AP این اجازه را به آن سرور می‌دهد که قادر به پشتیبانی از تعداد قابل توجهی نقاط دست‌یابی (AP) شود. تمرکز تصمیم‌گیری در مورد اعتبارسنجی و دست‌یابی فقط در یک سرور فارغ از حساسیت امنیتی آن، هزینه‌های AP و پیچیدگی شبکه را کاهش می‌دهد. در فصل ۸ ملاحظه خواهد شد که پروتکل جدید IEEE 802.11i نگرانی‌های امنیتی خانواده‌ی پروتکل 802.11 را چگونه در این رویکرد مورد توجه قرار می‌دهد.

۶-۵ پروتکل MAC در 802.11

پس از آنکه یک پایانه‌ی بی‌سیم وابستگی خود را با یک AP برقرار نماید آغاز به ارسال و دریافت قاب‌های داده به طرف نقطه‌ی دست‌یابی و برعکس آن می‌کند. از آنجائیکه چندین میزبان بی‌سیم ممکن است مایل به انتقال قاب‌های داده به صورت همزمان بر روی یک کانال مشابه باشند، یک پروتکل دست‌یابی چندگانه جهت هماهنگ‌سازی انتقال‌ها لازم و ضروری می‌باشد. در این حالت یک **پایانه** ممکن است یک میزبان بی‌سیم و یا یک AP باشد. همان‌گونه که در فصل ۵ و زیربخش ۶-۳ تحلیل شد به طور کلی سه گروه پروتکل دست‌یابی چندگانه وجود دارند: (۱) تقسیم کانال که شامل CDMA نیز می‌شود، (۲) دست‌یابی تصادفی و (۳) رعایت نوبت. با توجه به موقعیت خیره کننده‌ی اترنت و پروتکل دست‌یابی تصادفی آن، طراحان 802.11 یک پروتکل دست‌یابی تصادفی را نیز جهت شبکه‌های محلی بی‌سیم 802.11 گزینش نموده‌اند. این پروتکل دست‌یابی تصادفی اصطلاحاً **دست‌یابی چندگانه‌ی حس حامل با جلوگیری از وقوع برخورد**[1] **(CSMA/CA)** می‌باشد. بخش CSMA در این پروتکل مانند CSMA در پروتکل اترنت می‌باشد به این مفهوم که هر میزبان بی‌سیم قبل از انجام عملیات انتقال کانال را حس می‌کند و در صورت مشغول بودن آن از انجام انتقال اطلاعات خودداری می‌کند. اگرچه هر دو پروتکل اترنت و 802.11 از CSMA بهره می‌گیرند اما تفاوت‌های عمده‌ای بین دو پروتکل MAC وجود دارد. این تفاوت‌ها در ذیل توضیح داده شده‌اند.

نخست آن‌که جایگزین بهره‌گیری از آشکارسازی برخورد، پروتکل 802.11 از فناوری **ممانعت از برخورد** استفاده می‌کند. دوم به دلیل نرخ نسبتاً بالای خطای بیت در کانال‌های بی‌سیم، پروتکل 802.11 برعکس اترنت از روش انتقال مجدد و پیام تائید (ACK) در لایه‌ی پیوند یعنی (ARQ) بهره می‌گیرد. با یادآوری مطالب زیربخش‌های ۵-۴ و ۵-۶ در رابطه با الگوریتم آشکارسازی برخورد در اترنت، هر گره هنگام انتقال به کانال نیز گوش فرا می‌دهد. اگر در هنگام انتقال تشخیص دهد که کارت شبکه‌ی دیگری در حال انتقال همزمان با آن می‌باشد، آنگاه به انتقال پایان می‌دهد و پس از یک درنگ کوتاه کاملاً تصادفی مجدداً تلاش جهت از سرگیری عملیات انتقال می‌کند. برعکس پروتکل اترنت 802.11، پروتکل MAC لایه‌ی پیوند 802.11 عملیات تشخیص برخورد را به دو دلیل عمده‌ی ذیل پیاده‌سازی نمی‌کند:

۱- توان‌مندی تشخیص برخورد قاب‌های داده به مفهوم توانایی در ارسال سیگنال داده‌ی یک پایانه و دریافت و یا تشخیص این که آیا پایانه‌ی دیگری همزمان در حال انتقال می‌باشد؟ از آنجائیکه قدرت سیگنال دریافتی بسیار کوچک و ضعیف در مقایسه با توان سیگنال ارسالی در پروتکل کارت شبکه‌ی 802.11 می‌باشد بنابراین طراحی و ساخت سخت‌افزار تشخیص برخورد بسیار پرهزینه می‌باشد.

1- Carrier sense multiple access with collision avoidance

۲- از همه مهم‌تر، حتی اگر کارت شبکه قادر به ارسال و شنود همزمان و نیز قطع انتقال به هنگام تشخیص مشغولی کانال باشد، کارت شبکه هنوز هم قادر به شناسایی همه‌ی برخوردها به دلیل چالش پایانه‌ی نهان و افت ناگهانی سیگنال که در زیربخش ۶-۲ تحلیل شد، نمی‌باشد.

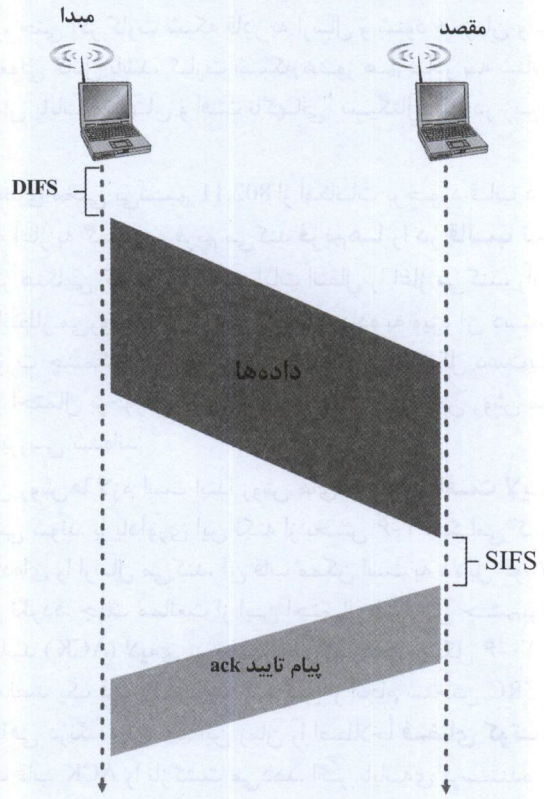
از آنجائیکه شبکه‌های محلی بی‌سیم 802.11 از امکانات برخورد قاب داده‌ها بهره نمی‌گیرند، هنگامی که یک پایانه آغاز به گسیل به فریم می‌کند **فریم‌ها را در قالب تمامیت آن‌ها ارسال می‌کند**، به بیان دیگر هنگامی که یک پایانه عملیات انتقال را آغاز می‌کند، راه بازگشتی برای آن وجود ندارد. همان‌گونه که انتظار می‌رود ارسال تمامی قاب‌های داده به ویژه آن دسته از قاب‌های طولانی در شرایط برخورد به صورت چشمگیری می‌تواند کیفیت و کارایی پروتکل دست‌یابی چندگانه را کاهش دهد. جهت کاستن از احتمال برخوردها، پروتکل 802.11 از چندین روش ممانعت از برخورد بهره می‌گیرد که در ادامه بررسی شده‌اند.

قبل از بررسی این روش‌ها لازم است ابتدا روش های **تائید دریافت لایه‌ی پیوند (ACK)** در پروتکل 802.11 بررسی شوند. با یادآوری این نکته از بخش ۶-۲ هنگامی که یک پایانه‌ی شبکه‌ی محلی بی‌سیم قاب داده‌ای را ارسال می‌کند، آن قاب ممکن است به دلایل گوناگون در قالب یکپارچه‌ی خود به مقصد تحویل نگردد. جهت ممانعت از این احتمال غیرقابل چشم‌پوشی، پروتکل MAC در 802.11 از پیام‌های تائید (ACK) لایه‌ی پیوند بهره می‌گیرد. در شکل ۶-۱۰ نشان داده شده است، هنگامی که پایانه‌ی مقصد یک قاب را دریافت کند پس از انجام سنجش CRC و اطمینان از سالم بودن فریم، مدت زمان کوتاهی درنگ می‌نماید. این زمان را اصطلاحاً **فضای کوتاه بین فریم[1] (SIFS)** نامیده‌اند و سپس یک قاب ACK را بازگشت می‌دهد. اگر پایانه‌ی فرستنده در یک فاصله‌ی زمانی معین این پیام تائید را دریافت نکند، آنگاه از پروتکل CSMA/CD جهت دست‌یابی کانال بهره می‌گیرد. اگر پس از تعداد مرتبه‌ی تعریف شده‌ای برای انتقال مجدد قاب‌های داده، یک پیام ACK دریافت نگردد، آنگاه پایانه‌ی فرستنده تلاش خود را متوقف و از فریم‌ها صرف‌نظر می‌کند.

پس از بررسی این موضوع که چگونه پروتکل 802.11 از پیاده‌های تائید لایه‌ی پیوند استفاده می‌کند، اکنون وقت آن رسیده است که پروتکل CSMA/CD – 802.11 تشریح شود. فرض کنید یک پایانه مانند یک AP و یا یک میزبان بی‌سیم قاب داده‌ای را جهت گسیل در اختیار دارد. مراحل ذیل احتمالاً انجام می‌گیرد:

- اگر در ابتدا پایانه حس نماید کانال آزاد است، قاب داده‌ی خود را پس از یک درنگ تعریف شده و معین بر مبنای شکل ۶-۱۰ ارسال می‌کند. این زمان کوتاه اصطلاحاً **فضای توزیع شده‌ی بین فریم‌ها[2] (DIFS)** نام دارد.
- در غیر این‌صورت، پایانه یک دوره‌ی زمانی عقب‌نشینی تصادفی را گزینش می‌کند و ارزش اختصاص یافته‌ی این زمان را به طرف پائین شمارش می‌کند (شمارش معکوس زمان)، تا به این نتیجه رسد که کانال آزاد شده است. بدیهی است تا آن هنگام که این حس وجود دارد که کانال مشغول است ارزش درون شمارنده مسدود و بدون تغییر باقی می‌ماند.

1- Short inter-frame spacing
2- Distributed inter-frame spacing



شکل ۶-۱۰ ♦ بهره‌گیری **802.11** از پیام‌های تائید لایه‌ی پیوند

- هنگامی که شمارنده به ارزش صفر می‌رسد، بدیهی است این وضعیت هنگامی پدیدار می‌شود که کانال آزاده شده باشد، آنگاه پایانه تمامی قاب خود را گسیل و سپس منتظر تائید (ACK) آن می‌شود.
- اگر پیام ACK دریافت شود آنگاه پایانه‌ی فرستنده مطمئن می‌شود که فریم آن سالم دریافت شده است و چنانچه قاب دیگری جهت ارسال دارد، مجدداً گام دوم در مراحل CSMA/CD را در همین مجموعه آغاز می‌کند. اگر پیام ACK دریافت نگردد، پایانه‌ی فرستنده وارد دوره‌ی عقب‌نشینی در مرحله‌ی دوم می‌شود و یک ارزش تصادفی را با یک تناوب زمانی طولانی‌تر گزینش می‌کند.

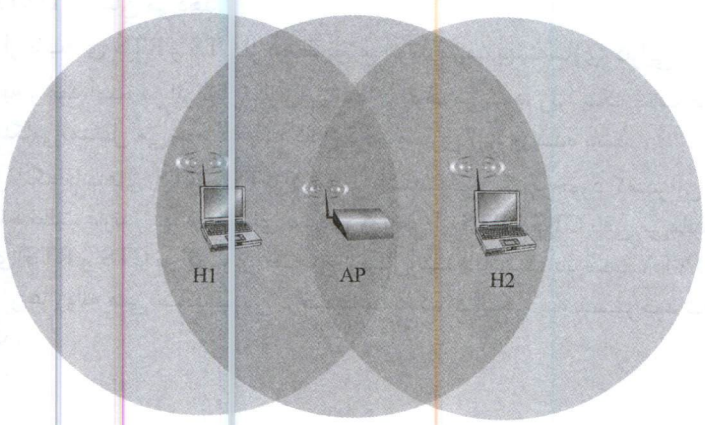
یادآور می‌شود در شرایط پروتکل دست‌یابی چندگانه CSMA/CD در اترنت در فصل ۵ یک گره بی‌درنگ پس از اطمینان از آزاد بودن کانال عملیات انتقال قاب را آغاز می‌نمود. در CSMA/CA یک پایانه هنگامی که مشغول شمارش معکوس است از ارسال قاب خودداری می‌کند حتی اگر حس کند کانال آزاد است. حال این پرسش مطرح می‌شود، چرا این دو پروتکل یعنی CSMA/CD و CSMA/CA چنین رویکرد متفاوتی را اختیار می‌کنند؟

جهت پاسخ این پرسش حالتی را در نظر بگیرید که در آن دو پایانه هر یک داده‌هایی را ارسال در اختیار دارند اما هیچ یک بی‌درنگ عملیات گسیل قاب داده را آغاز نمی‌کنند زیرا هر یک از آن‌ها بر این احساس می‌باشند که پایانه‌ی سومی در حال حاضر مشغول انتقال اطلاعات است. در حالت پروتکل

اترنت CAMA/CD، دو پایانه بی‌درنگ پس از این احساس که سوی انتقال خود را به پایان رسانیده است هر کدام آغاز به ارسال قاب داده می‌کنند و این باعث ایجاد برخورد می‌شود که در حالت پروتکل CSMA/CD چندان موضوع مهمی به شمار نمی‌آید زیرا هر دو فرستنده عملیات ارسال را متوقف و مانع ارسال بی‌حاصل اطلاعات بعدی می‌شوند. اما در پروتکل 802.11 این وضعیت کاملاً متفاوت است، زیرا این پروتکل وقوع برخورد را آشکارسازی و عملیات گسیل را نیز متوقف نمی‌کند، یک قاب داده‌ی دچار وضعیت برخورد در قالب تمامیت یکپارچه‌ی خود ارسال می‌شود. آنچه در 802.11 هدف می‌باشد ممانعت از برخورد در حدّ ممکن است. در 802.11 اگر دو پایانه مشغول بودن کانالی را حس نمایند هر دو بی‌درنگ وارد یک دوره‌ی عقب‌نشینی تصادفی می‌شوند با این امید که هر یک ارزش‌های متفاوتی را برای این عقب‌نشینی گزینش می‌کنند. اگر این ارزش‌ها گوناگون باشند بی‌درنگ پس از آن که یک کانال آزاد می‌شود، یکی از دو پایانه پیش از دیگری آغاز به عملیات انتقال می‌کند و اگر دو پایانه از یکدیگر نهان نباشند، آنگاه پایانه‌ی بازنده، سیگنال طرف برنده‌ی خود را ملاحظه می‌کند و در نتیجه عملیات شمارنده‌ی خود را متوقف می‌کند و از انتقال نیز خودداری نموده تا پایانه‌ی برنده، عملیات انتقال خود را کامل نماید. در این حالت است که از یک برخورد پرهزینه جلوگیری می‌شود. بدیهی است در 802.11 هنوز هم وقوع برخورد در این حالت اجتناب‌ناپذیر است، دو پایانه ممکن است از یکدیگر پنهان باشند و یا آنکه ارزش‌های تصادفی عقب‌نشینی که گزینش می‌کنند بسیار به هم نزدیک باشد به گونه‌ای که انتقال اطلاعات از پایانه آغازگر هنوز بایستی به پایانه‌ی دوم منتقل شود. یادآوری می‌شود که این دشواری در بحث الگوریتم دست‌یابی تصادفی در چارچوب شکل ۵-۱۴ نیز ملاحظه شد.

- **چالش پایانه‌های نهان (RTS و CTS)**

پروتکل MAC در 802.11 شامل یک روش ذخیره‌سازی مهم اما اختیاری می‌باشد که ممانعت از ایجاد برخورد می‌کند، حتی هنگامی که پایانه‌های نهان وجود دارند. این روش در چارچوب شکل ۶-۱۱ تحلیل می‌شود که دو پایانه‌ی بی‌سیم و یک نقطه‌ی دست‌یابی (AP) را نشان می‌دهد. هر دو پایانه بر اساس شکل ۶-۱۱ در محدوده‌ی برد (AP) قرار دارند که فضای تحت پوشش آن در قالب یک دایره‌ی سایه‌دار نشان داده شده است و هر دو با (AP) حالت شرکت‌پذیری دارند.



شکل ۶-۱۱ ♦ نمونه‌ی پایانه‌ی نهان: **H1** از پایانه‌ی **H2** پنهان است و برعکس

به علت وجود افت ناگهانی محدوده‌ی برد سیگنال، پایانه‌های بی‌سیم محدود به درون قلمروهای سایه‌دار در شکل ۶-۱۱ می‌باشند. بنابراین هر یک از پایانه‌های بی‌سیم از دیگری نهان می‌باشد اگر چه هر دو پایانه برای AP آشکار باشند.

اکنون این نکته بررسی می‌شود چرا پایانه‌های نهان می‌توانند مشکل‌آفرین باشند. فرض کنید پایانه‌ی H1 در حال انتقال یک قاب داده می‌باشد و در نیمه‌ی انجام عملیات انتقال پایانه‌ی H2 مایل به ارسال یک قاب داده به AP می‌باشد. فرستنده‌ی H2 قادر به شنیدن انتقال H1 نمی‌باشد بنابراین نخست برای یک دوره‌ی (DFIS) منتظر می‌شود و سپس فریم خود را انتقال می‌دهد و در نتیجه برخورد روی می‌دهد. حاصل این برخورد به مفهوم آن است که تمامی ظرفیت کانال در دوره‌ی زمانی دو انتقال H1 و H2 بی‌حاصل از دست‌رفته است.

جهت ممانعت از وقوع چنین وضعیتی پروتکل IEEE 802.11 این اجازه را می‌دهد که یک پایانه از پیام کوتاه **تقاضا برای ارسال**[1] **(RTS)** جهت کنترل فریم‌ها و از یک پیام کنترل کوتاه دیگر یعنی **آزاد جهت ارسال**[2] **(CTS)** برای **ذخیره‌سازی** (رزرو) دست‌یابی به کانال بهره گیرد. هنگامی که فرستنده مایل به ارسال یک قاب DATA می‌باشد، نخست یک فریم RTS به نقطه‌ی دست‌یابی (AP) گسیل می‌کند و از طریق آن حداکثر زمان لازم برای ارسال قاب DATA و پیام ACK برای آن را مشخص می‌کند. هنگامی که AP پیام RTS را دریافت نماید از طریق ارسال یک پیام CTS به پاسخ‌گویی آن می‌پردازد. این قاب CTS دو هدف را دنبال می‌کند: (۱) به فرستنده اختیار کامل می‌دهد که عملیات ارسال را انجام دهد و (۲) دیگر پایانه‌ها را مجبور می‌کند برای یک دوره‌ی زمانی معین از گسیل قاب‌های داده خودداری نمایند.

بنابراین در شکل ۶-۱۲ قبل از ارسال یک فریم DATA، پایانه‌ی H1 نخستین قاب RTS را فراپخش می‌کند و این پیام از سوی همه‌ی پایانه‌های در محدوده‌ی حلقه‌ی پوشش و حتی AP نیز شنیده می‌شود. پایانه‌ی AP سپس با یک پیام CTS واکنش نشان می‌دهد که این قاب CTS نیز از سوی همه‌ی پایانه‌های در محدوده‌ی پوشش که شامل H1 و H2 نیز می‌باشند دریافت می‌گردد. پایانه‌ی H2 پس از شنیدن CTS از انجام عملیات انتقال برای یک دوره‌ی زمانی معین که از سوی CTS تعریف شده است خودداری می‌کند. شکل ۶-۱۲ عملیات مربوط به سه قاب پیام‌های کوتاه RTS، CTS و قاب DATA را نشان می‌دهد.

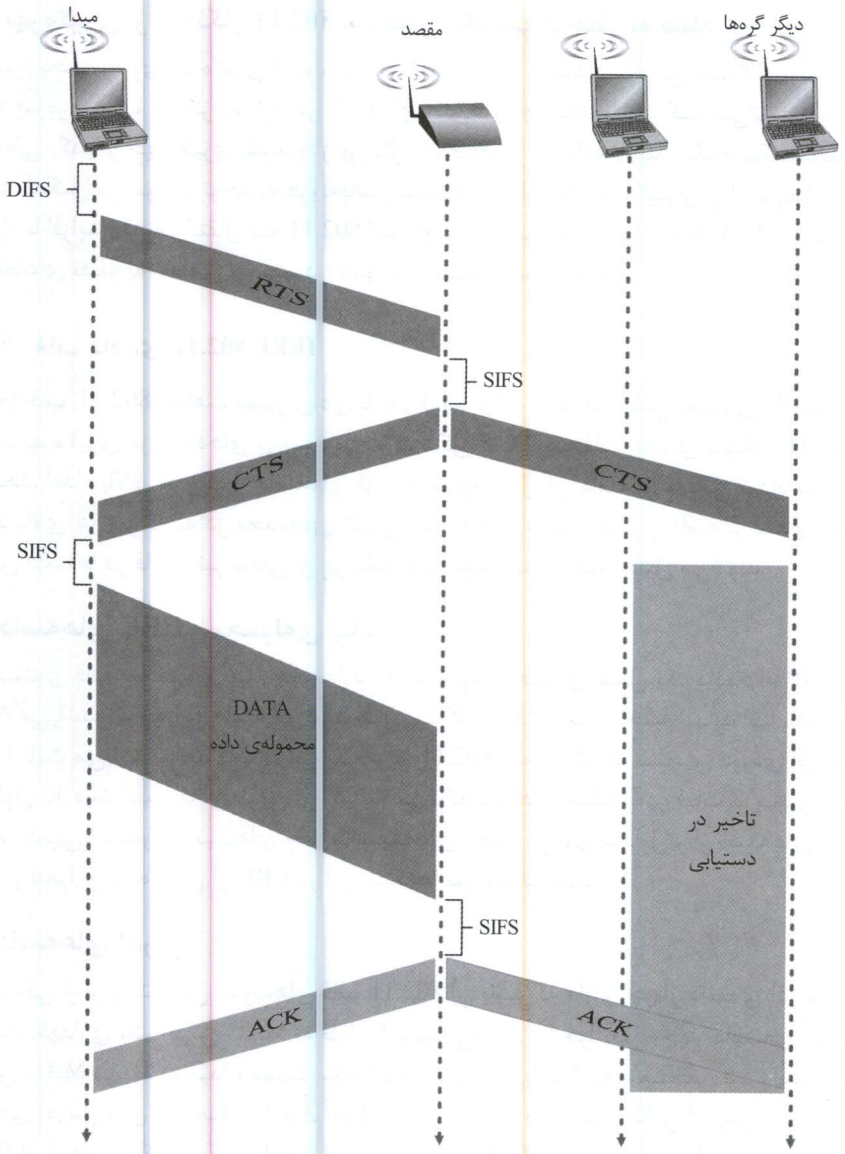
بهره‌گیری از قاب‌های RTS و CTS از دو طریق می‌تواند سبب بهینه‌سازی کارایی شود:

۱- چالش به وجود آمده در اثر نهان‌سازی پایانه‌ها برطرف می‌شود، زیرا یک قاب طولانی DATA فقط هنگامی منتقل می‌شود که یک کانال ذخیره‌سازی (رزرو) شده باشد.

۲- به دلیل آنکه قاب‌های RTS و CTS کوتاه می‌باشند، وقوع یک برخورد که در آن یک قاب RTS و CTS دخالت دارند تنها فقط برای دوره‌ی کوتاهی از RTS و CTS باقی می‌ماند. هنگامی که قاب‌های RTS و CTS به گونه‌ی صحیح ارسال شده باشند، دو قاب DATA و ACK که در پی آن‌ها روانه می‌شوند بدون برخورد با دیگر قاب‌ها به مسیرخود تا مقصد ادامه می‌دهند.

1- Request to send

2- Clear to send



شکل ۶-۱۲ ♦ ممانعت از برخورد با بهره‌گیری از قاب‌های **RTS** و **CTS**

اگر چه تبادل قاب‌های RTS/CTS می‌تواند سبب کاهش برخورد شود اما می‌تواند سبب تاخیر یا مصرف منابع کانال نیز باشد. به این دلیل از تبادل RTS/CTS فقط جهت ذخیره‌سازی کانال برای انتقال قاب طولانی DATA استفاده می‌شود. در عمل هر پایانه‌ی بی‌سیم می‌تواند یک آستانه‌ی RTS را به گونه‌ای تنظیم نماید که از تبادل RTS/CTS فقط وقتی استفاده شود که اندازه‌ی هر قاب داده از حدّ آستانه‌ی تعریف شده طولانی‌تر باشد. برای بسیاری از پایانه‌های بی‌سیم ارزش آستانه‌ی پیش‌فرض RTS بزرگتر از حداکثر طول قاب داده است بنابراین از مبادله RTS/CTS برای همه قاب‌های DATA گسیل شده خودداری می‌شود.

- **بهره‌گیری از پروتکل 802.11 به عنوان یک پیوند نقطه به نقطه**

تاکنون بحث بر روی بهره‌گیری از پروتکل 802.11 در یک محیط دست‌یابی چندگانه بود. یادآوری این نکته ضروری است که اگر دو گره هر یک دارای آنتن‌های جهت‌دار باشند که می‌توانند در خط دید ارتباطی یکدیگر جهت‌گیری شوند و از پروتکل 802.11 نیز استفاده نمایند آنگاه یک پیوند نقطه به نقطه تشکیل می‌شود. با توجه به هزینه‌پائین سخت‌افزار 802.11 بهره‌گیری از آنتن‌های جهت‌دار و همراه با افزایش قدرت انتقال به 802.11 این اجازه را می‌دهد که از امکانات ارزان قیمت جهت ارتباط‌های نقطه به نقطه‌ی بی‌سیم در کیلومترها مسافت استفاده نماید.

۶-۶ قاب داده‌ی IEEE 802.11

اگرچه قاب 802.11 شباهت بسیار زیادی با یک فریم اترنت دارد، اما شامل تعدادی دامنه‌ی ویژه نیز جهت بهره‌گیری در پیوندهای بی‌سیم می‌باشد. شکل ۶-۱۳ یک قاب داده ی پروتکل 802.11 را نشان می‌دهد. اعداد بالای هر یک از دامنه‌های قاب به مفهوم طول آن دامنه بر مقیاس تعداد بایت می‌باشند. اعداد بالای هر زیر دامنه در محدوده‌ی کنترل قاب به مفهوم طول هر زیر دامنه بر مبنای بیت می‌باشد. اکنون دامنه‌ها در قاب و نیز برخی از زیر دامنه‌های مهم کنترل قاب تحلیل می‌شوند.

- **دامنه‌های CRC و محموله‌ی پیام**

در هسته‌ی قاب، محموله‌ی پیام جای دارد که به صورت انحصاری شامل یک داده‌گرام IP و یک بسته‌ی ARP می‌باشد. اگرچه این دامنه می‌تواند طول حداکثر 2312 بایت را اختیار نماید اما معمولاً کمتر از 1500 بایت می‌باشد. مانند قاب اترنت یک فریم 802.11 شامل یک کد سنجش دوره‌ی افزونگی (CRC) به طول 32 بیت است که از طریق آن گیرنده می‌تواند بیت‌های خطا را در قاب دریافتی آشکارسازی نماید. بدیهی است وقوع بیت‌های خطا در شبکه‌های محلی بی‌سیم متداول‌تر از شبکه‌های کابلی محلی است و بنابراین بهره‌گیری از CRC در این شبکه‌ها سودمندتر است.

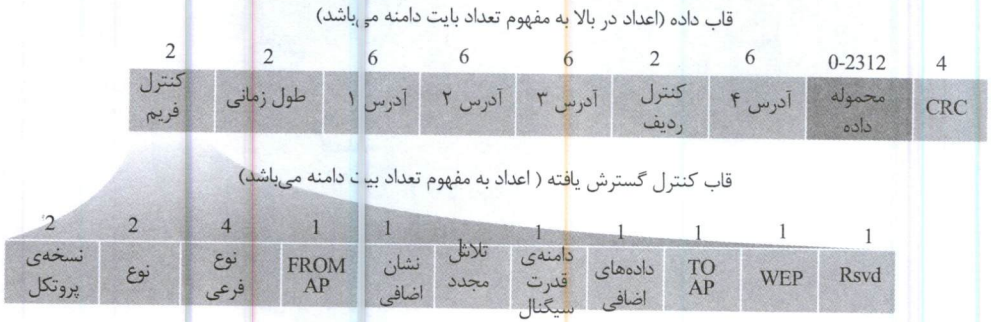
- **دامنه‌های آدرس**

شاید یکی از برجسته‌ترین تفاوت‌های قاب 802.11 آن باشد که دارای **چهار** دامنه‌ی آدرس هر یک با قابلیت نگهداری یک آدرس MAC به طول 6 بایت می‌باشد. اما چرا تعداد چهار دامنه‌ی آدرس؟ آیا دو آدرس MAC مربوط به مبدا و مقصد مانند اترنت کافی نمی‌باشد؟ آنچه مشخص شده است نیاز به سه دامنه‌ی آدرس برای پیاده‌سازی اهداف عملیات بین شبکه می‌باشد. یکی از این موارد جابه‌جایی داده‌گرام لایه‌ی شبکه از یک پایانه‌ی بی‌سیم از درون یک AP به یک واسط مسیریاب می‌باشد. از دامنه‌ی آدرس چهارم هنگامی استفاده می‌شود که نقاط دست‌یابی (AP) قاب‌ها را به یکدیگر در قالب روش ad-hoc یا مستقل مبادله می‌کنند. از آنجائیکه در این بخش از فصل فقط شبکه‌های زیرساخت بررسی می‌شوند بنابراین نقطه‌ی تمرکز را بر روی نخستین سه دامنه‌ی آدرس MAC قرار می‌دهیم. استاندارد پروتکل 802.11 این سه دامنه را به گونه‌ی ذیل تعریف می‌کند:

۱- آدرس 2 عبارت است از یک آدرس MAC از پایانه‌ای که قاب گسیل می‌کند. بنابراین اگر یک پایانه‌ی بی‌سیم فریمی را ارسال نماید آدرس MAC آن پایانه در دامنه‌ی شماره 2 از آدرس‌ها وارد می‌شود. به شکل مشابه اگر یک AP قابی را گسیل نماید، آدرس MAC آن AP نیز در دامنه‌ی آدرس 2 وارد می‌شود.

۲- آدرس 1 عبارت است از یک آدرس MAC پایانه‌ی بی‌سیم که قاب داده دریافت می‌دارد. بنابراین اگر یک پایانه‌ی بی‌سیم سیار (متحرک) فریمی را ارسال نماید، دامنه‌ی آدرس شماره 1 حامل آدرس MAC نقطه دست‌یابی (AP) مقصد می‌باشد. به شکل مشابه اگر یک AP قاب داده‌ای را گسیل نماید، آنگاه آدرس شماره‌ی 1 در درون خود آدرس MAC از پایانه‌ی بی‌سیم مقصد را حمل می‌کند.

۳- جهت درک مفهوم آدرس 3 یادآوری این نکته ضروری است که BSS که شامل AP و میزبان‌های بی‌سیم می‌باشد و بخشی از یک زیرشبکه به شمار می‌آید از طریق واسط مسیریاب به دیگر زیرشبکه‌ها متصل می‌شود. آدرس شماره 3 حامل آدرس MAC واسط این روتر می‌باشد.



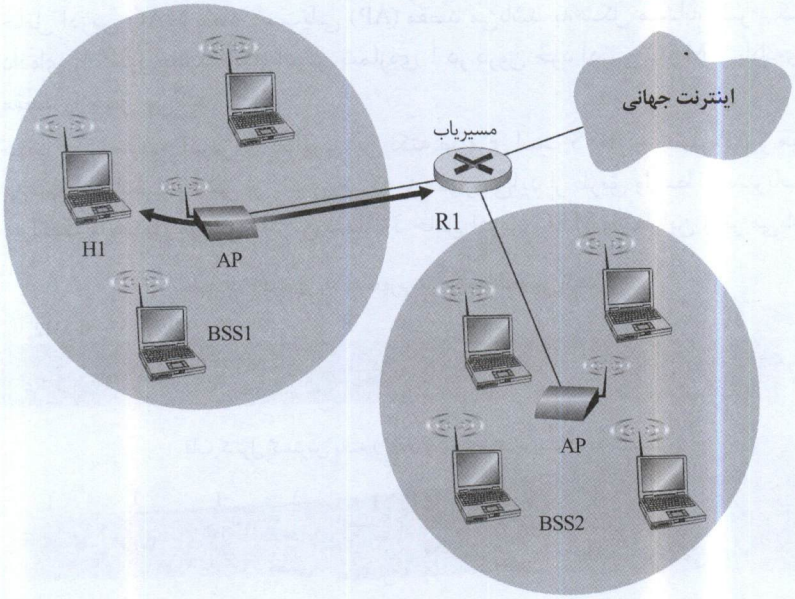
شکل ۶-۱۳ ♦ قاب داده‌ی **802.11**

برای درک عمیق اهداف آدرس شماره‌ی 3 به تحلیل یک مثال در چارچوب شکل ۶-۱۴ می‌پردازیم. در این شکل دو نقطه‌ی دست‌یابی (AP) وجود دارد که هر یک از آن‌ها تعدادی میزبان بی‌سیم را پشتیبانی می‌کند. هر AP دارای یک ارتباط مستقیم با مسیریاب است که AP را از طریق آن به اینترنت جهانی متصل می‌کند. بدیهی است یک نقطه‌ی دست‌یابی AP یک وسیله‌ی لایه‌ی پیوند است و بنابراین از تفسیر آدرس‌ها و داده‌گرام‌های IP ناتوان است. حال فرض کنید که یک داده‌گرام از واسط R1 در مسیریاب به پایانه‌ی بی‌سیم H1 منتقل می‌شود. روتر از وجود AP بین خود و H1 آگاهی ندارد. به عبارت دیگر H1 فقط یک میزبان در یکی از زیرشبکه‌هایی است که با روتر در ارتباط می‌باشند:

۱- مسیریاب شکل ۶-۱۴ از آدرس IP متعلق به H1 از طریق آدرس مقصد داده‌گرام IP آگاه است و بنابراین از مدول ARP جهت دست‌یابی آدرس MAC آن میزبان یعنی H1 همانند یک شبکه‌ی محلی اترنت عادی بهره می‌گیرد. پس از آن که آدرس فیزیکی MAC متعلق به H1 را دست‌یابی نمود، واسط R1 در مسیریاب داده‌گرام IP را در قالب یک فریم اترنت بسته‌بندی می‌کند. دامنه‌ی آدرس منبع در این قاب داده شامل آدرس فیزیکی MAC واسط R1 و دامنه‌ی آدرس مقصد آن نیز حامل آدرس MAC میزبان H1 می‌شود.

۲- هنگامی که قاب اترنت وارد AP می‌شود، نقطه‌ی دست‌یابی قاب اترنت 802.3 را قبل از گسیل به کانال بی‌سیم به یک فریم 802.11 تبدیل می‌کند. پس از آن AP آدرس‌های 1 و 2 را از اطلاعات آدرس MAC میزبان H1 و آدرس MAC خود به ترتیب پر می‌کند. برای آدرس شماره 3، نقطه‌ی دست‌یابی AP آدرس MAC واسط R1 را وارد می‌کند. از این طریق H1

می‌تواند با بهره‌گیری از آدرس3، آدرس MAC واسط روتر را که داده‌گرام را به زیرشبکه منتقل می‌نماید، دست‌یابی کند.



شکل ۶-۱۴ ♦ استفاده از دامنه‌ی آدرس‌ها در قاب‌های **802.11**. ارسال قاب بین **H1** و **R1**

اکنون حالتی را در نظر بگیرید که میزبان بی‌سیم H1 با جابه‌جایی یک داده‌گرام از H1 به R1 از خود واکنش نشان می‌دهد:

۱- میزبان H1 یک قاب 802.11 را خلق می‌کند و دامنه‌های آدرس‌های 1و 2 را با آدرس‌های MAC متعلق به AP و میزبان H1 به گونه‌ای که در بالا تشریح گردید، به ترتیب پر می‌کند. برای آدرس 3 میزبان H1 آدرس MAC متعلق به R1 را وارد می‌کند.

۲- هنگامی که AP قاب 802.11 را دریافت می‌دارد، آن را تبدیل به یک فریم اترنت می‌کند. دامنه‌ی آدرس مبدا برای این قاب آدرس MAC متعلق به H1 می‌باشد و دامنه‌ی آدرس مقصد مربوط به آدرس MAC واسط R1 می‌شود. بنابراین آدرس 3 این امکان را به AP می‌دهد که آدرس MAC مقصد مناسب را به هنگام ساختن فریم اترنت، شناسایی نماید. به عبارت دیگر آدرس شماره‌ی 3 نقش حیاتی را در ایجاد ارتباط بین BSS و شبکه‌ی محلی بی‌سیم بازی می‌کند.

- **شماره ردیف (SN)، طول زمانی و دامنه‌های کنترل قاب**

یادآور می‌شود در پروتکل 802.11 هرگاه یک پایانه یک فریم سالم را از میزبان دیگری دریافت نماید یک پیام کوتاه ACK را در پاسخ بازگشت می‌دهد. از آنجائیکه پیام ACK ممکن است مفقود گردد، پایانه‌ی ارسال کننده ممکن است نسخه‌های متعددی از یک فریم را گسیل نماید. همانگونه که در بحث پروتکل rdt2.1 در فصل سوم ملاحظه شد، بهره‌گیری از شماره ردیف این امکان را فراهم می‌آورد که یک گیرنده قادر به تمایز بین یک فریم تازه ارسال شده و انتقال مجدد یک فریم از پیش ارسال شده، باشد. دامنه‌ی شماره ردیف در قاب 802.11 نیز همان هدف را در لایه‌ی پیوند دنبال می‌کند که در لایه‌ی انتقال فصل سوم عمل می‌نمود.

پروتکل 802.11 اجازه می‌دهد که پایانه‌ی فرستنده کانالی را جهت انتقال برای مدت زمان معین که شامل زمان انتقال فریم داده و نیز زمان برگشت پیام ACK می‌شود، ذخیره‌سازی (رزرو) نماید. این ارزش زمانی در درون دامنه‌ی طول زمانی قاب برای هر دو حالت RTS و CTS تعریف می‌شود.

همان‌گونه که در شکل ۶-۱۳ نشان داده شده است دامنه‌ی کنترل قاب شامل تعداد قابل توجهی از زیر دامنه‌ها می‌باشد. برخی از این زیردامنه‌های مهم عبارتند از:

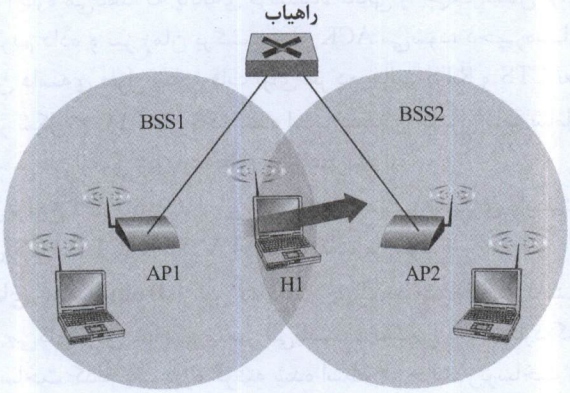
- **نوع و زیرنوع (Subtype)**. از این دامنه‌ها جهت تمیزی وابستگی یا شرکت‌پذیری قاب‌های RTS، CTS، ACK، DATA استفاده می‌شود.
- **دامنه‌های to و یا from**. از این دو دامنه جهت تعریف مفاهیم دامنه‌های مختلف آدرس استفاده می‌شود. این مفاهیم متغیر می‌باشند و بستگی به آن دارد که از روش‌های ad-hoc و یا زیرساخت کدام یک بهره گرفته شده است. در حالت زیرساخت نیز این مفاهیم وابسته به این موضوع می‌باشند که آیا میزبان بی‌سیم و یا یک AP قاب ارسال نموده است؟
- **دامنه‌ی WEP** که از آن جهت اشاره به این موضوع که آیا از روش رمزگذاری امنیتی استفاده شده است یا خیر بهره‌گیری می‌شود. موضوع WEP در فصل ۸ بررسی شده است.

۶-۷ تحرک و جابه‌جایی در یک زیرشبکه‌ی IP مشابه

برای افزایش محدوده‌ی فیزیکی یک شبکه‌ی محلی بی‌سیم، شرکت‌ها و دانشگاه‌ها در اغلب موارد از چندین BSS در درون زیرشبکه‌های با IP مشابه استفاده می‌کنند. این مهم سبب دامن زدن به موضوع تحرک در بین BSS ها می‌شود. به عبارت دیگر چگونه پایانه‌های بی‌سیم به گونه‌ای مشابه و همانند از یک BSS به دیگری جابه‌جا می‌شوند در حالی که یک تعامل TCP در جریان را برقرار نگهداری می‌کنند؟

در این زیربخش ملاحظه خواهد شد که عامل تحرک بین BSSها را می‌توان از طریق به کارگیری روش‌های نسبتاً ساده در حالی که BSSها بخشی از یک زیرشبکه می‌باشند تحقق بخشید. هنگامی که پایانه‌ها بین زیرشبکه‌ها جابه‌جا می‌شوند، پروتکل‌های مدیریت تحرک پیشرفته پیچیده‌تری مانند آنچه در بخش‌های بعدی همین فصل یعنی ۶ – ۱۰ و ۶ – ۱۱ مورد مطالعه قرار گرفته‌اند، ضروری می‌باشد.

در حال حاضر یک نمونه ویژه از تحرک بین BSSها در زیرشبکه‌ای مشابه تحلیل می‌شود. شکل ۶-۱۵ دو BSS متصل به یکدیگر را با یک میزبان H1 نشان می‌دهد که در حال جابه‌جایی از BSS1 به BSS2 می‌باشد. از آنجائیکه در این مثال وسیله‌ی ارتباطی بین دو BSS یک مسیریاب نمی‌باشد، در نتیجه همه‌ی پایانه‌ها در دو BSS که شامل نقاط دست‌یابی (AP) نیز می‌شوند متعلق به زیرشبکه‌ی IP مشابه می‌باشند. بنابراین هنگامی که H1 از BSS1 به سمت BSS2 حرکت می‌کند، این پایانه می‌تواند آدرس IP و همگی ارتباط‌های در جریان TCP خود را جاری نگاه دارد. اما اگر وسیله‌ی ارتباطی بین آن‌ها یک مسیریاب باشد، آنگاه پایانه‌ی H1 نخست بایستی یک آدرس IP جدید را در زیرشبکه‌ای که درون آن در حال تحرک است، دست‌یابی نماید. این تغییر آدرس سبب اختلال و سرانجام قطع هر ارتباط TCP در جریان در H1 می‌شود. در بخش ۶-۱۱ ملاحظه خواهد شد که چگونه یک پروتکل مدیریت تحرک لایه‌ی شبکه مانند پروتکل متحرک IP می‌تواند به منظور ممانعت از وقوع چنین وضعیتی به کار گرفته شود.



شکل ۶-۱۵ ♦تحرک در زیرشبکه‌ی مشابه

اکنون این پرسش مطرح می‌شود، چه اتفاق ویژه ای به هنگام حرکت H1 از BSS1 به BSS2 روی می‌دهد؟ پاسخ آن است هنگامی که H1 آغاز به حرکت می‌کند و به تدریج از AP1 فاصله می‌گیرد، سیگنال‌های روبه ضعف را از سوی AP1 شناسایی و دریافت می‌کند و به دنبال آن آغاز به ردیابی سیگنال‌های قوی‌تر می‌کند. در هنگام انجام این فعالیت H1 قاب‌های ردیابی (بیکن) را از AP2 دریافت می‌کند که در اغلب شرکت‌ها و دانشگاه‌ها از نقطه‌نظر SSID مشابه AP1 می‌باشد. پس از آن H1 خود را از وابستگی به AP1 می‌رهاند و وارد شرکت‌پذیری با AP2 می‌شود و این در حالی است که آدرس IP خود را و نیز تعامل در جریان TCP را حفظ و نگهداری می‌کند.

این رویکرد چالش جابه‌جایی بین میزبان و AP را برطرف می‌سازد. اما راه‌یاب شکل ۶-۱۵ چه واکنشی از خود نشان می‌دهد؟ چگونه از جابه‌جایی میزبان از یک نقطه‌ی دست‌یابی به نقطه‌ی دست‌یابی به نقطه‌ی دیگر آگاه می‌شود؟ در فصل ۵ ملاحظه شد یکی از ویژگی‌های راه‌یاب‌ها خود یادگیری آن است که سبب پیکربندی و تعریف خودکار جداول گسیل جلو در آن‌ها می‌شود. این ویژگی خودآموزی با ظرافت ویژه‌ای جابه‌جایی و تحرک کاربران را پاسخگویی می‌کند. اما نکته‌ی ضروری دیگر آن است که راهیاب‌ها برای پشتیبانی کاربرانی که مرتباً و. به سرعت در حال تعویض می‌باشند و نیازمند حفظ و نگهداری ارتباط‌های TCP خود به هنگام تحرک در بین BSSها می‌باشند، طراحی نشده‌اند. جهت درک بهتر این مهم توجه به این نکته ضروری است که قبل از جابه‌جایی، راه‌یاب اطلاعات مربوط به جفت نمودن آدرس MAC میزبان H1 با واسط خروجی راه‌یاب را در جدول گسیل به جلو خود نگهداری می‌کند تا از طریق آن واسط، دسترسی به H1 امکان‌پذیر گردد. اگر H1 در ابتدا در BSS1 باشد، آنگاه داده‌گرام به مقصد H1 از طریق AP1 گسیل می‌شود. هنگامی که H1 خود را وابسته به BSS2 می‌کند قاب‌های آن بایستی به سوی AP2 هدایت شوند. یک راه‌کار آن است که AP2 یک فریم اترنت فراپخش که آدرس مبدا H1 در آن حمل می‌شود را بی‌درنگ پس از تعریف وابستگی جدید به سوئیچ گسیل نماید. هنگامی که راه‌یاب این قاب را دریافت کند، جدول گسیل جلوی خود را بر اساس آن به روزرسانی می‌کند و این امکان را تعریف می‌کند که میزبان H1 از طریق نقطه‌ی دست‌یابی AP2 قابل دسترسی باشد. گروه استاندارهای 802.11 در حال تکمیل پروتکل‌های بین AP می‌باشد که چالش‌های مربوط به این موارد و موضوعات وابسته را در آینده برطرف می‌سازد.

در ادامه‌ی مطالب مربوط به مجموعه‌ی پروتکل‌های 802.11 دو توان‌مندی مهم که در شبکه‌های

802.11 وجود دارند، بررسی و تحلیل می‌شوند. این توان‌مندی‌ها کاملاً در استاندارد 802.11 تعریف نشده‌اند بلکه از طریق روش‌های تعریف شده در این استانداردها قابلیت پیاده‌سازی دارند. این سبب می‌شود تولید کنندگان مختلف این امکانات را بنابر اراده و میل خود در تجهیزاتی که طراحی و تولید می‌کنند پیاده‌سازی نمایند و در نتیجه بازار رقابت پر رونقی را به وجود آورند. این دو توان‌مندی عبارتند از:

۱- گزینش اختیاری سرعت انتقال 802.11

بنابر شکل ۶-۳ انواع گوناگون روش‌های مدولاسیون سرعت‌های متنوعی را می‌توانستند برای ارزش‌های گوناگون SNR فراهم آورند. به عنوان مثال یک کاربر سیار 802.11 که در ابتدا 20 متر از پایانه‌ی مبنا فاصله دارد را با یک SNR قوی تصور کنید. با توجه به این موضوع که اندازه‌ی SNR بالا می‌باشد کاربر از طریق بهره‌گیری از یک روش مدولاسیون لایه‌ی فیزیکی می‌تواند سرعت انتقال چشمگیری را در حالی فراهم آورد که همزمان نرخ BER نیز پائین باشد.

در این مرحله ظاهراً دشواری وجود ندارد. اکنون فرض کنید که کاربر در حرکت باشد و به تدریج از پایانه‌ی مبنا همراه با ارزش SNR در حال کاهش دور شود. در این حالت اگر روش مدولاسیون پروتکل 802.11 در بین کاربر و پایانه‌ی مبنا تغییر نکند نرخ BER به سرعت افزایش یافته و این تغییر با کاهش هرچه بیشتر SNR رشد فزاینده‌ای را به خود می‌گیرد و سرانجام قاب‌های داده سالم دریافت نخواهند شد.

به همین دلیل برخی از پیاده سازی‌های 802.11 از توان‌مندی گزینش اختیاری سرعت بهره‌مند می‌باشند. این خودسازگاری در سرعت انتقال گزینش روش مدولاسیون لازم برای لایه‌ی فیزیکی را به همراه دارد. گزینش روش مدولاسیون بر اساس ویژگی کانال اخیر و یا جاری مورد استفاده می‌باشد. به عنوان مثال پروتکل Lucent WaveLan – II 802.11b نرخ انتقال چندگانه را برای قاب‌های داده فراهم می‌آورد. اگر یک گره دو قاب را به صورت پی‌درپی بدون دریافت پیام ACK ارسال نماید که مفهوم آن وجود خطای بیت در کانال است آنگاه سرعت انتقال به نرخ پائین بعدی تنزل می‌یابد. اگر تعداد 10 قاب داده‌ی پی‌درپی ارسال گردد و پیام تائید ACK نیز دریافت شود و یا اگر زمان‌شمار پروتکل که زمان آخرین کاهش سرعت را سنجش می‌کند به انتهای خود نزدیک شود آنگاه در هر یک از این دو حالت سرعت انتقال به نرخ بالاتر بعدی افزایش می‌یابد. این روش گزینش اختیاری سرعت مشابه الگوی رهگیری کنترل تراکم TCP می‌باشد که در هنگام وجود شرایط نامطلوب (دریافت پیام‌های ACK) نرخ انتقالی تا زمانی افزایش می‌یافت که شرایط نامطلوب شود، (پیام NACK دریافت گردد). به عبارت دیگر هنگامی که وضعیت نامطلوبی روی دهد، سرعت کاهش می‌یابد. گزینش اختیاری سرعت 802.11 و کنترل تراکم TCP را می‌توان شبیه به یک کودک دانست که والدین خود را بیشتر و بیشتر در معرض فشار جهت خواسته‌ای قرار می‌دهد که سرانجام به دلیل محدودیت برای آن‌ها با پاسخ مشابه "دیگر کافی است" روبه‌رو می‌شود. در نتیجه‌ی شنیدن پیام اعتراض والدین خود کودک تا مدتی عقب‌نشینی می‌کند و هنگامی که حِس کند شرایط مطلوب است تلاش برای خواسته‌ی خود را مجدداً آغاز می‌کند.

۲-مدیریت توان (قدرت)

قدرت یک منبع، سرمایه‌ی بسیار با ارزشی در تجهیزات سیار می‌باشد. بنابراین استاندارد 802.11 توان‌مندی مدیریت توان را پشتیبانی می‌کند که به گره‌های 802.11 اجازه می‌دهد زمان مورد نیاز خود برای عملیات حِس، انتقال، دریافت و نیز دیگر مدارهای لازم روشن را در چندین وضعیت به حداقل

ممکن کاهش و مدیریت نمایند. عملیات مدیریت قدرت 802.11 به گونه‌ی ذیل اعمال می‌شود. به صورت انحصاری مابین دو وضعیت خواب (sleep) و بیدار (wake) تغییر می‌کند. بدیهی است این تغییر مانند یک دانشجوی خواب‌آلوده در کلاس درس نمی‌باشد! یک گره از طریق تغییر ارزش **بیت مدیریت قدرت** در سرپیام قاب 802.11 به مقدار 1، نقطه‌ی دست‌یابی (AP) را از وضعیت به خواب رفتن خود آگاه می‌کند. یک زمان‌شمار (تایمر) در گره فعال می‌شود تا بی‌درنگ قبل از آنکه AP قاب ردیابی خود را ارسال کند آن گره را بیدار نماید. یادآوری می‌شود یک AP در حالت عادی هر 100 میلی‌ثانیه یک قاب ردیابی (بیکن) ارسال می‌کند. از آنجائیکه AP با مشاهده‌ی ارزش 1 در بیت انتقال قدرت می‌داند که آن گره به خواب رفته است بنابراین متوجه این موضوع می‌باشدکه هیچ‌گونه فریمی را نبایستی به آن گره گسیل نماید و در نتیجه هر قاب داده‌ی ارسالی به مقصد میزبان خواب رفته را جهت انتقال در زمان بعدی در حافظه‌ی خود نگهداری می‌کند.

یک گره دقیقاً قبل از آنکه AP فریم ردیابی را ارسال نماید بیدار می‌شود و بی‌درنگ وارد وضعیت کاملاً فعال می‌شود. بدیهی است فرصت بیدار و فعال شدن فقط 250 میکروثانیه است ومانند دانشجوی خواب‌آلوده نیست که چندین دقیقه لازم دارد تا از خواب‌آلودگی خلاصی یافته و کاملاً هوشیار شود! قاب‌های ردیابی که از سوی AP گسیل می‌شوند حامل فهرستی از گره‌ها می‌باشند که قاب‌های داده‌ی آن‌ها در AP ذخیره شده‌اند. اگر قاب‌های ذخیره شده‌ای برای گره وجود نداشته باشد آن گره می‌تواند مجدداً به حالت خواب بازگشت نماید، در غیر این‌صورت گره به گونه‌ای انحصاری تقاضای ارسال قاب‌های ذخیره شده را از طریق ارسال پیام نظرخواهی (Polling) به AP خواهد نمود. با توجه به زمان تناوب 100 میلی‌ثانیه برای پیام ردیابی، زمان 250 میکروثانیه جهت بیدار و فعال شدن و نیز یک زمان بسیار کوتاه جهت دریافت و سنجش قاب ردیابی و حصول اطمینان از عدم قاب‌های ذخیره شده‌ی یک گره که فریمی جهت ارسال و یا دریافت ندارد، می‌تواند 99% از میزان توان مورد نیاز خود در خواب باشد و بنابراین صرفه‌جویی بسیار چشمگیر در مصرف انرژی باطری آن حاصل می‌شود. به عبارت دیگر از کل زمان نهفتگی باطری مقدار 99% از آن همواره در اثر خواب بودن گره صرفه‌جویی می‌شود.

۶-۸ فراتر از پروتکل‌های 802.11: بلوتوث و WiMAX

همان‌گونه که در شکل ۶-۲ نشان داده شده، استاندارد IEEE 802.11 برای شبکه‌های WiFi با هدف ارتباط بین تجهیزات با فاصله‌ی حداکثر 100 متر طراحی شده است. بدیهی است این فاصله می‌تواند در حالتی که از 802.11 در یک پیکربندی نقطه به نقطه با بهره‌گیری از آنتن‌های جهت‌دار استفاده می‌شود، گسترش بیشتری داشته باشد. دو پروتکل دیگر IEEE 802 عبارتند از پروتکل بلوتوث (Bluetooth) که در استاندارد [IEEE 802.15 2009] تعریف شده است و دیگری پروتکل WiMAX بر مبنای دو نسخه استاندارد: [IEEE 802.16d 2004,IEEE 802.16e 2005] می‌باشد. این دو پروتکل در قالب استانداردهای خود جهت انجام عملیات ارتباطی به ترتیب در فواصل کوتاه‌تر و بلندتر از 100 متر طراحی شده‌اند.

- **بولوتوث**

شبکه‌ای که بر مبنای پروتکل IEEE 802.15.1 پیاده‌سازی شده باشد دارای عملکردی در فاصله‌ی کوتاه، توان مصرفی پائین و هزینه‌ی اندک می‌باشد. این فناوری جایگزین کابلی اتصال تجهیزاتی مانند کامپیوترهای همراه، تجهیزات جانبی، گوشی‌های سلولی همراه و PDA ها با هزینه‌ی مصرف، بُرد و

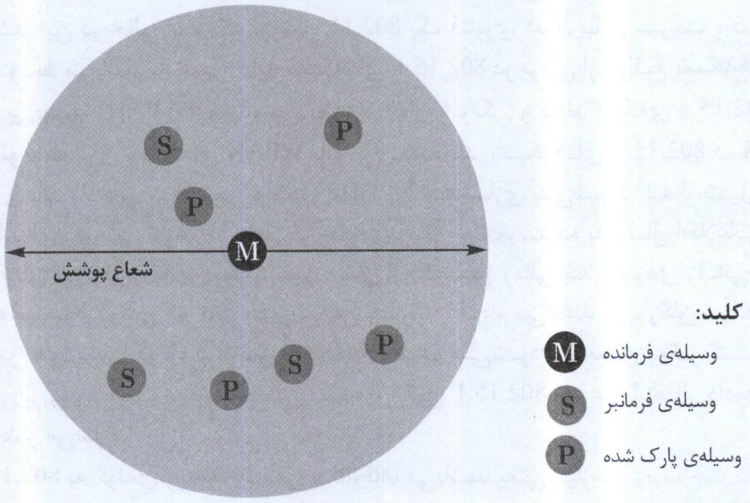
سرعت پایین می‌باشد. این در حالی است که پروتکل 802.11 یک فناوری دست‌یابی سرعت و قدرت مصرف بالا با برد متوسط می‌باشد. به همین دلیل شبکه‌های 802.15.1 در برخی از مواقع **شبکه‌های ناحیه‌ای شخصی بی‌سیم**[1] **(WPAN)** نامیده می‌شوند. لایه‌های فیزیکی و پیوند شبکه‌ی 802.15.1 بر مبنای مشخصات **بلوتوث** برای شبکه‌های WPAN پایه‌ریزی شده‌اند. شبکه‌های 802.15.1 در قالب روش TDM و پهنای باند رادیویی بدون نیاز به مجوز 2.4 GHz پیاده‌سازی می‌شوند که از شیارهای زمانی 625 میکروثانیه‌ای بهره می‌گیرند. در طول هر حفره‌ی زمانی یک فرستنده به ارسال اطلاعات یک کانال از تعداد 79 کانال می‌پردازد. با این روش، تغییر کانال از یک شیار زمانی به حفره‌ی زمانی بعد کاملاً تصادفی انجام می‌شود. روشی که کانال‌ها پرش بین شیارها را انجام می‌دهند در واژگان شبکه‌های بی‌سیم **طیف گسترده‌ی پرش کوتاه فرکانس**[2] **(FHSS)** نامیده می‌شود. به بیان دیگر گسترش انتقال در بستر زمان بر روی طیف فرکانسی می‌باشد. پروتکل 802.15.1 سرعت انتقال داده‌ی تا 4 Mbps را امکان‌پذیر می‌سازد.

شبکه‌های 802.15.1 به گونه‌ی روزمره مستقل (ad-hoc) می‌باشند یعنی: نیازمند زیرساخت شبکه مانند یک نقطه‌ی دست‌یابی (AP) جهت ارتباط تجهیزات 802.15.1 نمی‌باشند. بنابراین این‌گونه تجهیزات خود مجبور به سازمان‌دهی یکدیگر می‌شوند. چگونگی انجام این سازمان‌دهی به این گونه است که تجهیزات 802.15.1 نخست در قالب یک **شبکه‌ی ذره‌بینی**[3] شامل حداکثر 8 وسیله‌ی فعال مانند آنچه در شکل ۶-۱۶ نشان داده شده است، سازمان‌دهی می‌شوند. یکی از این تجهیزات نقش هدایت کننده‌ی اصلی و 7 گره‌ی دیگر نقش فرمان‌بر را بازی می‌کنند. گره‌ی فرمانده بر سراسر شبکه‌ی ذره‌بینی حاکم است، ساعت آن در این شبکه زمان و فواصل ان را تعریف می‌کند. این گره می‌تواند انتقال را در هر یک از شیارهای شماره‌ی فرد انجام دهد و یک فرمان‌بر می‌تواند فقط پس از ارتباط فرمانده با آن در شیار قبلی مبادرت به انتقال نماید و حتی پس از آن نیز فرمان‌بر فقط قادر به ارتباط با فرمانده می‌باشد. علاوه بر گره‌ی فرمان‌بر حداکثر تعداد 255 وسیله‌ی پارک شده نیز می‌توانند در شبکه وجود داشته باشد. این تجهیزات تا هنگامی‌که وضعیت آن‌ها از حالت پارک به فعال از سوی فرمانده تغییر نکند، نمی‌توانند از امکانات ارتباطی شبکه بهره‌مند شوند. علاقه‌مندان به مطالعه‌ی بیشتر در این زمینه می‌توانند به سامانه‌ی رسمی IEEE مراجعه نمایند.

- **شبکه‌های WiMAX**

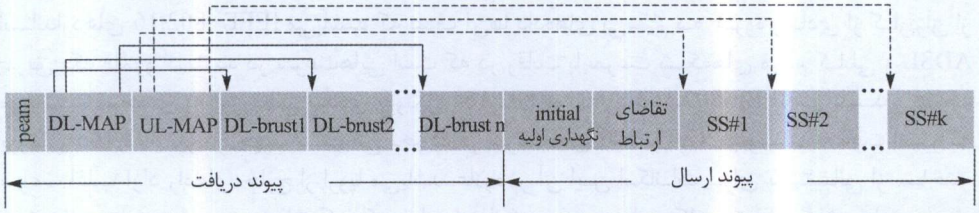
شبکه‌های **ارتباط جهانی برای دست‌یابی امواج مایکروویو**[4] **(WiMAX)** خانواده‌ای از استانداردهای IEEE 802.16 می‌باشند که هدف آن‌ها داده‌های بی‌سیم به گروه زیادی از کاربران از طریق یک قلمرو گسترده در سرعت‌هایی است که در رقابت با سرعت شبکه‌های مودم کابلی و ADSL می‌باشند. نسخه ی به روزرسانی شده‌ی پروتکل 802.16a استاندارد IEEE 802.16d می‌باشد و هدف پروتکل 802.16e نیز در جهت پشتیبانی تحرک در سرعت‌های 100 تا 120 کیلومتر در ساعت است که سرعت اغلب آزاد راه‌های خارج از اروپا می‌باشد. علاوه بر آن این امکانات برای پشتیبانی از ساختار پیوندهای متنوع ویژه‌ی تجهیزات کوچک و با منابع انرژی محدود مانند کامپیوترها و گوشی‌های همراه و تبلت‌ها می‌باشد.

1- Wireless personal area networks
2- Frequency hopping spread spectrum
3- Piconet
4- World interoperability for microwave access



شکل ۶-۱۶ یک شبکه‌ی ذره‌بینی بلوتوث

ساختار 802.16 بر این مبنا پایه‌گذاری شده است که یک پایانه‌ی مبنا به صورت متمرکز خدمات را برای تعداد انبوهی از مشتری‌های بالقوه که به آن‌ها پایانه‌های مشترک گفته می‌شود، فراهم می‌آورد. بدیهی است این مشترکین همگی وابستگی به آن پایانه‌ی مبدا دارند. با این بیان پروتکل WiMAX مانند هر دو شبکه‌ی WiFi از لحاظ روش زیرساخت و شبکه‌های تلفنی سیار (سلولی) می‌باشد. بر مبنای ساختار قاب TDM نشان داده شده در شکل ۶-۱۷، پایانه‌ی مبنا هماهنگی انتقال بسته‌های لایه‌ی پیوند در هر دو جهت گسیل یعنی از گره‌ی مبنا به پایانه‌های مشترکین (downstream) و از گره‌های مشترکین به پایانه‌ی مبنا (upstream) را انجام می‌دهد. در این زیر بخش از واژه‌ی **بسته** جایگزین **قاب** که در پروتکل 802.11 و دیگر بسته‌های لایه‌ی پیوند استفاده می‌شد جهت تمایز بین جزء داده در لایه‌ی پیوند و ساختار قاب TDM نشان داده شده در شکل ۶-۱۷، بهره گرفته می‌شود. بنابراین پروتکل WiMAX به گونه‌ی مالتی پلکس تقسیم زمان عمل می‌کند اگرچه زمان قاب‌بندی بنابر آنچه در ادامه توضیح داده می‌شود متغیر است. بدیهی است WiMAX از روش تسهیم تقسیم فرکانس (FDM) نیز بهره می‌گیرد. این روش در فصل‌های پیشین به اختصار تحلیل شده است.



شکل ۶-۱۷ ساختار قاب TDM در پروتکل 802.16

در آغاز فریم، پایانه‌ی مبنا ابتدا فهرستی از **پروتکل‌های دست‌یابی رسانه (MAP)** را از جانب خود به سوی گره‌های مشترکین در قالب یک پیام گسیل می‌کند. از این طریق گره‌ی مبنا پایانه‌های مشترک را از ویژگی‌های لایه‌ی فیزیکی شامل روش‌های مدولاسیون، رمزگذاری و عوامل تصحیح خطا آگاه می‌کند. ارسال این اطلاعات به مفهوم آن است که انتقال زنجیره‌ی بسته‌ها در درون قاب چگونه

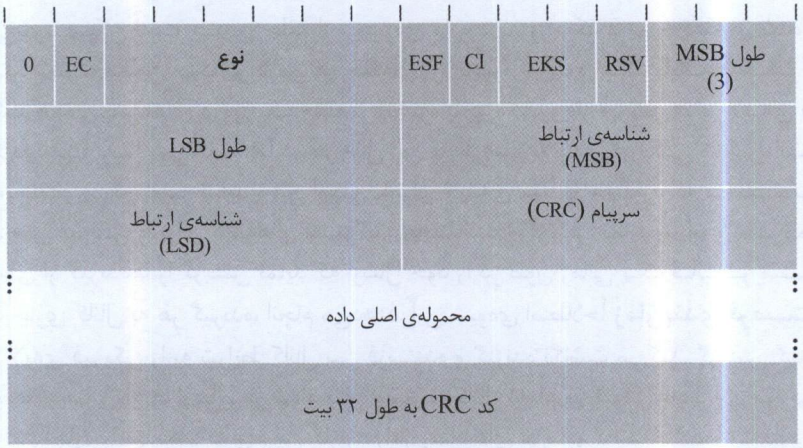
انجام می‌شود. ممکن است چندین حلقه از زنجیره‌ی داده در درون یک فریم جایگذاری شده باشد و یا انکه چندین بسته ممکن است در درون هر حلقه‌ای از زنجیره به سوی یک پایانه‌ی مشترک گسیل شده باشد. همه‌ی بسته‌های درون یک حلقه از زنجیره با بهره‌گیری از ویژگی‌های لایه‌ی فیزیکی از سوی گره‌ی مبنا ارسال می‌شوند. اما در هر حال این ویژگی‌های لایه‌ی فیزیکی ممکن است از یک حلقه به حلقه‌ی دیگر تغییر نماید و این امکان را برای پایانه‌ی مبنا فراهم آورد که مناسب‌ترین شیوه‌ی انتقال لایه‌ی فیزیکی را برای پایانه‌های مشترک دریافت کننده گزینش نماید. پایانه‌ی مبنا ممکن است مجموعه‌ای از گیرنده‌ها را گزینش نماید که ارسال خود را در طول زمانی یک قاب بر مبنای حدس وضعیت جاری کانال به هر گیرنده، انجام می‌دهند. این شیوه‌ی اصطلاحاً **زمان‌بندی فرصت‌طلبانه**[1] پروتکل لایه‌ی فیزیکی را به شرایط کانال بین فرستنده و گیرنده انطباق می‌کند. گزینش گیرنده‌هایی که بسته‌ها به سوی آن‌ها ارسال می‌شوند بنابر وضعیت کانال انجام می‌شود. رعایت این موارد به پایانه ی مبنا این اجازه را می‌دهد که بهترین کاربرد را از رسانه‌های بی‌سیم داشته باشد. استاندارد WiMAX مجموعه‌ای از عوامل ویژه را به لایه‌ی فیزیکی که در یک وضعیت مفروض بایستی مورد استفاده قرار گیرد، تحمیل نمی‌کند. این تصمیم در اختیار تولید کننده‌ی تجهیزات WiMAX و کارگزاران شبکه گذارده شده است.

یک پایانه‌ی مبنا در WiMAX هم چنین دست‌یابی پایانه‌ی مشترک را به کانال ارسال به طرف بالا از طریق بهره‌گیری از پیام‌های **UL – MAP**[2] هماهنگ می‌کند. این پیام‌ها مقدار زمان اختصاص یافته به هر پایانه‌ی مشترک را جهت دست‌یابی در زیر فریم‌های UL که به دنبال یکدیگر جاری می‌شوند، کنترل می‌کند. بار دیگر استاندارد WiMAX، اجباری را جهت هیچ سیاست‌گذاری ویژه‌ای برای اختصاص زمان کانال UL به یک پایانه‌ی مشتری تحمیل نمی‌کند به عبارت دیگر تصمیم بر عهده‌ی کارگزار شبکه است. اما WiMAX شیوه‌ای مانند پیام‌های کنترل UL – MAP را فراهم می‌آورد که از طریق آن پیاده‌سازی یک سیاست واگذاری مقدار زمان دست‌یابی متغیر به کانال برای پایانه‌های مختلف مشترک امکان‌پذیر می‌شود. بخش نخست زیرقاب UL از سوی مشترکین جهت ارسال پیام‌های کنترل پیوند رادیویی، پیام‌های درخواست پذیرش و اعتبارسنجی در شبکه‌ی WiMAX و نیز پیام پروتکل‌های مدیریت سطح پیشرفته مانند DHCP و SNMP بکار برده می‌شود.

شکل ۶-۱۸ الگوی بسته‌ی WiMAX MAC را نشان می‌دهد. آنها دامنه‌ای که قابل توجه است، دامنه‌ی شناسه‌گر ارتباط در سر پیام است. پروتکل WiMAX یک معماری اتصال‌گرا است که اجازه می‌دهد هر ارتباط از کیفیت خدمات (QoS) عوامل ترافیک و دیگر اطلاعات بهره‌مند باشد. فراهم‌آوری QoS و چگونگی آن باز هم بر عهده‌ی کارگزار شبکه است.

پروتکل WiMAX شیوه‌ی سطح پائین تخمین کانال و دامنه‌های پذیرش درخواست ارتباط جهت حمل اطلاعات بین پایانه‌ی مبنا و میزبان را فراهم می‌کند اما رویکرد کلی و یا سیاست‌گذاری را برای تامین کیفیت خدمات (QoS) فراهم نمی‌آورد. اگرچه هر یک از پایانه‌های مشترکین دارای یک آدرس MAC ویژه به طول 48 بیت مانند شبکه‌های 802.11 و 802.3 می‌باشند اما این آدرس MAC بیشتر در قالب یک شناسه‌گر تجهیزات در WiMAX به شمار می‌آید، زیرا ارتباط بین نقاط انتهایی سرانجام به شناسه‌گر ارتباطی در مقابل آدرس‌های ارسال و دریافت انتهای یک ارتباط نگاشت می‌شود.

1- Opportunistic scheduling
2- Upstream Link Media Access Protocol



شکل ۶-۱۸ ♦ بسته‌ی 802.16

هر چند تحلیل ما از پروتکل WiMAX اندک و خلاصه است، اما موضوعات بسیار دیگری مانند مدیریت قدرت مصرفی، تعویض و جابه‌جایی، زمان‌بندی وابستگی کانال در انتقال بسته‌های MAC از پایانه‌ی مبنا، پشتیبانی QoS و امنیت مواردی هستند که در ارتباط با این شبکه‌ها جایگاه ویژه‌ای را از لحاظ تجزیه و تحلیل دارند. با توجه به این‌که استانداردهای 802.16e در حال حاضر جنبه‌ی پژوهشی دارند سیستم‌های WiMAX قطعاً در سال‌های آتی تکامل و گسترش بیشتری را شاهد خواهند بود. خوانندگان علاقه‌مند به مراجعه به سامانه‌ی رسمی IEEE جهت پیشرفت‌های جدیدتر در این چارچوب دعوت می‌شوند.

۶-۹ دست‌یابی اینترنت به شیوه‌ی سلولی

در بخش قبل چگونگی دست‌یابی یک میزبان به اینترنت از طریق پروتکل WiFi و در نزدیکی یک نقطه‌ی دست‌یابی 802.11 تحلیل شد. در عین حال بر این نکته نیز تاکید شد که اغلب قلمروهای WiFi دارای محدوده پوشش کوچک با قطری مابین 10 تا 100 متر می‌باشند. پروتکل با پوشش وسیع‌تر WiMAX نیز که در حال حاضر در شرایط پژوهش و تجربه است و بایستی تکمیل شود. حال این پرسش مطرح می‌شود که در صورت ضرورت جهت دست‌یابی به اینترنت بی‌سیم، بدون در اختیار داشتن قلمروهای WiFi چه باید کرد؟

با توجه به نفوذ و گستردگی خیره کننده‌ی شبکه‌های تلفنی سیار در بسیاری از نقاط گیتی یک استراتژی طبیعی گسترش شبکه‌های سلولی به گونه‌ای است که نه تنها به پشتیبانی خدمات تلفنی صوتی بپردازد بلکه دست‌یابی بی‌سیم اینترنت را نیز به صورت گسترده فراهم آورد. بدیهی است به شکل ایده‌آل این دست‌یابی اینترنتی در سرعت نسبتاً مطلوب بالا و همراه با تحرک و قدرت جابه‌جایی بی‌نظیر به گونه‌ای می‌باشد که اجازه می‌دهد کاربران تعامل‌های TCP خود را در هر شرایط سیار مانند هنگامی که در ترن یا اتوبوس در حال حرکت هستند، کاملاً پایدار نگهداری نمایند. با سرعت (bps) نسبتاً قابل توجه در دو سوی ارسال و دریافت کاربر نیز حتی قادر به تعامل‌های همایش تصویری می‌شود در حالی که از نقطه‌ای به نقطه‌ی دیگر در حرکت است. بدیهی است این ویژگی اخیر چندان هم دور از دسترس نمی‌باشد. تا هنگام تحریر این اثر در بهار 2012 میلادی بسیاری از گوشی‌های تلفنی

سیار و شرکت‌های مخابراتی سیار در ایالات متحده خدمات دست‌یابی اینترنت سلولی را با هزینه‌ی کمتر از 50 دلار در ماه با سرعت دریافت و ارسال حداقل صدها کیلوبیت در ثانیه در اختیار مشترکین خود قرار داده‌اند. سرعت انتقال داده‌ی چندین مگابیت بر ثانیه به عنوان خدمات پهنای باند گسترده‌ی داده مانند HSPDA در حال حاضر در نقاط مختلف در حال گسترش می‌باشد.

در این بخش فناوری‌های دست‌یابی اینترنت سلولی جدید و جاری بازنگری می‌شوند و تمرکز بر روی نخستین بخش از این دست‌یابی بی‌سیم یعنی گوشی همراه سلولی و زیرساخت شبکه‌ی کابلی قرار می‌گیرد. در بخش‌های بعدی چگونگی انجام ارتباط‌ها و مسیردهی آن‌ها به یک کاربر در حال حرکت مابین پایانه‌های مبنا (BS) تحلیل می‌شود. تمامی مطالب در اطراف سطح پیشرفته‌ی فناوری‌های سلولی خواهد بود و ارتباط‌های سلولی مدرن با عمق و دقت بیشتری بررسی خواهند شد و این مهم به دلیل نفوذ و سیطره‌ی روزافزون آن‌ها در روند روبه‌رشد انتقال‌های سیار می‌باشد.

• بازنگری معماری یک شبکه‌ی سلولی

در رابطه با موضوع معماری شبکه‌های سلولی در این بخش، رویکرد بر کاربرد واژه‌ی **سیستم جهانی برای ارتباطات سیار**[1] **(GSM)** و استانداردهای مربوط به آن پایه‌ریزی می‌شود. در دهه‌ی 1980 میلادی اروپایی‌ها جهت بنیان‌گذاری شبکه‌ای فراگیر در سراسر اروپا که تلفن دیجیتال سلولی را جایگزین شبکه‌های بی‌شمار و ناسازگار تلفنی آنالوگ در آن قاره نماید مجموعه استانداردهای GSM را بنیان نهادند. این فناوری نوین در اوایل دهه‌ی 1990 میلادی با موفقیت خیره‌کننده‌ای در سراسر اروپا گسترش یافت و از آن هنگام به بعد GSM با رشد چشمگیر خود، بیش از 80% مشترکین سلولی را در سراسر جهان که از فناوری GSM بهره می‌گیرند شامل شده است. هنگامی که مردمان در رابطه با فناوری سلولی گفتگو می‌کنند، ایشان اغلب این پدیده را به نسل‌های مختلف آن تقسیم‌بندی می‌کنند. به عنوان نمونه نسل ابتدایی این فناوری تنها برای انتقال ترافیک مکالمه و صدا طراحی گردید و در طبقه‌بندی به آن **سیستم‌های 1G** لقب دادند. بنابراین نسل اول GSM یعنی سیستم‌های 1G صرفاً به گونه‌ی FDM جهت ارتباط‌های صوتی طراحی شده بود. این سیستم‌ها در حال حاضر کاملاً از رده خارج شده‌اند و جایگزین آن‌ها سیستم‌های نسل دوم یعنی 2G شده است. سیستم‌های 2G کاملاً دیجیتال، بر مبنای TDM طراحی شده‌اند و آن‌ها نیز ابتدا به قصد انتقال مکالمه‌های تلفنی به گونه‌ی دیجیتال مورد بهره‌برداری قرار گرفتند، پس از چندی سیستم‌های 2.5G معرفی شدند که انتقال داده را نیز علاوه بر مکالمه پشتیبانی می‌نمایند. به عبارت دیگر سیستم‌های نسل 2.5G دست‌یابی اینترنت و مکالمه‌های تلفنی را پشتیبانی می‌کنند. در حال حاضر سیستم‌های نسل سوم یعنی 3G در حال معرفی و بهره‌برداری قرار دارند. این سیستم‌ها یعنی 3G علاوه بر پشتیبانی تمامی امکانات نسل قبلی خود یعنی 2.5G به صورت فزاینده تاکید بر توان‌مندی‌های انتقال داده بر مبنای دست‌یابی پیوندهای با سرعت‌های بیشتر و بیشتر دارند. در ادامه ابتدا به بررسی سیستم‌های نوین 3G و به دنبال آن 2G و در انتها به ضریب نفوذ این دو سیستم می‌پردازیم. بدیهی است سیستم‌های 4G نیز در انتهای فصل ۶ تحلیل می‌شوند.

۱- مقایسه‌ی سیستم سلولی 3G با شبکه‌های محلی بی‌سیم (WiMAX)

بسیاری از اپراتورهای شبکه‌های تلفنی سیار سلولی در حال حاضر از فناوری 3G با سرعت داده‌ی

1- Global system for mobile communications

2Mbps در فضای سرپوشیده (درونی) و 384 Kbps و بالاتر در خارج (برونی) بهره‌برداری می‌کنند. سیستم‌های 3G در قالب قوانین بین‌المللی تنظیم و اختصاص پهنای باند فرکانس رادیویی به کار گرفته شده‌اند و برخی از اوپراتورها مجبور به پرداخت هزینه‌های قابل ملاحظه به دولت‌ها جهت دریافت مجوز پهنای باند فرکانس رادیویی می‌باشند. سیستم‌های 3G این امکان را برای کاربران فراهم می‌کنند که **دست‌یابی اینترنت** را از نقاط کاملاً دور افتاده‌ی غیر قابل دسترس به گونه‌ی سیار و متحرک به شکلی که در حال حاضر از طریق دست‌یابی به ارتباطات تلفنی سیار انجام می‌گیرد، دنبال نمایند. به عنوان مثال فناوری 3G این اجازه را به یک کاربر می‌دهد که اطلاعات مربوط به نقشه‌ی راه‌ها را در حالی از طریق اینترنت دست‌یابی نماید که در درون وسیله‌ی نقلیه‌ی خویش در حال حرکت می‌باشد. بدیهی است این تنها یک نمونه از هزاران امکان قابل بهره‌برداری می‌باشد. در حال حاضر بسیاری از متخصصان این پرسش را مطرح می‌کنند که آیا فناوری 3G با توجه به هزینه‌ی آن و نیز رقابت شدید فناوری شبکه‌های محلی بی‌سیم WiMAX با موفقیت همراه خواهد بود؟ تمرکز استدلال این پژوهشگران بر پایه‌ی موارد ذیل می‌باشد:

- زیرساخت شبکه‌های محلی بی‌سیم در حال ظهور تقریباً در سراسر جهان نفوذ خود را آشکار نموده است. شبکه‌های محلی بی‌سیم IEEE 802.11 با سرعت 54 Mbps استقبال فزاینده‌ای همراه می‌باشند. تقریباً همه‌ی کامپیوترهای همراه و تبلت‌های با کارت‌های شبکه‌ی محلی 802.11 تجهیز شده‌اند.

علاوه بر آن ابزار و امکانات مصرفی اینترنت در حال ظهور مانند دوربین‌های بی‌سیم و قاب‌های تصویری نیز از کارت‌های شبکه‌ی محلی بی‌سیم و با قدرت مصرف پائین بهره می‌گیرند.

- شبکه‌های WiMAX که در بخش های قبل تحلیل شدند خدمات داده‌ی قلمرو وسیع را برای کاربران سیار در سرعت‌های چندین مگابیت برثانیه و حتی بالاتر نوید می‌دهند. به عنوان مثال شرکت Sprint Nextel سرمایه‌گذاری چند میلیارد دلاری را در زمینه‌ی بهره‌برداری از WiMAX نموده است.
- پایانه‌های برمبنای شبکه‌های محلی بی‌سیم قادر به بهره‌گیری از خدمات تلفنی سیار نیز می‌باشند. گوشی‌های تلفن سیار آتی قادر به ارتباط با هر دو امکان شبکه‌ی تلفن سیار سلولی و یا یک شبکه‌ی IP با بهره‌گیری از یک پیوند VOIP مانند اسکایپ (Skype) خواهند بود. بنابراین از هر دو نوع خدمات مکالمه‌ی سلولی و داده‌ی 3G عبور خواهند کرد.

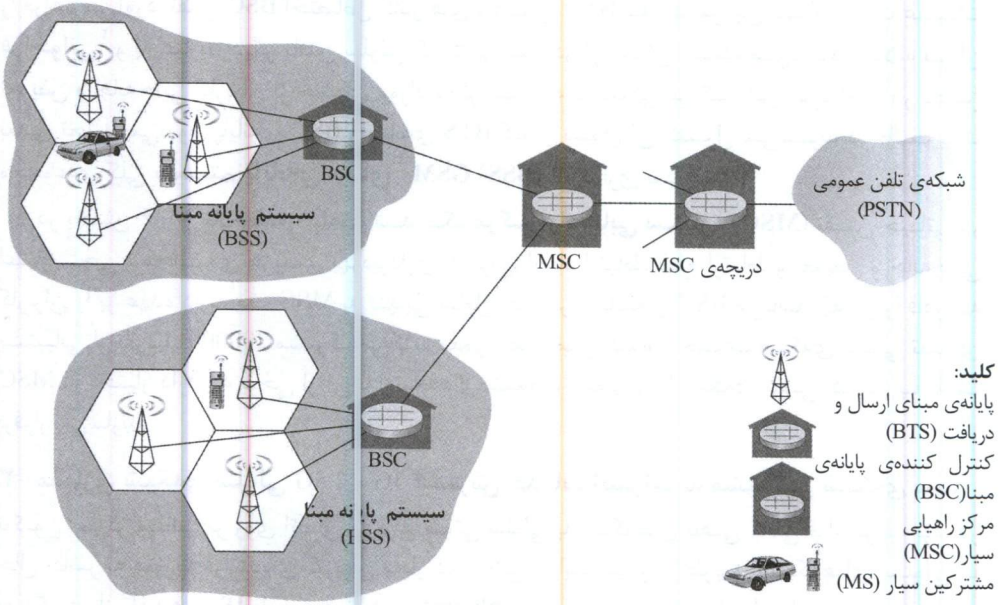
بدیهی است بسیاری از متخصصیان دیگر نیز بر این باور هستند که 3G نه تنها یک فناوری موفق خواهد بود بلکه به گونه‌ی خیره کننده‌ای نحوه‌ی زندگی و کار مردمان را نیز متحول خواهد نمود. یادآوری این نکته نیز در هر حال ضروری است که هر دو فناوری 3G و WiFi سیطره‌ی خود را گسترش خواهند داد، به ویژه از نقطه نظر تجهیزات بی‌سیم رهگیر که به گونه‌ی خودکار به گزینش فناوری دست‌یابی می‌پردازند که با بهره‌گیری از آن بهترین خدمات را در یک نقطه‌ی فیزیکی فراهم آورند.

۲- معماری شبکه‌ی سلولی 2G ارتباط‌های صوتی به شبکه‌ی تلفنی

واژه‌ی **سلولی**[1] اشاره به این حقیقت می‌کند که قلمرو در پوشش از سوی یک شبکه‌ی سلولی به تعدادی ناحیه‌ی پوشش جغرافیایی تقسیم می‌شود و هر ناحیه‌ی جغرافیایی اصطلاحاً **سلول** نامیده

1- Celluar

می‌شود و به صورت نمادین با یک شکل هندسی 6 ضلعی منتظم نشان داده می‌شود. شکل ۶-۱۹ بخشی از یک ساختار سلولی را نشان داده است. همانند استانداردهای 802.11 در شبکه‌های WiFi که از عناصر گوناگون تشکیل شده است فناوری GSM نیز دارای اجزای تشکیلاتی ویژه‌ی خود می‌باشد. هر سلول شامل یک **پایانه‌ی مبنای فرستنده و گیرنده**[1] (BTS) است که وظیفه‌ی آن ارسال و دریافت سیگنال‌ها از پایانه‌ی سیار در سلول خویش است. قلمرو پوشش یک سلول به عوامل زیادی وابسته می‌باشد که از جمله عبارتند از: قدرت انتشار BTS، قدرت انتشار تجهیزات کاربران، موانع ساختمانی در مسیر سلول و ارتفاع مؤثر آنتن پایانه‌ی مبنا . اگرچه در شکل ۶-۱۹ هر سلول شامل یک پایانه‌ی فرستنده و گیرنده مبنا (BTS) در میانه‌ی سلول است اما بسیاری از سیستم‌ها در حال حاضر BTS را در گوشه‌ای از قلمرو پوشش نصب می‌کنند که با سه سلول در تعامل یا ارتباط می‌باشد. با این شیوه یک BTS به تنهایی با بهره‌گیری از آنتن‌های جهت‌دار قادر به تامین خدمات برای تعداد سه سلول می‌شود.



شکل ۶-۱۹ ♦ عناصر معماری شبکه‌ی سلولی **GSM** نسل **2G**

استاندارد GSM برای سیستم‌های سلولی 2G از ترکیب FDM/TDM رادیویی به عنوان واسط فضایی خود بهره می‌گیرند. در فصل اول ملاحظه گردید در FDM خالص (خام) هر کانال FDM بین تعدادی پهنای باند فرکانس تقسیم می‌شد که هر یک از آن‌ها اختصاص به یک مکالمه یا ارتباط داشت. علاوه بر ان در فصل اول ملاحظه شد در TDM خام نیز زمان، مابین قاب‌ها تقسیم‌بندی می‌شد و هر فریم خود به تعدادی شیار یا حفره‌ی زمانی تقسیم می‌شد و هر شیار زمانی اختصاص به یک ارتباط یا مکالمه داشت. بدیهی است در حالت TDM چون هر شیار در قالب یک قاب قرار گرفته بود و فریم نیز در فاصله‌ی زمانی تغییر می‌یافت هر شیار در هر دوره‌ی تغییر می‌توانست به بیش از یک ارتباط اختصاص یابد. در سیستم‌هایی مانند 2G که از شیوه‌ی ترکیب FDM/ TDM بهره می‌گیرند هر کانال

1- Base transmission station

مابین تعدادی پهنای باند فرکانسی تقسیم می‌شود و هر پهنای باند فرکانسی نیز از لحاظ زمانی بین قاب‌ها تقسیم می‌گردد و شیارهای زمانی نیز در درون هر فریم توزیع می‌شوند. بنابراین برای یک سیستم ترکیبی FDM/TDM اگر کانال بین F پهنای باند فرکانسی تقسیم شده باشد و زمان نیز بین T شیار حفره‌ی زمانی آنگاه هر کانال قادر است تعداد F.T مکالمه یا ارتباط همزمان را پشتیبانی نماید.

سیستم‌های GSM شامل باندهای فرکانسی 200 KHz است که هر یک از آن‌ها تعداد 8 مکالمه‌ی TDM را پشتیبانی می‌کند. در سیستم GSM مکالمه بین دو سرعت 13 Kbps و 12.2Kbps رمزگذاری می‌شود. یک پروتکل رقیب GSM نیز IS – 95CDMA نام دارد که در حال حاضر با CDMA2000 جایگزین شده است و از شیوه‌ی دست‌یابی چندگانه‌ی تقسیم کد که در بخش قبل از آن سخن رفت، بهره می‌گیرد. به همین دلیل کاربران IS – 95 قادر به برقراری ارتباط با GSM و برعکس آن نمی‌باشند.

معمولاً **کنترل کننده‌ی پایانه‌ی مبنا**[1] **(BSC)** در یک شبکه‌ی GSM ممکن است از لحاظ فیزیکی هم‌مکان با یک BTS باشد اما به گونه‌ی ویژه یک BSC به تنهایی پشتیبانی خدمات ده‌ها BTS را فراهم می‌آورد. نقش BSC اختصاص کانال‌های رادیویی BTS به مشترکین سیار جهت عملیات **فراخوانی**[2] و به عبارت دیگر یافتن سلولی که کاربر سیار در آن ساکن است، می‌باشد. علاوه بر آن تعویض و جابه‌جایی کاربران از جمله‌ی موارد دیگر است که پشتیبانی می‌کند. این مورد اخیر در بخش بعدی تحلیل می‌شود. پایانه‌ی BSC و پایانه‌ی BTS که از سوی آن کنترل می‌شوند در قالب یک مجموعه‌ی کلی **سیستم پایانه‌ی مبنای**[3] **GSM (BSS)** نام‌گذاری شده است.

در بخش ۶-۱۱ ملاحظه خواهد شد یک **مرکز راه‌یابی سیار**[4] **(MSC)** نقش حیاتی در اعتبارسنجی و محاسبه‌ی هزینه‌ی بهره‌برداری کاربران، ایجاد ارتباط، قطع ارتباط و تعویض و جابه‌جایی کاربران را بر عهده دارد. یک MSC به تنهایی شامل حداکثر 5 پایانه‌ی BSC می‌باشد که آن را قادر به پشتیبانی از تقریباً 200000 مشترک می‌کند. معمولاً هر تأمین‌کننده‌ی خدمات شبکه‌ی سلولی تعدادی MSC در اختیار دارد که برخی از آن‌ها ارتباط‌های شبکه‌ی سلولی را با شبکه‌ی تلفنی کابلی یا ثابت برقرار می‌سازند.

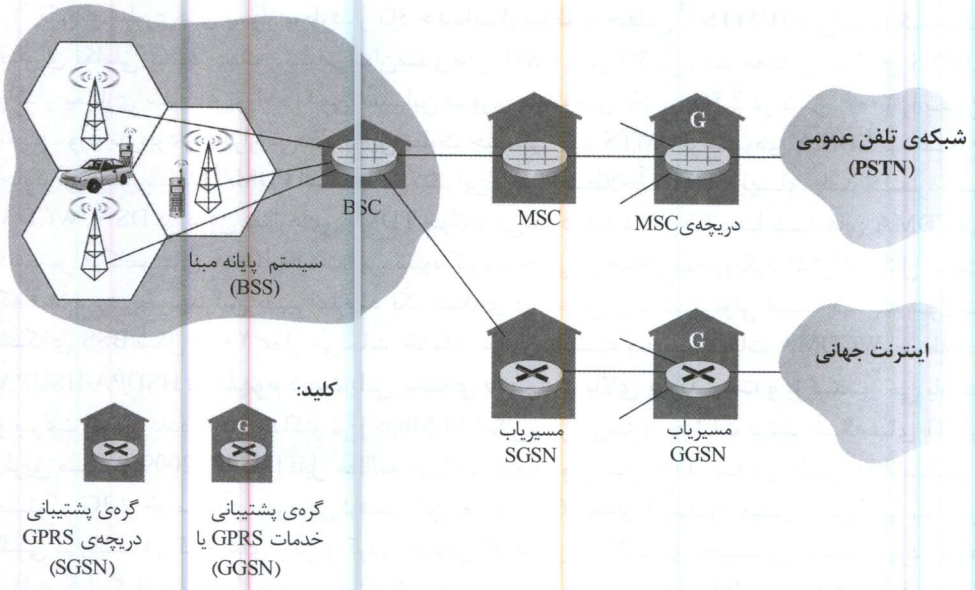
۳- معماری شبکه‌ی سلولی 2.5G و 3G گسترش خدمات اینترنت به مشترکین شبکه‌ی سیار

تاکنون تمرکز مطالب بر روی اتصال کاربران صوتی سلولی به شبکه‌های تلفنی کابلی قرار گرفته بود. در حال حاضر به صورت فزاینده‌ای کاربران سیار دست‌یابی به اینترنت را از طریق شبکه‌های سلولی با بهره‌گیری از تلفن‌های **IP** یا به عبارت دیگر **iphones**، بلاک‌بری، کامپیوترهای لپ‌تاپ و بسیاری دیگر انجام می‌دهند. یک روش جهت رسیدن به این مهم بهره‌گیری از زیرساخت 2G برمبنای شکل ۶-۱۹ می‌باشد که از تلفن سلولی جهت ارتباط شماره‌گیری به یک ISP بهره می‌گیرد همانند کاربران خانگی دهه‌ی 1990 میلادی که از طریق خطوط تلفن ثابت جهت دست‌یابی به یک ISP با بهره‌گیری از یک مودم شماره‌گیر استفاده می‌کردند تا در ارتباط با اینترنت قرار گیرند. نقطه ضعف این شیوه سرعت فوق‌العاده پائین و کند این گونه ارتباط‌ها بود. همین سرعت کند ده‌ها کیلو بیت بر ثانیه نیز در اختیار ارتباط‌های سلولی شماره‌گیری شده قرار خواهد گرفت. در حالی که ایده‌آل دست‌یابی IP به سیستم

1- Base station controller
2- Paging
3- Base station system
4- Mobil switching centers

پایانه‌ی مبنا (BS) با بهره‌گیری از خطوط با پهنای باند گسترده و کانال‌های صوتی چندگانه و از همه مطلوب‌تر دست‌یابی به شبکه‌های رادیویی بهینه می‌باشد که از طریق آن‌ها بتوان کاربران سیار را به BSS در سرعت‌های بالا ارتباط داد. این دقیقاً رویکردی است که سیستم‌های سلولی 2.5G و 3G از آن بهره می‌گیرند.

شکل ۶-۲۰ معماری شبکه‌ی سلولی را برای GSM نسل 2.5G نشان می‌دهد که گسترش یافته‌ی GSM نسل 2G در دست‌یابی به سرعت بالاتر اینترنت می‌باشد. شیوه‌ی بکار گرفته شده از سوی طراحان GSM نسل 2.5G آشکار است: **هسته‌ی شبکه‌ی تلفن سلولی GSM آزاد** نگاه‌داری می‌شود. این مهم از طریق دست‌یابی اینترنت در لبه یا نبش شبکه انجام می‌شود که حالت عملکرد ارزش افزوده‌ی مجزا در برابر عملکرد یکپارچه‌سازی شده در درون هسته‌ی شبکه‌ی تلفن سلولی موجود را دارد. این توامندی افزوده در شبکه‌ی دست‌یابی رادیویی پیاده‌سازی می‌شود. عنصر BSC از طریق معرفی یک شبکه‌ی مجزا از **گره‌های پشتیبانی خدمات GPRS** یا (SGSN) این امکانات را در قالب کلی آن تامین می‌نماید.



شکل ۶-۲۰ ♦ معماری شبکه‌ی داده و صدای سلولی نسل **2.5 G**

در BSC داده‌گرام IP حامل کانال‌های FDM/TDM در واسط فضایی خود از BSC به سوی SGSN گسیل می‌شوند و از طریق ارتباط با MSC عملیات اعتبارسنجی، جابه‌جایی و دیگر عملیات مربوط به کاربران را انجام می‌دهند. علاوه بر همه‌ی این موارد سیگنالینگ ترافیک را نیز بایستی اضافه نمود. داده‌گرام‌های IP از BSC در دو جهت رفت و برگشت از قلمرو بزرگ‌تر از سوی SGSN گسیل می‌شوند که در شبکه‌ی دست‌یابی رادیویی **خدمات رادیویی بسته‌ای عمومی**[2] **(GPRS)** در درون GSM

1- Serving GPRS support node
2- General packet radio service

نسل 2.5G تعریف شده است تا امکان بهره‌برداری پویای کانال‌های رادیویی چندگانه را برای داده‌های IP در اختیار کاربران قرار دهد. نتیجه‌ی حاصل از آن سرعت حداکثر 115 Kbps می‌باشد. در پی گام‌های GPRS **پروتکل سرعت افزوده‌ی داده برای تکامل فراگیر**[1] **(EDGE)** معرفی شده است که سرعت انتقال داده‌ی یک شبکه‌ی GSM/GPRS را به حداکثر 384Kbps ارتقا می‌دهد. علاقه‌مندان جهت دست‌یابی اطلاعات بیشتر در مورد EDGE به مستند [Ericsson 2009] هدایت می‌شوند.

سیستم‌های سلولی 3G جهت پشتیانی خدمات تلفنی و نیز سرعت بالاتر نسل 2.5G یک نیاز ضروری به شمار می‌آیند، به ویژه سیستم‌های 3G بر مبنای اهداف ذیل به عنوان یک ضرورت الزامی پیشنهاد شده است:

- نرخ 144Kbps در سرعت متداول رانندگی
- نرخ 384 Kbps برای کاربران ثابت فضای خارج و کاربران سیار با سرعت حرکت پیاده‌روی معمولی
- نرخ 2 Mbps برای کاربران فضاهای سرپوشیده

یکی از فناوری‌های بسیار پرطرفدار 3G **خدمات ارتباطات جهانی**[2] **(UMTS)** می‌باشد که یک فناوری تکاملی 2.5G جهت پشتبانی توان‌مندی‌های GSM نسل 3G می‌باشد. معماری شبکه‌ی UMTS وامدار معماری شناخته تر GSM می‌باشد. این فناوری شبکه‌های داده‌ی 2.5G در شکل ۶-۲۰ را بدون تغییر مورد بهره‌برداری قرار می‌دهد با این تفاوت چشمگیر که UMTS از شیوه‌ی CDMA به عنوان جایگزین روش FDM/TDMA استفاده می‌کند. این روش اصطلاحاً CDMA پهنای باند گسترده یا DS – WCDA در درون شیارهای TDMA خوانده می‌شود که در آن قاب‌ها شیارهای TDMA در دسترس فرکانس‌های چندگانه است. این شیوه کاربرد جالبی از همه‌ی سه رویکرد اشتراک کانال است که قبلاً معرفی گردید. این تغییر نیازمند یک شبکه‌ی دست‌یابی بی‌سیم سلولی است که به موازات شبکه‌ی BSS شکل ۶-۲۰ عمل می‌نماید. خدمات داده‌ی وابسته به مشخصات WCDMA اصطلاحاً HSDPA/HSUPA به **مفهوم دست‌یابی بسته‌ی با سرعت بالای پیوند رفت و برگشت**[3] می باشد و سرعت انتقال داده را در حداکثر نرخ 14 Mbps امکان‌پذیر می‌سازد. جزئیات بیشتر شبکه‌های 3G از طریق مستند [3GPP 2009] قابل مطالعه می‌باشد. تا ماه ژوئن سال 2009 میلادی تقریباً 200 میلیون مشترک 3G از خدمات آن بهره می‌گرفتند. این تعداد 6.7% همه‌ی 3 میلیارد مشترک سیار در سراسر گیتی می‌باشد. در کشورهای ژاپن و کره‌ی جنوبی که فناوری 3G برای نخستین مرتبه مورد بهره برداری قرار گرفت، بیش از نیمی از مشترکین سیار این کشورها از امکانات 3G بهره گرفتند. در اروپا کشور پیشتاز این فناوری ایتالیا با تقریباً یک‌سوم مشترکین 3G و دیگر کشورهای جلودار این فناوری انگلستان، اتریش، استرالیا و سنگاپور می‌باشند.

با ارایه‌ی چندین نسخه از مشخصات نسل 3G و به کارگیری آن‌ها آیا می‌توان بر این باور بود که نسل بعدی 4G نیز در نزدیکی ظهور قرار گرفته است؟ پاسخ آری و نیز خیر می‌باشد. در حال حاضر تعریف رسمی و مدون از سیستم‌های 4G و معماری آن‌ها وجود ندارد در حالیکه تولیدکنندگانی وجود دارند که تجهیزات نسل 4G یعنی WiMAX را تولید کرده‌اند، این تجهیزات به مراتب از کارایی بیشتری نسبت به 3G برخوردار می‌باشند. قطعاً هنگامی که سیستم‌های 4G تعریف و پیاده‌سازی گردند

1- Enhanced data rates for global evolution
2- Universal mobile telecommunications service
3- High speed downlink/up link packet access

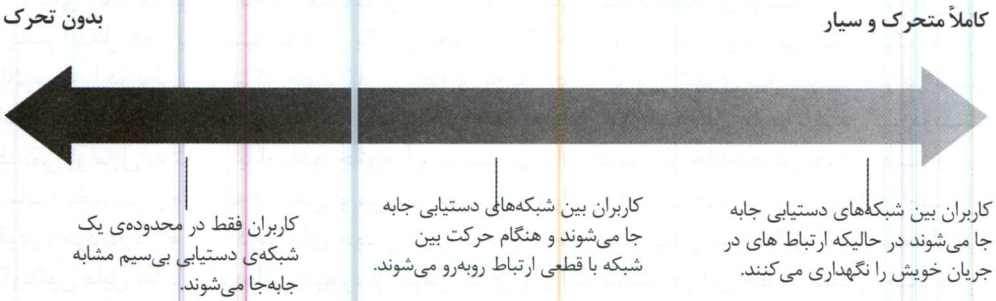
آن‌ها در سرعت به مراتب بالاتری از 3G عمل خواهند نمود شاید 1 Gbps و یا بیشتر، پروتکل‌های اینترنت را کامل‌تر از نسل پیشین خود به کار می‌گیرند و تمرکز آن‌ها بر شیوه و کاربرد چند رسانه‌ای، خدمات بر مبنای موقعیت مکانی و امنیت بیشتر خواهد بود.

۶-۱۰ مدیریت جابه‌جایی

پس از بررسی پیوندهای ارتباطی بی‌سیم در یک شبکه‌ی کامپیوتری بدون کابل اکنون وقت آن رسیده است که موضوع تحرک یا جابه‌جایی در این پیوندها تحلیل شوند. از نقطه‌نظر فراپخش یک گره‌ی سیار به مفهوم آن است که آن گره نقطه‌ی اتصال خود به شبکه را در طول زمان تغییر می‌دهد. از آنجائیکه واژه‌ی تحرک مفاهیم گوناگون را در محیط عملکردی شبکه‌های تلفنی و کامپیوتری به خود اختصاص داده است بنابراین ضروری است از ابعاد مختلف توان‌مندی تحرک یا جابه‌جایی در پیوندها بررسی شوند:

- **از نقطه‌نظر لایه‌ی شبکه تا چه میزان یک کاربر می‌تواند سیار باشد؟** یک کاربر سیار فیزیکی مجموعه‌ای از چالش‌های گوناگون را به لایه‌ی شبکه تحمیل می‌کند. این دشواری‌ها بستگی به چگونگی حرکت و جابه‌جایی کاربرد در بین نقاط شبکه دارد. در یک سوی طیف در شکل ۶-۲۱ یک کاربرد ممکن است یک کامپیوتر همراه حامل یک کارت شبکه‌ی بی‌سیم در اطراف یک ساختمان باشد. در بخش ۶-۷ ملاحظه شد چنین کاربری ممکن است از نقطه‌نظر یک لایه‌ی شبکه سیار نباشد. علاوه بر آن اگر کاربر از وابستگی به نقطه‌ی دست‌یابی مشابهی بدون توجه به موقعیت مکانی برخوردار باشد، آن کاربر حتی از نقطه‌نظر لایه‌ی پیوند نیز سیار نخواهد بود.

در سوی دیگر این طیف حالتی را در نظر بگیرید که کاربری با سرعت 150 کیلومتر در ساعت در یک آزادراه در حال حرکت است و از چندین نقطه‌ی دست‌یابی شبکه‌ی بی‌سیم عبور می‌کند و مایل است ارتباط TCP بدون وقفه‌ای را به یک کاربرد دوردست در طول حرکت و جابه‌جایی خود نگهداری نماید. این کاربر **یقیناً** سیار است. در بین دو وضعیت کاملاً متمایز از یکدیگر کاربری را می‌توان در نظر گرفت که کامپیوتر همراه خود را از یک مکان مانند دفتر کار و یا شبانه‌گاه دانشکده به موقعیت دیگری مانند منزل یا کلاس درس منتقل می‌کندو مایل است به شبکه در مکان جدید مرتبط شود. این کاربر نیز سیار است اگرچه سرعت جابه‌جایی آن کمتر از کاربر درون آزادراه است. علاوه بر این ممکن است این کاربر نیازمند نگهداری یک ارتباط جاری TCP به هنگام عبور بین نقاط ارتباطی شبکه نباشد. شکل ۶-۲۱ طیف میزان و درصد جابه‌جایی کاربران را از نقطه‌نظر لایه‌ی شبکه نشان می‌دهد.



شکل ۶-۲۱ ♦ انواع میزان تحرک از نقطه‌نظر لایه‌ی شبکه

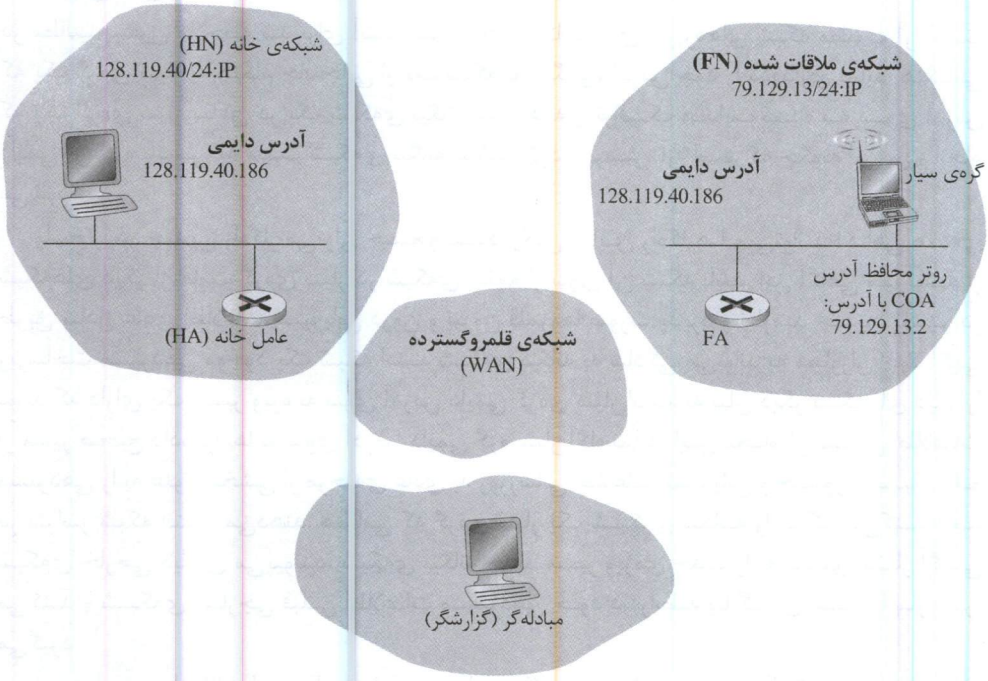
- **اهمیت ثابت ماندن همیشگی آدرس گره‌ی سیار تا چه میزانی است؟** در شبکه‌های تلفنی سیار، شماره تلفن کاربر که همان آدرس لایه‌ی شبکه است همواره به هنگام عبور از یک نقطه به نقطه‌ی دیگر ثابت و غیرقابل تغییر باقی می‌ماند. آیا یک کامپیوتر همراه نیز بایستی از نقطه نظر آدرس IP به هنگام جابه‌جایی در شبکه‌های IP ثابت وبدون تغییر باشد؟

پاسخ این پرسش وابستگی شدید به کاربردهای در حال اجرا دارد. برای کاربر درون خودرو که با سرعت 150 کیلومتر در ساعت در حرکت است و مایل به نگهداری یک ارتباط TCP بدون وقفه می‌باشد، پایدار ماندن آدرس IP مشابه بسیار مطلوب است. یادآور می‌شود در فصل ۳ یک کاربرد اینترنت نیازمند آگاهی از آدرس IP و شماره‌ی پورت عاملی است که با آن در ارتباط می‌باشد. اگر یک موجودیت سیار قادر به نگهداری آدرس IP هنگام عبور و جابه‌جایی در بین نقاط دست‌یابی باشد عامل تحرک از نقطه‌نظر کاربرد پنهان می‌شود بنابراین ارزش قابل توجهی برای این شفاف‌سازی وجود دارد. یک کاربرد نیازمند نگرانی از تغییرات آدرس IP نمی‌باشد و کد کاربری مشابه قادر به تأمین خدمات هر دو ارتباط سیار و ثابت می‌باشد. در ادامه‌ی مطالب همین فصل ملاحظه خواهد شد که IP سیار این شفافیت را حفظ نموده و این اجازه را به یک گره‌ی سیار می‌دهد که آدرس IP خود را در عین حال جابه‌جایی بین شبکه‌ها ثابت نگاه دارد. از طرف دیگر یک کاربر سیار دیگر ممکن است تنها نیاز به خاموش کردن کامپیوتر همراه خود از دفتر کار و حمل آن به منزل و سپس روشن نمودن آن و ادامه‌ی کار در محل مسکونی خود داشته باشد. اگر کامپیوتر همراه در این حالت فقط به عنوان یک مشتری در یک کابرد مشتری – سرور از محل منزل خویش باشد، آدرس IP ویژه‌ی استفاده شده از چندان اهمیتی برخوردار نمی‌باشد و در اختیار گرفتن آدرس IP موقت از جانب DHCP کار را به سرانجام می‌رساند.

- **چه نوع پشتیبانی زیرساخت کابلی وجود دارد؟** در همگی موارد بالا انحصاراً فرض بر آن بود که از یک زیرساخت ثابت جهت ارتباط کاربر سیار استفاده می‌شود. از جمله‌ی این زیرساخت‌ها می‌توان از شبکه‌ی ISP خانگی، شبکه‌ی دست‌یابی بی‌سیم محل کار و یا شبکه‌ی دست‌یابی بی‌سیم فرودگاه و غیره نام برد. اگر چنین زیرساختی وجود نداشته باشد چه اتفاقی روی می‌دهد؟ اگر دو کاربر در درون و یا در نزدیکی ارتباط با یکدیگر باشند آیا می‌توانند در غیاب هر زیرساخت لایه‌ی شبکه با یکدیگر ارتباط برقرار نمایند؟ پاسخ آن است که شبکه‌های ad – hoc چنین توان‌مندی‌هایی را فراهم می‌آورند. این بخش از شبکه‌سازی سیار با رشد سریع خود یکی از مهم‌ترین زمینه‌ها به شمار می‌آید.

برای آنکه موضوعات تأمین امکانات برای یک کاربر سیار جهت نگاهداری ارتباط TCP در جریان بیشتر آشکار شود یک شبیه‌سازی از یک فرد جوان اصطلاحاً خانه به دوش آورده می‌شود. یک جوان 20 ساله را در نظر بگیرید که جهت کار از نقطه‌ای به نقطه‌ی دیگر و از شهری به شهر دیگر در سفر است و در اغلب اوقات مجبور به تغییر آدرس‌های خود به دلیل این جابه‌جایی‌ها می‌باشد. اگر دوست قدیمی او مایل به تماس با او باشد چگونه آن دوست می‌تواند آدرس یار خانه‌به‌دوش خود را دست‌یابی نماید؟ نخستین راه‌کار که به ذهن او می‌رسد تماس با خانواده‌ی دوست خود می‌باشد زیرا این احتمال قوی وجود دارد که وی آدرس‌های خود را مرتباً با خانواده‌اش ثبت کند، زیرا ممکن است به دلایل گوناگون مایل باشد خانواده‌اش همواره از آدرس جاری وی آگاه باشند. در این حالت مسکن خانوادگی این جوان خانه‌به‌دوش که دارای یک آدرس دایمی است مکانی است که از طریق آن دیگران می‌توانند

به عنوان نخستین گام در جهت ارتباط با وی که سیار می‌باشد، تلاش خود را انجام دهند. گام ارتباطی بعد ممکن است از طریق نامه‌ی پستی ارسالی به خانواده‌ی وی و سپس از سوی آن‌ها به دوست مورد نظر باشد و یا آن که مستقیماً از طریق آدرس بدست آمده از خانواده‌ی وی ارسال نامه انجام گیرد.



شکل ۶-۲۲ ♦ عناصر ابتدایی معماری یک شبکه‌ی سیار

در یک قلمرو شبکه آدرس دایمی یک گره‌ی سیار مانند کامپیوتر و یا گوشی همراه و یا یک تبلت اصطلاحاً **شبکه‌ی خانه**[1] **(HN)** نام دارد و موجودیت درون شبکه خانه که عملیات مدیریت و هماهنگی تحرک را از سوی گره‌ی سیار انجام می‌دهد **عامل خانه**[2] **(HA)** نامیده شده است. شبکه‌ای که در حال حاضر گره‌ی سیار در آن ساکن است **شبکه‌ی بیگانه**[3] **(ملاقات شده)** و موجودیت درون این شبکه که تسهیلات لازم را در اختیار گره‌ی سیار برای عملیات مدیریت تحرک وی قرار می‌دهد **عامل بیگانه**[4] **(FA)** نامگذاری کرده‌اند. برای کاربران سیار پیشرفته شبکه‌ی خانه احتمالاً شبکه‌ی شرکت و محل کار آن‌ها می‌باشد در حالیکه شبکه‌ی بیگانه ممکن است شبکه‌ی یک هم‌کار باشد که آن‌ها بازدید نموده‌اند. معمولاً یک **مبادله‌گر**[5] **(گزارشگر)** موجودیتی است که مایل به ارتباط با گره‌ی سیار می‌باشد. شکل ۶-۲۲ علاوه بر این اصول، مبانی آدرس‌دهی را نیز در این رابطه نشان می‌دهد. در

1- Home Network
2- Home Agent
3- Foreign Network
4- Foreign Agent
5- Correspondent

شکل ۶-۲۲ عوامل در قالب مجموعه‌ای شرکت‌پذیر با مسیریاب‌ها به عنوان فرآیندهای در حال اجرا بر روی روترها نشان داده شده‌اند، اما آن‌ها می‌توانند بر روی میزبان‌ها و سرورهای دیگر شبکه نیز در حال اجرا باشند.

- **آدرس‌دهی**

در مطالب پیشین ملاحظه شد برای آشکار شدن تحرک کاربر برای کاربردهای شبکه مطلوب آن است که یک گره‌ی سیار به هنگام جابه‌جایی از یک شبکه به دیگری، آدرس خود را نگهداری نماید. هنگامی که یک گره‌ی سیار ساکن در یک شبکه‌ی بیگانه است همه‌ی ترافیک هدایت شده به سوی آدرس دایمی آن گره بایستی به سمت شبکه‌ی بیگانه هدایت گردد. پرسش آن است که چگونه این مهم انجام می‌شود؟

پاسخ را در چندین راه‌کار می‌توان جستجو نمود. یکی از این راه‌کارها می‌تواند آگاهی همه‌ی شبکه‌های دیگر از اقامت گره‌ی سیار در شبکه‌ی بیگانه از سوی آن شبکه باشد. این آگاهی می‌تواند از طریق تبادل عادی اطلاعات مسیریابی درون و بیرون قلمروها صورت پذیرد و نیازمند چندین تغییر در زیرساخت مسیردهی موجود یک شبکه است. شبکه‌ی بیگانه به سادگی می‌تواند به مجاوران خود آگهی نماید که دارای یک مسیر ویژه به سوی آدرس دایمی گره‌ی سیار است. به بیان دیگر شبکه‌های دیگر را از مسیر صحیح داده‌گرام‌ها به سوی آدرس دایمی گره سیار آگاه نماید. این مجاوران سپس اطلاعات مسیردهی را به عنوان بخشی از مرحله‌ی عادی به روزرسانی اطلاعات مسیریابی و جداول گسیل جلو در سراسر شبکه انتشار می‌دهند. هنگامی که گره‌ی سیار یک شبکه‌ی بیگانه را ترک می‌کند و به شبکه‌ی خارجی دیگری می‌پیوندد، شبکه‌ی بیگانه جدید مسیر ویژه‌ی جدید را به گره‌ی سیار آگهی می‌کند و شبکه‌ی خارجی قبلی اطلاعات مسیردهی خود در رابطه با گره‌ی سیار را باز پس می‌گیرد.

این فرآیند دو اشکال را همزمان برطرف می‌سازد: (۱) این عمل بدون اعمال تغییرات عمده در زیرساخت لایه‌ی شبکه صورت می‌پذیرد، (۲) شبکه‌های دیگر از موقعیت گره‌ی سیار آگاه می‌باشند و به راحتی نیز مسیردهی داده‌گرام‌ها به گره‌ی سیار صورت می‌پذیرد، زیرا جداول گسیل جلو داده‌گرام‌ها را به سوی شبکه‌ی خارجی هدایت می‌کنند. یک نقطه ضعف این رویکرد در مقیاس‌پذیری شبکه می‌باشد. اگر مدیریت تحرک عهده‌دار مسئولیت مسیریاب‌های شبکه باشد، آنگاه روترها بایستی اطلاعات مربوط به مسیردهی میلیون‌ها گره‌ی سیار را در جداول گسیل جلوی خود نگهداری و هنگامی که گره‌ها جابه‌جا می‌شوند آن‌ها را به روزرسانی نمایند.

رویکرد جایگزین که در عمل نیز از آن استفاده می‌شود انتقال عملکرد تحرک از هسته‌ی شبکه به نبش شبکه می‌باشد. موضوعی که در مطالعه‌ی معماری اینترنت به آن اشاره رفت. رویکرد طبیعی این پیاده‌سازی از طریق یک شبکه‌ی خانه در گره‌ی سیار است. مشابه همان روشی که والدین جوان خانه به دوش جهت رهگیری موقعیت فرزند خود انجام می‌دهند. به عبارت دیگر عامل خانه در شبکه‌ی خانه‌ی گره‌ی سیار اقدام به ردیابی شبکه‌ی بیگانه که گره‌ی سیار در آن ساکن شده است می‌کند. یک پروتکل بین گره‌ی سیار یا یک عامل بیگانه که آن گره را نمایندگی می‌کند و عامل خانه جهت به روزرسانی موقعیت مکانی گره‌ی سیار لازم و ضروری است .

اکنون به جزئیات عامل بیگانه می‌پردازیم. بر مبنای شکل ۶-۲۲ منظور اصلی ردگیری عوامل خارجی در روترهای نبش شبکه‌ی بیگانه می‌باشد. یک نقش اصولی عامل بیگانه تعریف یک مراقب

آدرس[1] (COA) برای گره‌های سیار می‌باشد که در آن بخش شبکه‌ی COA نگاشتی را از شبکه‌ی خارجی ارایه می‌دهد. بنابراین دو آدرس وابسته به یک گره‌ی سیار وجود دارد. یکی از این دو، **آدرس دایمی[2]** است و دیگری آدرس COA می‌باشد که اغلب اوقات به آن **آدرس بیگانه[3]** نیز گفته می‌شود. در مثال ۶-۲۲ آدرس دایمی گره‌ی سیار 128.119.40.186 است. هنگامی که گره‌ی سیار، شبکه‌ی با آدرس 79.129.13/24 را ملاقات می‌کند، گره‌ی سیار دارای یک آدرس COA به صورت 79.129.13.2 می‌شود. نقش دوم عامل بیگانه با یک آدرس اختصاص یافته‌ی COA می‌باشد. در ادامه ملاحظه خواهد شد که از COA جهت مسیردهی مجدد داده‌گرام‌ها به گره‌ی سیار از طریق عامل بیگانه استفاده می‌شود.

اگرچه از لحاظ عملیاتی گره‌ی سیار و عامل بیگانه از یکدیگر مجزا شده‌اند اما آگاهی از این نکته ضروری است که گره‌ی سیار می‌تواند عهده‌دار مسئولیت عامل بیگانه نیز باشد. به عنوان مثال گره‌ی سیار می‌تواند یک آدرس COA را از شبکه‌ی خارجی با بهره‌گیری از پروتکل DHCP اخذ نماید و شخصاً عامل خانه را از COA خویش آگاه نماید.

۶-۱۱ مسیردهی به یک گره‌ی سیار

ملاحظه نمودیم چگونه یک گره‌ی سیار آدرس COA را شناسایی و اخذ می‌کند و نیز چگونه عامل خانه از آن آدرس آگاه می‌شود. اما این دانش از آدرس COA از سوی عامل خانه تنها بخشی از مشکل به شمار می‌آید. سوی دیگر دشواری چگونگی انجام آدرس‌دهی و هدایت داده‌گرام‌ها به گره‌ی سیار است؟ از آنجائیکه عامل خانه فقط از موقعیت گره‌های سیار آگاهی دارد بنابراین آدرس‌دهی داده‌گرام‌ها به سوی آدرس دایمی گره‌ی سیار و ارسال آن به درون زیرساخت لایه‌ی شبکه فقط کافی نیست و فعالیت‌های بیشتری لازم می‌باشد. در ارتباط با این موضوع دو رویکرد را می‌توان شناسایی نمود. این دو عبارتند از مسیردهی مستقیم و غیر مستقیم.

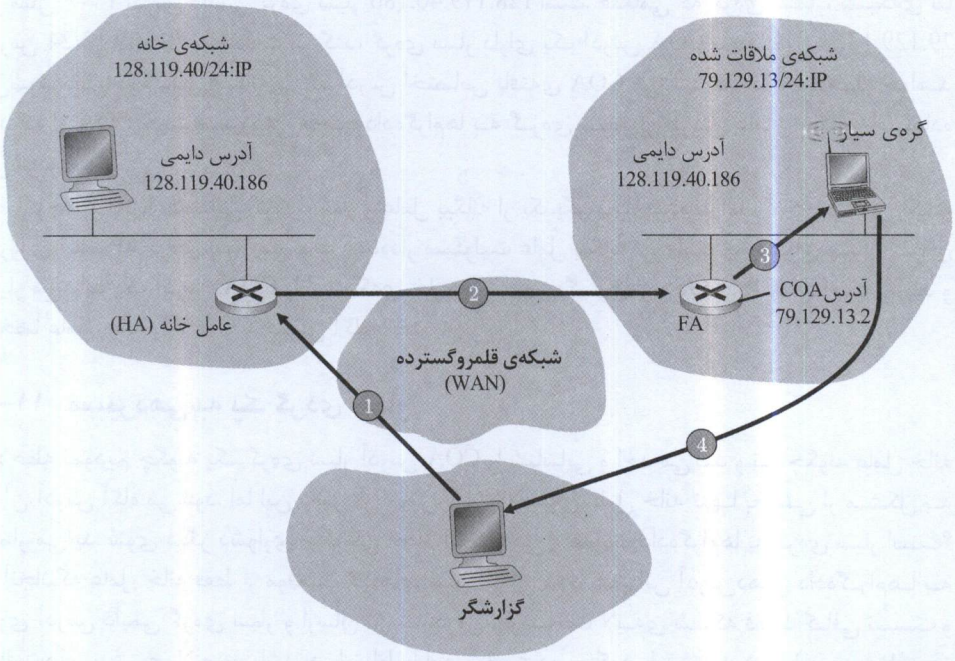
• مسیردهی غیر مستقیم به یک گره‌ی سیار

ابتدا یک مبادله‌گر (گزارشگر) را در نظر آورید که مایل به گسیل یک داده‌گرام به یک گره‌ی سیار است. در شیوه‌ی مسیردهی غیر مستقیم، گزارشگر داده‌گرام را به آدرس دایمی گره‌ی سیار هدایت و داده‌گرام را وارد شبکه می‌کند. در این لحظه مبادله‌گر نگران اقامت گره‌ی سیار در شبکه‌ی خانه و یا ان که وی یک شبکه‌ی خارجی را ملاقات می‌کند یا نه، نمی‌باشد. این موضوع به مفهوم آن است که عامل تحرک کاملاً برای گزارشگر آشکار است. مسیردهی چنین داده‌گرام‌هایی نخست به سوی شبکه‌ی خانه‌ی گره‌ی سیار می‌باشد که در گام اول شکل ۶-۲۳ نشان داده شده است.

اکنون نگاهی به عامل خانه می‌اندازیم. این عامل علاوه بر مسئولیت برقراری ارتباط با یک عامل بیگانه به منظور رهگیری COA متعلق به گره‌ی سیار، عملکرد مهم دیگری را نیز بر دوش خود حمل می‌کند. وظیفه‌ی دوم عامل خانه رصد داده‌گرام‌های هدایت شده به سوی گره‌هایی است که شبکه‌ی خانه‌ی آن‌ها همان شبکه‌ی عامل خانه است اما در حال حاضر در یک شبکه ی بیگانه مقیم می‌باشند. عامل خانه این داده گرامها را شناسایی و سپس آنها را به سوی یک گره ی سیار در یک فرآیند دو

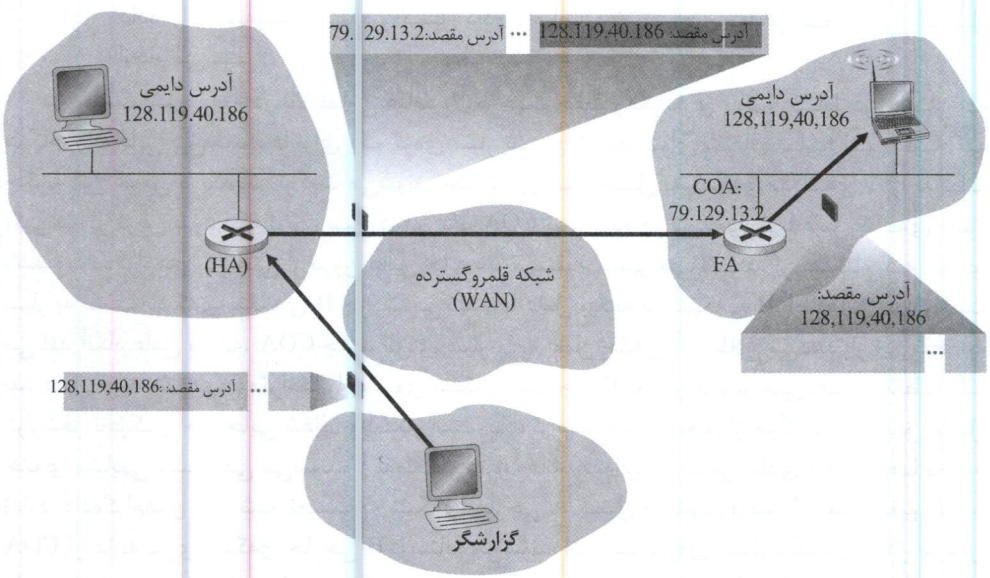
1- Care of address
2- Permanent address
3- Foreign address

مرحله‌ای گسیل می‌کند. نخست داده‌گرام به عامل بیگانه ارسال می‌شود. این مهم از طریق بهره‌گیری از آدرس COA گره‌ی سیار در گام ۲ در شکل ۶-۲۳ صورت می‌گیرد و سپس گسیل به سوی گره‌ی سیار از طریق عامل بیگانه انجام می‌شود که مرحله‌ی ۳ در شکل ۶-۲۳ می‌باشد.



شکل ۶-۲۳ ♦ مسیردهی غیرمستقیم به یک گره‌ی سیار

برای عامل خانه ضروری است که هدایت داده‌گرام را با بهره‌گیری از COA گره‌ی سیار انجام دهد تا آن که لایه‌ی شبکه قادر به مسیردهی داده‌گرام به شبکه‌ی بیگانه باشد. از طرف دیگر بهینه است که یکپارچگی داده‌گرام متعلق به گزارشگر بدون هیچ‌گونه تغییری حفظ شود. زیرا کاربرد دریافت کننده‌ی داده‌گرام بی‌خبر از گسیل داده‌گرام از سوی عامل خانه می‌باشد. دست‌یابی به این دو هدف از طریق وادار نمودن عامل خانه نسبت به بسته‌بندی کامل داده‌گرام متعلق به مبادله‌گر در یک داده‌گرام جدید و بزرگ‌تر امکان‌پذیر می‌شود. این داده‌گرام بزرگ‌تر به سوی COA گره‌ی سیار هدایت و تحویل آن می‌شود. عامل بیگانه‌ی مالک COA داده‌گرام را دریافت و بسته‌بندی آن را بازگشایی می‌کند. به بیان دیگر داده‌گرام اصلی گزارشگر را از بسته‌ی جدید جدا می‌شود و بر اساس گام ۳ در شکل ۶-۲۳ به سوی گره‌ی سیار هدایت می‌گردد. شکل ۶-۲۴ نشان داده است چگونه داده‌گرام اصلی مبادله‌گر به سمت شبکه‌ی خانه هدایت می‌شود. یک داده‌گرام بسته‌بندی شده به سوی عامل بیگانه ارسال و داده‌گرام اصلی تحویل گره‌ی سیار می‌شود. برای خوانندگان نکته سنج آشکار است که عملیات بسته بندی و بازگشایی مجدد بسته بسیار شبیه به مفاهیم تونل‌زنی در فصل ۴ در چارچوب دو پروتکل IP و چندپخش IPV6 می‌باشد.



شکل ۶-۲۴ ♦ بسته‌بندی داده‌گرام اصلی مبادله‌گر و بازگشایی آن

مرحله‌ی بعدی در نظر گرفتن چگونگی ارسال داده‌گرام از سوی گره‌ی سیار به گزارشگر می‌باشد. در حقیقت این عملیات به سادگی انجام می‌گیرد، گره‌ی سیار داده‌گرام خویش را **مستقیماً** به گزارشگر از طریق بهره گیری از آدرس دایمی خود به عنوان آدرس مبدا و آدرس گزارشگر به عنوان آدرس مقصد ارسال می‌دارد. از آنجائیکه گره‌ی سیار از آدرس گزارشگر آگاهی دارد بنابراین دیگر نیازی به مسیردهی داده‌گرام برای بازگشت به عامل خانه نمی‌باشد. این عملکرد در گام ۴ شکل ۶-۲۳ نشان داده شده است.

بحث مربوط به مسیردهی غیرمستقیم را با فهرستی جدید از عملیات لایه‌ی شبکه جهت پشتیبانی عامل تحرک به پایان می‌بریم:

- **پروتکل یک گره‌ی سیار به عامل بیگانه.** گره‌ی سیار هنگام اتصال به شبکه‌ی بیگانه وضعیت خود را با عامل بیگانه ثبت می‌کند. به صورت مشابه یک گره‌ی سیار هنگام ترک خود از شبکه‌ی بیگانه اطلاعات ثبت شده را باز پس می‌گیرد.
- **پروتکل ثبت یک عامل بیگانه به عامل خانه.** عامل بیگانه نسبت به ثبت اطلاعات COA گره‌ی سیار با عامل خانه اقدام می‌کند. هنگامی که گره‌ی سیار شبکه‌ی خود را ترک می‌کند عامل بیگانه نیازی به باز پس‌گیری اطلاعات COA آن گره‌ی سیار ندارد زیرا عملیات ثبت‌نام مجدد یک COA جدید هنگامی که گره‌ی سیار به یک شبکه‌ی جدید وارد می‌شود وجود این نیاز را برطرف می‌سازد.
- **پروتکل بسته بندی داده‌گرام یک عامل خانه.** بسته بندی و ارسال داده‌گرام اصلی گزارشگر در درون یک داده‌گرام آدرس‌دهی شده به COA انجام می‌گیرد.

- **پروتکل بازگشایی بسته‌ی یک عامل بیگانه**. بازیابی داده‌گرام اصلی گزارشگر از داده‌گرام بسته‌بندی کننده و گسیل داده‌گرام اصلی به گره‌ی سیار از سوی این پروتکل انجام می‌شود.

در مطالب اشاره رفته در بالا تمامی عناصر لازم جهت حفظ و نگهداری یک ارتباط در جریان به هنگام جابه‌جایی بین شبکه‌ها برای یک گره‌ی سیار تشریح گردید. برای شفاف‌سازی این نکته که چگونه این عوامل با یکدیگر ترکیب می‌شوند، یک گره‌ی سیار متصل به شبکه‌ی بیگانه‌ی A را در نظر بگیرید. اگر فرض شود این گره‌ی سیار دارای یک COA ثبت شده در شبکه‌ی A با عامل خانه‌ی خود باشد و داده‌گرام‌های دریافتی از درون عامل خانه غیر مستقیم مسیردهی شوند، آن هنگام که گره‌ی سیار به سمت شبکه‌ی بیگانه‌ی B حرکت می‌کند و با عامل بیگانه در شبکه‌ی B، اطلاعات خود را ثبت می‌کند آنگاه عامل بیگانه، COA جدید گره‌ی سیار را به عامل خانه‌ی آن اطلاع می‌دهد. از این نقطه به بعد عامل خانه همه‌ی داده‌گرام‌ها را به سوی شبکه‌ی خارجی B تغییر مسیر می‌دهد. از نقطه‌نظر گزارشگر تحرک و جابه‌جایی شفاف و آشکار است. داده‌گرام‌ها قبل و بعد از حرکت از طریق عامل خانه‌ی مشابهی مسیردهی می‌شوند. از نقطه‌نظر عامل خانه وقفه‌ای در جریان عادی داده‌گرام‌ها وجود ندارد. داده‌گرام‌های وارد شده نخست به شبکه‌ی خارجی A گسیل می‌شوند و پس از اعمال تغییرات در COA آن‌ها به سوی شبکه‌ی خارجی B فرستاده می‌شوند. اما آیا گره‌ی سیار وقفه‌ای را در جریان عادی داده‌گرام‌ها به هنگام جابه‌جایی بین شبکه‌ها مشاهده می‌کند؟ تا زمانی که فاصله‌ی زمانی بین قطع گره‌ی سیار از شبکه‌ی A که با وقوع آن دیگر قادر به دریافت داده‌گرام از A نمی‌باشد و پیوستن آن به شبکه‌ی B که هنگام وقوع آن COA جدید را با عامل خانه‌ی خود ثبت می‌کند، بسیار اندک باشد، تعداد بسیار کمی از داده‌گرام‌ها تلف می‌شوند. از نتیجه‌ی مطالب فصل ۳ یادآور می‌شود که هر ارتباط انتها به انتهایی می‌تواند دچار فقدان داده‌گرام‌ها در اثر تراکم شبکه گردد. بنابراین فقدان اتفاقی داده‌گرام‌ها در بین ارتباط هنگامی که یک گره بین شبکه‌ها جابه‌جا می‌شود امری غیر عادی به شمار نمی‌آید. در صورت نیاز به ارتباط فارغ از فقدان داده‌گرام‌ها شیوه‌های اعمال شده در لایه‌های بالاتر عملیات بازیابی داده‌گرام‌ها را انجام می‌دهند فارغ از این که آیا اتلاف داده‌گرام‌ها در اثر جابه جایی کاربر بوده است و یا تراکم شبکه. شیوه‌ی مسیردهی غیر مستقیم در IP سیار بر مبنای مستند [RFC3344] می‌باشد که در بخش ۶-۱۲ مورد مطالعه قرار می‌گیرد.

- **مسیردهی مستقیم به یک گره‌ی سیار**

رویکرد مسیریابی غیرمستقیم در شکل ۶-۲۳ از نقطه نظر کارآمدی دارای کمبودی است که اصطلاحاً به آن **چالش مثلث مسیریابی**[1] گفته می‌شود. داده‌گرام‌های هدایت شده به سوی گره‌ی سیار ابتدا لازم است به سوی عامل خانه و سپس شبکه‌ی بیگانه گسیل شوند. این وضعیت حتی هنگامی که مسیرهای به مراتب کار آمدتر بین گزارشگر و گره‌ی سیار وجود دارند، اجتناب ناپذیر است. در حادترین وضعیت می‌توان یک کار بر سیار (A) را که در حال عبور از شبکه‌ی خارجی یک همکار (B) خود می باشد تصور نمود. هر دو کار بر در کنار یکدیگر مشغول مبادله‌ی داده‌ها از طریق شبکه می‌باشند. داده‌گرام‌های ارسالی از سوی عامل گزارشگر که در این حالت B می‌باشد به سوی عامل خانه‌ی کار بر A هدایت شده و پس از آن مجدداً به شکل بیگانه بازگشت داده می‌شود!

1- Triangle routing problem

مسیردهی مستقیم با افزایش پیچیدگی، ضعف حاصل از چالش مثلث مسیریابی بر طرف می‌شود. در شیوه‌ی مسیر یابی مستقیم یک **عامل گزارشگر** در شبکه‌ی متعلق به خود ابتدا COA عامل سیار را یادگیری می‌کند. این مهم از طریق در خواست از عامل خانه با این فرض که گره‌ی سیار ارزش COA ثبت شده‌ی خود با عامل خانه را به روز رسانی نموده است، انجام می‌شود. بدیهی است این عملیات در شیوه ی مسیردهی غیر مستقیم نیز انجام می‌شد. علاوه بر این ممکن است مبادله‌گر خود وظیفه‌ی عامل گزارشگر را مانند گره‌ی سیار که می‌توانست وظیفه‌ی عملیات عامل بیگانه را انجام دهد، بر عهده گیرد. این موارد در شکل ۶-۲۵ از طریق گام‌های 1و 2 انجام می‌شود. عامل گزارشگر سپس داده گرام‌ها را مستقیماً بهCOA گره‌ی سیار تونل می‌زند. روش تونل زنی به COA در این حالت مانند رفتار عامل خانه در مراحل 3و4 در شکل ۶-۲۵ می‌باشد.

اگر چه مسیردهی مستقیم چالش مثلث مسیریابی را بر طرف می‌سازد اما دو اشکال عمده ی دیگر که در ذیل توضیح داده شده است را نیز سبب می‌شود:

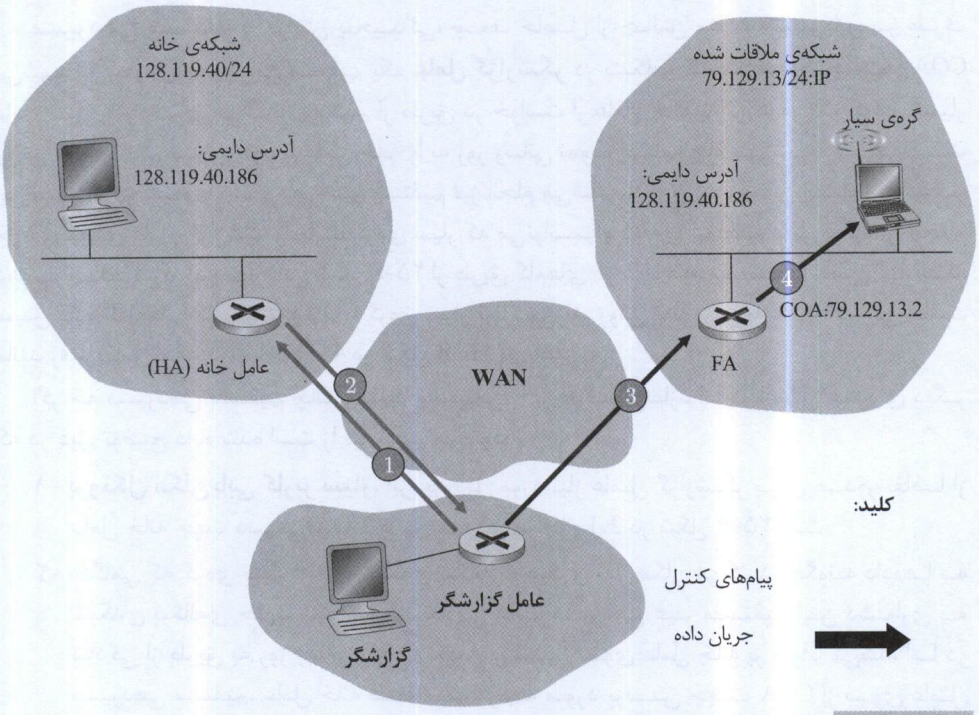
۱- **پروتکل مکان‌یابی کاربر سیار.** این پروتکل مورد نیاز عامل گزارشگر برای صدور تقاضا از عامل خانه جهت دسیابی COA گره‌ی سیار در مراحل 1و2 در شکل ۶-۲۵ است.

۲- هنگامی که گره‌ی سیار از یک شبکه‌ی بیگانه به دیگری نقل مکان می‌کند، چگونه داده‌ها به شبکه‌ی بیگانه‌ی جدید منتقل می‌شوند؟ در حالت مسیردهی غیر مستقیم این دشواری به سادگی از طریق به روز رسانی COA نگهداری شده از سوی عامل خانه بر طرف می‌شد، اما در مسیردهی مستقیم، عامل خانه فقط یک مرتبه مورد پرسش جهت COA از سوی عامل گزارشگر آن هم در آغاز تعامل قرار می‌گیرد. بنابراین به روز رسانی COA در عامل خانه اگر چه ضروری است اما کافی جهت بر طرف نمودن چالش مسیر یابی داده به شبکه‌ی بیگانه‌ی جدید گره‌ی سیار نمی‌باشد.

یک راهکار می‌تواند خلق یک پروتکل جدید جهت آگاه‌سازی گزارشگر از تغییرات COA باشد. یک شیوه‌ی جایگزین و آنچه در حقیقت در عمل در شبکه‌های GSM مورد بهره‌برداری قرار می‌گیرد به گونه‌ای است که توضیحات آن به دنبال می‌آید. فرض کنید جریان داده‌ها در حال حاضر از شبکه‌ی بیگانه به سوی گره‌ی سیار هنگامی که آن گره در آغاز تعامل در آن مکان مقیم بوده است (یعنی مرحله‌ی ۱ در شکل ۶-۲۵)، گسیل می‌شود. عامل خارجی در شبکه‌ی بیگانه که گره‌ی سیار نخستین مرتبه دستیابی نموده است در واژگان شبکه به **عامل خارجی وابسته**[1] معروف می‌باشد.

هنگامی که گره‌ی سیار به یک شبکه‌ی بیگانه‌ی جدید (مانند مرحله‌ی 2 درشکل ۶-۲۶) وارد می‌شود، گره‌ی سیار با عامل بیگانه (گام3) ثبت نام می‌کند، کارگزار بیگانه ی جدید عامل خارجی وابسته را از طریق COA جدید گره‌ی سیار (گام4) تعریف می‌کند. هنگامی که کارگزار خارجی وابسته، یک داده‌گرام بسته‌بندی شده را از یک گره‌ی سیار خارج شده، دریافت می‌دارد، پس از آن می‌تواند داده‌گرام را مجدداً بسته‌بندی و آن را با بهره‌گیری از COA جدید در گام 5 به سوی گره‌ی سیار ارسال نماید. اگر گره سیار مجدداً جابه‌جا شود و به یک شبکه‌ی بیگانه‌ی جدید وارد گردد، عامل بیگانه در آن شبکه‌ی تازه ملاقات شده در تماس با کارگزار بیگانه‌ی وابسته قرار می‌گیرد تا عملیات گسیل داده به این شبکه خارجی جدید را هماهنگ نماید.

1- Anchor foreign agent



شکل ۶-۲۵ ♦ مسیردهی مستقیم به یک کاربر سیار

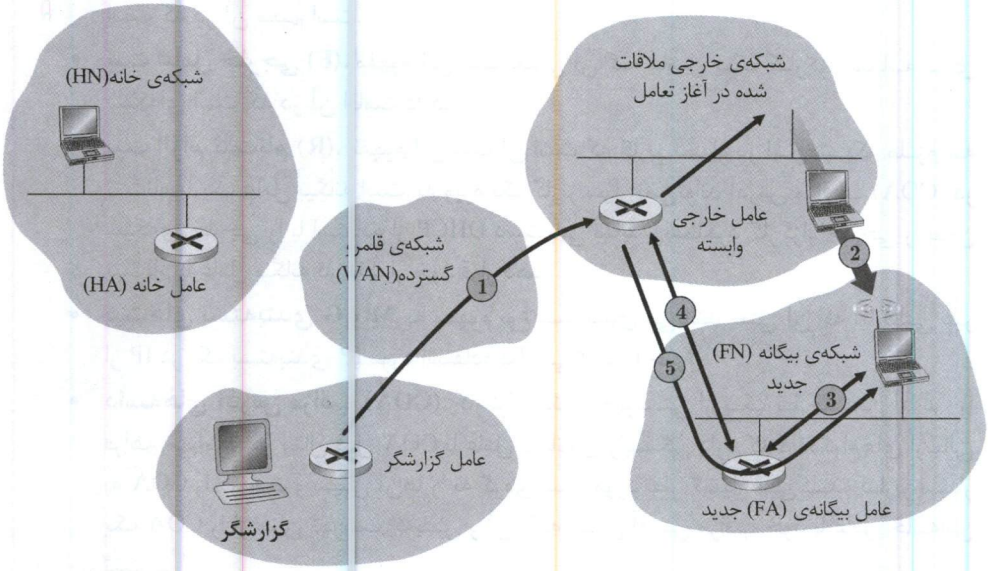
۶-۱۲ پروتکل IP سیار

معماری اینترنت و پروتکل‌های پشتیبانی تحرک مجموعاً به نام **پروتکل IP سیار** شناخته می‌شوند و در مستند [RFC3344] پروتکل IPV4 تعریف شده‌اند. پروتکل IP سیار یک استاندار انعطاف پذیر است و بسیاری از شیوه های مختلف مانند عملیات با و یا بدون عامل بیگانه، اعمال شیوه‌های چندگانه برای عامل ها و گره‌های سیار جهت بازیابی یکدیگر، بهره‌گیری از COA یکتا و چندگانه و انواع قالب‌های بسته‌بندی را پشتیبانی می‌کند. با چنین توان‌مندی‌هایی پروتکل IP سیار یک استاندارد پیچیده است و نیاز به توضیحات مفصل و فراگیر در این زمینه دارد. هدف ما در این مقطع تحلیل مهمترین چارچوب و اصول پروتکل IP سیار و نمونه‌هایی ازکاربردهای آن می‌باشد. معماری IP سیار شامل بسیاری از عناصری است که در بخش‌های پیشین به آنها اشاره رفت. این عناصر عبارت بودند از: کارگزاران خانه، بیگانه، مراقب‌های آدرس(COA) و بسته‌بندی و باز گشایی بسته ها. استاندارد جاری [RFC3344] به تعریف کاربرد مسیردهی غیر مستقیم گره‌های سیار می‌پردازد. استانداردIP سیار شامل سه بخش اساسی در قالب ذیل است:

۱- **شناسایی عامل.** پروتکل IP سیار رویکردهایی را تعریف می‌کند که از سوی کارگزاران خانه و بیگانه جهت آگهی خدمات آنها به گره های سیار بکار گرفته می‌شوند. علاوه بر این پروتکل‌هایی را نیز تعریف می کند که گره‌های سیار از طریق آنها خدمات ارایه شده از سوی این کارگزاران را مورد استفاده قرار می دهند.

۲- **ثبت نام با عامل خانه.** پروتکل IP سیار شیوه‌هایی را تعریف می کند که گره ی سیار و یا عامل بیگانه از آنها جهت ثبت نام COAبا یک عامل خانه‌ی گره‌ی سیار و یا عکس آن استفاده می‌کند.

۳- **مسیردهی غیرمستقیم داده‌گرام‌ها.** استاندارد پروتکل‌های IP سیار، شیوه‌ای که داده‌گرام‌ها به گره‌های سیار از طریق یک عامل گسیل می‌شوند را تعریف می‌کند. این رویکردها شامل قواعدی برای گسیل داده‌گرام‌ها، سنجش شرایط خطا و چندین قالب از عملیات بسته‌بندی بر مبنای دو مستند [RFC2003 RFC2004] می‌باشند.



شکل ۶-۲۶ ♦ انتقال سیار بین شبکه‌های با مسیردهی مستقیم

در سراسر استاندارد IP سیار اعمال سیاست‌های امنیتی بسیار چشمگیر و پر اهمیت می‌باشند. به عنوان مثال اعتبارسنجی یک گره‌ی سیار جهت اطمینان از عدم ثبت COA مخرب و مجعول از سوی مهاجمان با یک عامل خانه می‌باشد که میتواند باعث گردد همه‌ی داده‌گرام‌های هدایت شده به یک آدرس IP به سوی آن مهاجم تغییر مسیر دهند. شیوه‌هایی که IP سیار از آنها جهت پیاده‌سازی دغدغه‌های امنیتی خود بهره می‌گیرد در فصل ۸ بررسی شده‌اند.

• شناسایی عامل

یک گره‌ی IP سیار تازه وارد به یک شبکه بدون توجه به این نکته که آیا متصل به یک شبکه‌ی بیگانه است و یا آنکه به شبکه‌ی خانه‌ی خود بازگشت می‌نماید، ضروری است که شناسه‌ی عامل خانه یا بیگانه را دست‌یابی نماید. یقیناً کشف عامل خارجی جدید با یک آدرس شبکه‌ی جدید این امکان را ایجاد می‌کند که لایه‌ی شبکه در یک گره‌ی سیار از حرکت خود با درون یک شبکه‌ی خارجی جدید

آگاه شود. این فرآیند معروف به **کشف عامل**[1] می‌باشد. به دو شیوه امکان کشف عامل میّسر است. این دو عبارتند از **آگهی عامل**[2] یا **تقاضای عامل**[3].

در شیوه‌ی آگهی عامل یک کارگزار خانه یا بیگانه خدمات خود را با بهره‌گیری از پروتکل کشف مسیریاب در مستند [RFC 1256] اطلاع رسانی می‌کند. عامل متناوباً یک پیام ICMP را با نوع دامنه‌ی با ارزش 9 که ویژه‌ی کشف مسیریاب است بر روی همه‌ی پیوندهایی که به آن‌ها متصل است، فراپخش می‌کند. پیام کشف مسیریاب همچنین شامل یک آگهی تحرک عامل است که اطلاعات کافی مورد نیاز گره‌ی سیار را فراهم می‌آورد. در جریان دامنه‌ها عوامل مهم ذیل نیز قرار دارند:

- **بیت عامل خانه (H).** مفهوم این بیت آن است که کارگزار یک عامل خانه برای شبکه‌ای است که در آن مقیم است.
- **بیت عامل خارجی (F).** مفهوم این بیت یعنی آن که عامل یک کارگزار بیگانه برای شبکه‌ای است که در آن اقامت دارد.
- **بیت الزام ثبت‌نام (R).** مفهوم این بیت آن است که کاربر سیار در این شبکه ملزم به ثبت‌نام با یک عامل بیگانه است. به ویژه یک کاربر سیار نمی‌تواند آدرس مراقب (COA) در شبکه‌ی خارجی را با استفاده از DHCP دست‌یابی نماید و عملکرد کارگزار خارجی را بدون ثبت‌نام با عامل بیگانه در اختیار خود قرار دهد.
- **بیت‌های بسته‌بندی G و M.** به مفهوم نوع بسته‌بندی می‌باشد یعنی آن که آیا قالبی غیر از IP در یک بسته‌بندی IP مورد استفاده قرار می‌گیرد یا خیر؟
- **دامنه‌های آدرس مراقب (COA).** کارگزار بیگانه فهرستی از یک یا چندین آدرس را فراهم می‌آورد. در مثال ذیل COA با عامل بیگانه‌ای وابستگی دارد که داده‌گرام‌های ارسالی به COA را دریافت و سپس آن‌ها را به گره‌ی سیار مورد نظر گسیل می‌کند. کاربر سیار یک COA را از میان فهرست گزینش و آن را به عنوان آدرس مراقب خود با عامل خانه‌اش ثبت می‌کند.

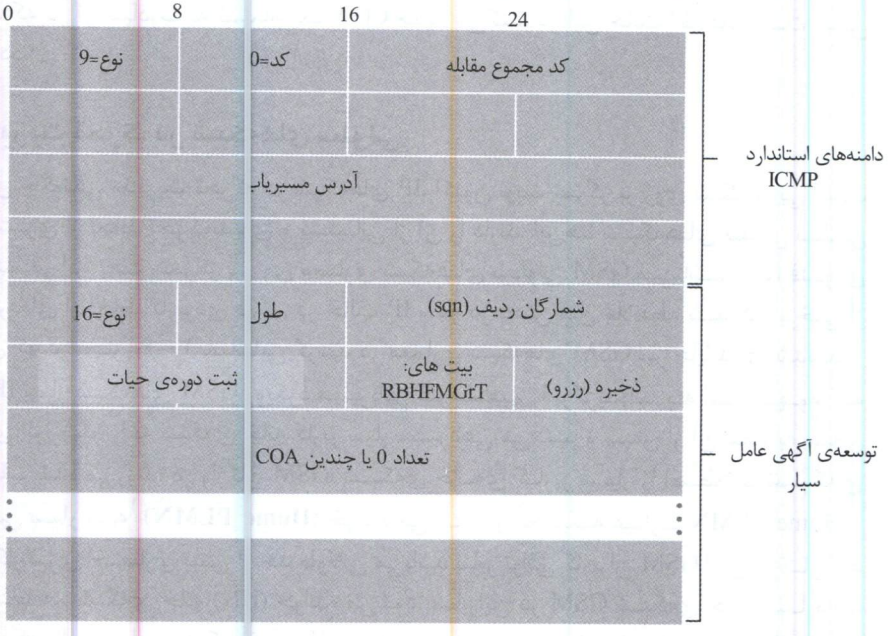
شکل ۶-۲۷ برخی از دامنه‌های کلیدی در پیام آگهی عامل را نشان می‌دهد. در **شیوه‌ی تقاضای عامل** گره‌ی سیار مایل به یادگیری بی‌درنگ دریافت یک آگهی عامل پیام تقاضای یک عامل را می‌تواند فراپخش نماید. این پیام به سادگی یک پیام ICMP با ارزش نوع 10 می‌باشد. کارگزار دریافت کننده‌ی تقاضا آگهی عامل را مستقیماً به گره‌ی سیار تک پخش می‌کند. پس از آن گره می‌تواند، از این موضوع که آیا یک آگهی ناخوانده دریافت نموده است یا نه اطمینان حاصل نماید.

- **ثبت‌نام با عامل خانه**

هنگامی که گره‌ی سیار IP یک آدرس COA را دریافت نماید، آن آدرس بایستی با عامل خانه ثبت شود. این مهم از دو شیوه امکان‌پذیر است. نخست از طریق کارگزار بیگانه که COA را با عامل خانه ثبت می‌کند و دوم مستقیماً از سوی گره‌ی سیار IP. در ذیل روش نخست در قالب انجام چهار مرحله تحلیل می‌شود:

1- Agent discovery
2- Agent advertisment
3- Agent solicitation

۱- به دنبال دریافت آگهی بیگانه، گره‌ی سیار پیام ثبت IP سیار را به آن کارگزار گسیل می‌کند. پیام ثبت در درون یک داده‌گرام UDP حمل و تحویل درگاه 434 می‌شود. پیام حامل اطلاعات: ثبت آدرس، اطلاعات COA آگهی شده از سوی عامل بیگانه، آدرس کارگزار خانه (HA)، آدرس دایمی گره‌ی سیار (PA)، دوره ی زمان حیات ثبت آدرس (TTL) و یک شناسه‌ی ثبت 64 بیت می‌باشد. دوره‌ی زمان ثبت تقاضا شده عددی است بر مقیاس ثانیه که اطلاعات ثبت شده، در طول آن زمان فقط اعتبار دارد. اگر ثبت اطلاعات مجدداً در عامل خانه در طول حیات تجدید نگردد آنگاه اطلاعات ثبت شده از اعتبار ساقط می‌شود. شناسه‌ی ثبت همانند یک شماره ردیف ثبتی است که از آن برای نگاشت پاسخ ثبت‌نام دریافتی با یک تقاضای ثبت بهره‌گیری می‌شود، که در گام‌های بعدی توضیح داده شده است.



شکل ۶-۲۷ ♦ پیام بازیابی **ICMP** مسیریاب به همراه آگهی افزوده‌ی عامل سیار

۲- کارگزار بیگانه پیام ثبت‌نام را دریافت می‌دارد و آدرس IP دایمی گره‌ی سیار را ثبت می‌کند. عامل بیگانه اکنون می‌داند که مجبور به کاوش داده‌گرام‌هایی است که آدرس مقصد در بسته‌بندی آن‌ها با آدرس دایمی گره‌ی سیار نگاشت می‌شود. کارگزار بیگانه سپس یک پیام ثبت IP سیار که درون یک داده‌گرام از نوع UDP است را به درگاه 434 در عامل خانه تحویل می‌نماید. این پیام حامل اطلاعات COA، HA، MA، تقاضای قالب بسته‌بندی، دوره‌ی زمان حیات ثبت تقاضا و شناسه‌ی ثبت می‌باشد.

۳- کارگزار خانه تقاضای ثبت را دریافت می‌دارد. آن را اعتبارسنجی می‌کند و از سالم بودن آن نیز اطمینان حاصل می‌کند. سپس کارگزار خانه دو آدرس COA و آدرس IP دایمی گره‌ی سیار را به یکدیگر وابسته می‌کند. پس از آن هر زمان داده‌گرامی دارد عامل خانه و به سوی گره‌ی سیار هدایت شده باشد، آن داده‌گرام به سوی COA تونل‌زنی و بسته‌بندی می‌شود. کارگزار خانه یک

پاسخ ثبت IP سیار را گسیل می‌نماید که حامل اطلاعات HA، MA، دوره‌ی حیات واقعی ثبت و شناسه‌گر ثبت تقاضایی است که برای این پاسخ رضایت بخش خواهد بود.

۴- کارگزار خارجی پاسخ ثبت را دریافت می‌دارد و سپس آن را به سوی گره‌ی سیار گسیل می‌کند.

در این نقطه است که مراحل ثبت‌نام نهایی می‌شود و گره سیار از آن به بعد می‌تواند داده‌گرام‌های هدایت شده به سوی آدرس دایمی خود را دریافت نماید. شکل ۶-۲۸ این مراحل را نشان می‌دهد. بدیهی است کارگزار خانه یک دوره‌ی زمان حیات (TTL) که مقدار آن کمتر از زمان حیات تقاضا شده از سوی گره‌ی سیار است را تعریف می‌کند. لزوماً یک کارگزار خارجی نیاز به حذف اطلاعات ثبتی با COA در هنگام ترک گره‌ی سیار از شبکه ندارد. این فرآیند به گونه‌ی خودکار انجام می‌شود. هنگامی که گره‌ی سیار به شبکه‌ی جدیدی وارد می‌شود این شبکه ممکن است یک شبکه‌ی خانه و یا بیگانه‌ی دیگری باشد که در آن مبادرت به ثبت‌نام یک COA جدید می‌کند، در این حالت اطلاعات ثبت قبلی حذف می‌گردد.

۶-۱۳ مدیریت تحرک در شبکه‌های سلولی

پس از تحلیل چگونگی مدیریت تحرک در شبکه‌های IP، اکنون نوبت تمرکز بر روی شبکه‌هایی است که قدمت بیشتری از لحاظ تحرک‌پذیری و پشتیبانی از آن را دارند. این‌ها شبکه‌های تلفنی سلولی سیار می‌باشند. در این بخش تمرکز بر روی معماری شبکه‌های سلولی GSM می‌باشد زیرا فناوری کامل و گسترده‌ای از لحاظ کاربردی دارند. در حالت IP سیار در بخش قبل ملاحظه شد که برخی از اصول بنیادی که در بخش ۶-۱۰ شناسایی گردید در معماری شبکه‌های GSM نیز جایگذاری شده‌اند.

همانند IP سیار شبکه‌های GSM رویکرد مسیریابی غیرمستقیم را برگزیده‌اند. به این مفهوم که ابتدا مکالمه‌ی گزارشگر را به شبکه‌ی خانه کاربر سیار مسیردهی می‌کنند و سپس از آن نقطه به سوی شبکه‌ی ملاقات شده می‌روند. در واژگان GSM، شبکه‌ی خانه‌ی کاربر سیار را اصطلاحاً **شبکه‌ی قلمرو عمومی سیارخانه[1] (Home PLMN)** خوانده می‌شود. از آنجائیکه عبارت Home PLMN از نقطه نظر به‌کارگیری اختصاری بیش از حد طولانی می‌باشد بنابر توافق کاربران GSM این واژگان به اختصار در اینجا هم شبکه‌ی خانه (HN) خوانده می‌شود. بنابراین در GSM شبکه‌ی خانه اشاره به سلولی می‌کندکه کاربر سیار مشترک آن می‌باشد و همه ماهه صورت حساب‌های کاربری از طریق آن سلول محاسبه می‌شود. شبکه‌ی PLMN ملاقات شده به اختصار شبکه‌ی ملاقات (VN) شده خوانده می‌شود و عبارت است از شبکه‌ای که کاربر سیار GSM در حال حاضر در آن ساکن است. همانند حالت IP سیار مسئولیت‌های دو شبکه‌ی خانه و ملاقات شده کاملاً متفاوت از یکدیگر می‌باشند و این تفاوت‌ها را می‌توان در قالب ذیل برشمرد:

۱- شبکه‌ی خانه بانک اطلاعاتی را نگهداری می‌کند که **ثبات موقعیت خانه[2] (HLR)** نام دارد. این پایگاه حامل شماره تلفن دایمی، اطلاعات شناسایی هر یک از مشترکین می‌باشد. علاوه بر آن HLR حامل اطلاعات موقعیت‌های جاری مشترکین نیز می‌باشد. به بیان دیگر اگر یک کاربر سیار در حال کاوش و ردیابی شبکه‌ی سلولی دیگری می‌باشد، HLR حامل اطلاعات کافی جهت بازیابی یک آدرس در شبکه‌ی ملاقات شده‌ای است که یک مکالمه از کاربری سیار

1- Home public land mobile networks
2- Home location register

بایستی مسیردهی شود. در ادامه ملاحظه می شود برای انجام این مهم از یک راه‌یاب ویژه در شبکه‌ی خانه که اصطلاحاً به آن **مرکز راه‌یابی خدمات سیار دریچه**[1] **(GMSC)** گفته می‌شود، بهره‌گیری شده تا گزارشگر یک مکالمه را به یک کاربر سیار ارتباط دهد. لازم به توضیح است که معمولاً GMSC را **MSC خانه** نیز می‌نامند.

۲- شبکه‌ی ملاقات شده نیز پایگاه داده ای را نگهداری می‌کند که به آن **ثبات موقعیت ملاقات کننده**[2] **(VLR)** گفته می‌شود. پایگاه VLR شامل اطلاعات هر کاربر سیاری است که به شکل جاری از بخشی از امکانات شبکه به وسیله VLR سرویس دهی شده است. اطلاعات VLR بنابراین پویا می باشد. هر گاه یک کاربر سیار وارد شبکه و یا از آن خارج می‌شود این اطلاعات نیز با توجه به فعالیت آن وارد و یا خارج می‌شود. معمولاً VLR هم مکان با MSC است که هماهنگ کننده برقراری ارتباط با و یا از شبکه‌ی ملاقات شده می‌باشد. در عمل یک اوپراتور شبکه‌ی سلولی حالت یک شبکه‌ی خانه را برای مشترکین خویش دارد و برای کاربران سیار دیگر که مشترک اپراتورهای دیگر می‌باشند حالت شبکه‌ی ملاقات شده را اختیار می‌کند.

۶-۱۴ مسیردهی مکالمه‌ها به یک کاربر سیار

در این مقطع چگونگی واگذاری یک مکالمه به یک کاربر سیار GSM در یک شبکه‌ی ملاقات شده تحلیل می‌شود. مثال ساده‌ی این کاربر در قالب شکل ۶-۲۹ به صورت سه مرحله‌ی عملیاتی توصیف شده است:

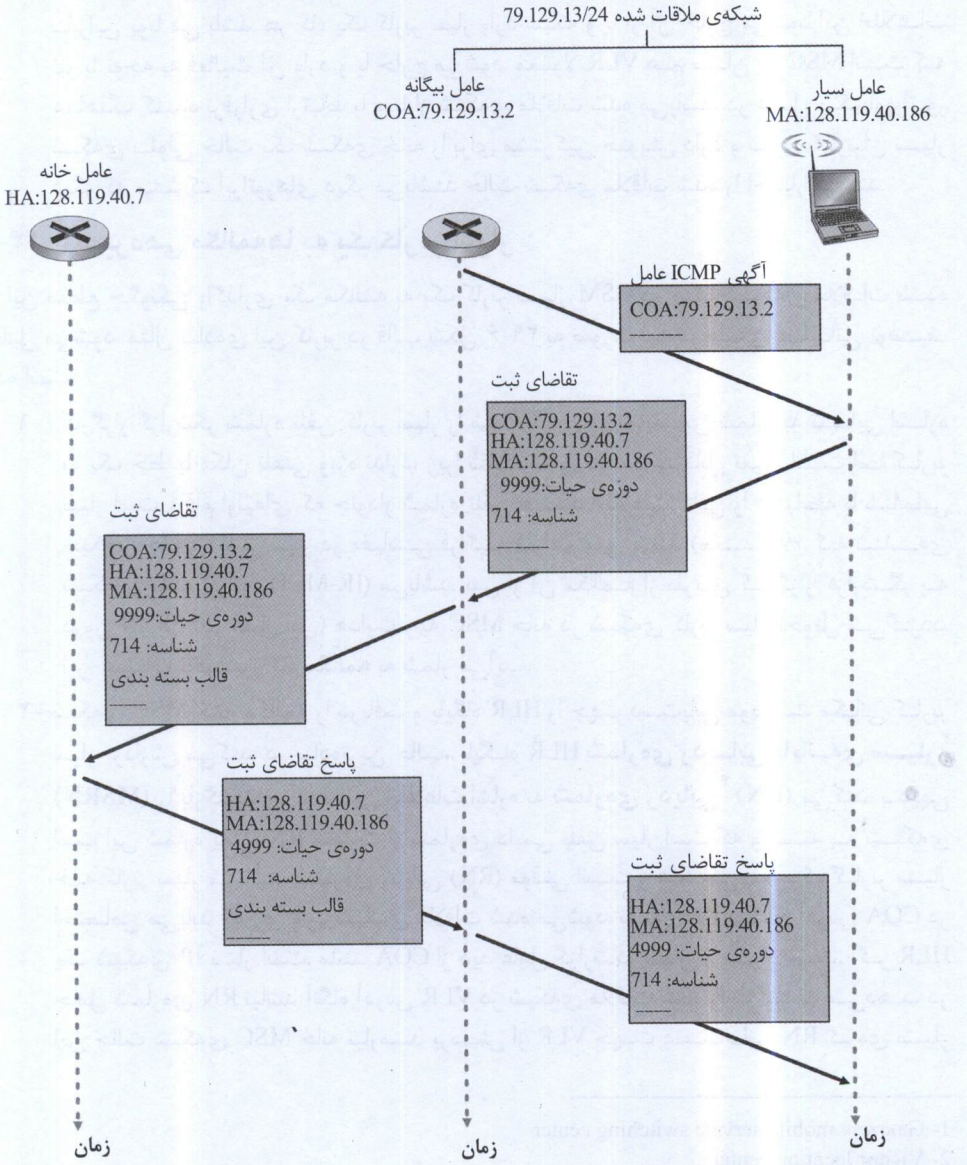
۱- کارگزار گزارشگر شماره تلفن کاربر سیار را شماره‌گیری می‌نماید. این شماره به تنهایی اشاره به یک خط یا مکان تلفنی ویژه ندارد، زیرا شماره تلفن ثابت و غیرقابل تغییر است اما کاربر سیار است. ارقام اولیّه‌ای که جلودار شماره تلفن هستند اطلاعات کافی را در رابطه با شناسایی شبکه‌ی خانه‌ی کاربر سیار در مقیاسی فراگیر فراهم می‌آورند (مانند 98+ که شناسه‌ی شبکه‌ای مانند اپراتور (IR-MCI) می‌باشد. پس از آن مکالمه از طریق کارگزار گزارشگر به درون PSTN (در فصل اول) هدایت و به MSC خانه در شبکه‌ی کاربر سیار تحویل می‌گردد. این عملیات نخستین گام مکالمه به شمار می‌آید.

۲- شبکه‌ی MSC خانه مکالمه را دریافت و پایگاه HLR را جهت دست‌یابی موقعیت مکانی کاربر سیار پردازش می‌کند. در ساده‌ترین حالت، پایگاه HLR **شماره‌ی ردیابی پایانه‌ی سیار**[3] **(MSRN)** را بازگشت می‌دهد. این اطلاعات اشاره به **شماره‌ی ردیابی**[4] **(RN)** می‌کند. بدیهی است این شماره یعنی RN متفاوت از شماره‌ی دایمی تلفن سیار است که وابسته به شبکه‌ی خانه کاربر سیار می‌باشد. شماره‌ی ردیابی (RN) موقتی است و هنگامی به یک کاربر سیار اختصاص می‌یابد که وی وارد شبکه‌ی ملاقات شده می‌شود. نقش RN مشابه آدرس COA در یک شبکه‌ی IP سیار است. مانند COA از دید عامل گزارشگر پنهان و سیار است. اگر HLR حامل شماره‌ی RN نباشد آنگاه آدرس VLR در شبکه‌ی ملاقات شده را بازگشت می‌دهد. در این حالت شبکه‌ی MSC خانه نیازمند پرسش از VLR جهت دست‌یابی RN گره‌ی سیار

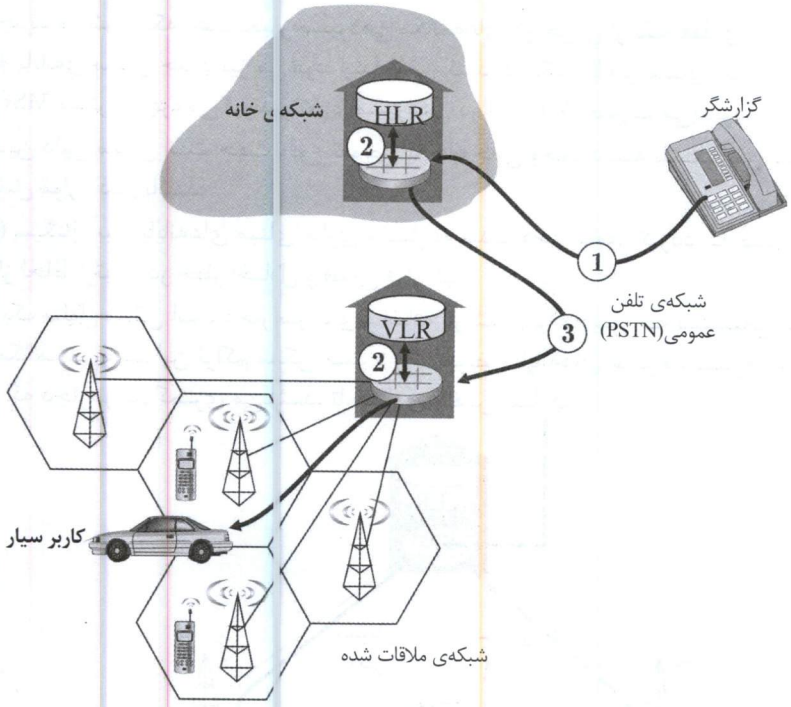
1- Gateway mobile service switching center
2- Visitor location center
3- Mobile station roaming number
4- Roaming number

می‌باشد. اما چگونه HLR شماره ردیابی و یا آدرس VLR را شناسایی و اخذ می‌کند؟ چه اتفاقی برای این ارزش‌ها هنگامی که کاربری سیار از یک شبکه ملاقات شده به شبکه‌ی دیگر نقل مکان می‌کند روی می‌دهد؟ پاسخ این پرسش‌ها در ادامه آمده است.

۳- با در اختیار داشتن RN، شبکه‌ی MSC خانه گام دوم مکالمه را از درون شبکه به MSC در شبکه‌ی ملاقات شده بر می‌دارد. هنگام مسیردهی مکالمه از گزارشگر به MSC خانه و از آنجا به MSC ملاقات شده و سپس به پایانه‌ی مبنا که کاربر سیار را خدمات‌رسانی می‌کند، مرحله‌ی پایانی و نهایی تکمیل یک مکالمه به شمار می‌آید.



شکل ۶-۲۸ ♦ آگهی عامل و ثبت‌نام **IP** سیار



شکل ۶-۲۹ ♦ واگذاری یک مکالمه به یک کاربر سیار با بهره‌گیری از مسیردهی غیر مستقیم

یک پرسش بدون پاسخ در گام 2 دست‌یابی اطلاعات در مورد مکان کاربر سیار از سوی HLR می‌باشد. هنگامی که دستگاه تلفن گوشی همراه روشن می‌شود و یا آن که وارد بخشی از یک شبکه‌ی ملاقات شده‌ای می‌شود که VLR جدیدی را پوشش می‌دهد. آن گوشی همراه با شبکه‌ی ملاقات شده ثبت‌نام می‌نماید. این مهم از طریق تبادل پیام‌های سیگنالینگ، بین گوشی همراه و VLR انجام می‌شود. پایگاه داده‌ی VLR ملاقات شده آنگاه پیام درخواست به روزرسانی یک موقعیت را به پایگاه HLR گوشی همراه گسیل می‌کند. این پیام HLR گوشی همراه متقاضی مکالمه را از شماره RN گوشی همراه طرف مخاطب و مکالمه و یا آدرس VLR که از طریق آن می‌توان RN مخاطب را اخذ نمود، آگاه می‌کند. و به عنوان بخشی از تبادل پایگاه VLR اطلاعات مربوط به مشترک را از HLR همراه دست‌یابی می‌نماید و سپس خدمات قابل واگذاری به کاربر سیار در شبکه‌ی ملاقات شده را تعریف می‌کند.

۶-۱۵ تعویض در GSM

یک **جابه‌جایی**[1] در GSM هنگامی به وقوع می پیوندد که یک پایانه‌ی سیار وابستگی (شرکت‌پذیری) خویش را از یک پایانه به دیگری در هنگام انجام مکالمه تغییر می دهد. همانگونه که در شکل ۶-۳۰ نشان داده شده است یک مکالمه‌ی سیار در آغاز و قبل از تعویض از درون یک پایانه‌ی مبنا مسیردهی می‌شود و پس از جابه‌جایی از طریق پایانه‌ی مبنای دیگری به آن گوشی همراه هدایت می‌شود. بدیهی است نتیجه‌ی حاصل از جابه‌جایی بین پایانه‌های مبنا نه تنها سبب ارسال و دریافت بین پایانه‌های

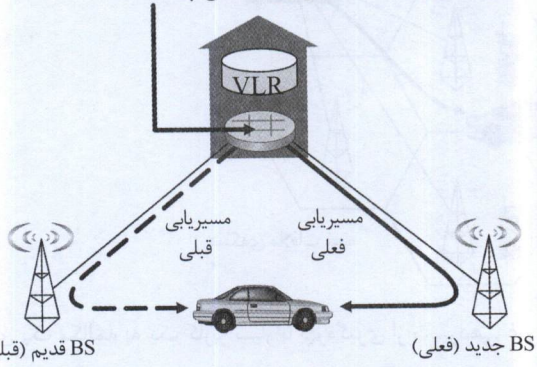
1- Handoffs

مبنای جدید می‌شود بلکه باعث تغییر مسیردهی مکالمه‌های در جریان از یک نقطه‌ی راه‌یابی در درون شبکه به پایانه‌ی مبنای جدید نیز می‌شود. ابتدا فرض کنید که یک پایانه‌ی مبنای قدیم و جدید هر دو از یک MSC مشترک بهره می‌گیرند و تغییر مسیردهی در این MSC صورت می‌پذیرد.

چندین دلیل ممکن است جهت وقوع تعویض یا جابه‌جایی وجود داشته باشد. این دلایل ممکن است شامل موارد ذیل باشند:

الف) سیگنال بین پایانه‌های مبنای جاری و سیار به شدت دچار ضعف گردد به صورتی که یک مکالمه از لحاظ ارتباط در خطر اختلال و قطعی قرار گیرد.

ب) یک سلول ممکن است دچار سرریزی و اضافه بار شود زیرا مجبور به پشتیبانی تعداد نسبتاً انبوهی مکالمه می‌باشد. این تراکم ممکن است در اثر تعویض پایانه‌های همراه (سیار) به سلول‌های مجاوری که دچار تراکم کمتری می‌باشند، تا حدودی کاهش پیدا کند.



شکل ۶-۳۰ ♦ موارد تعویض بین پایانه‌های مبنا در یک **MSC** مشترک

معمولاً یک گوشی همراه به شکل متناوب قدرت سیگنال‌های ره‌گیری (بیکن) را از مکان پایانه‌ی مبنای خود و نسز سایر پایانه‌های مبنای مجاور که قادر به دریافت آن‌ها می‌باشد، اندازه‌گیری می‌کند. این مقادیر اندازه‌گیری شده یک یا دو مرتبه به پایانه‌ی مبنای جاری گوشی همراه گزارش می‌شود. جابه‌جایی در GSM معمولاً از سوی پایانه‌ی قدیم بر اساس این اندازه‌گیری‌ها صورت می‌پذیرد. بار جاری ترافیک گوشی‌های همراه در سلول‌های مجاور و تأثیر عوامل دیگر به اعمال این تعویض کمک می‌کند. استاندارد شبکه‌های GSM الگوریتم ویژه‌ای را به منظور کاربرد از سوی پایانه‌ی مبنا برای اعمال جابه‌جایی تعریف نکرده است، اما بر اساس آنچه در شکل ۶-۳۱ نشان داده شده است می‌توان مراحل عملیاتی که یک پایانه‌ی مبنا جهت تعویض یک کاربر سیار اعمال می‌کند را به خوبی ملاحظه نمود. این مراحل عبارتند از:

۱- پایانه‌ی مبنای (BS) قدیم شبکه‌ی MSC ملاقات شده را از وقوع تعویض آگاه می‌کند و یک یا چندین BS را که در جابه‌حایی کاربر سیار دخالت داشته‌اند، معرفی می‌نماید.

۲- شبکه‌ی MSC ملاقات شده اقدام به برقراری مسیری به نام BS جدید می‌نماید و منابع مورد نیاز جهت حمل مکالمه‌ی مسیردهی شده را تامین می نماید و به BS جدید سیگنال لازم جهت تعویض قریب‌الوقوع کاربر سیار را گسیل می‌کند.

۳- پایانه‌ی مبنای جدید (BS) یک کانال رادیویی برای بهره‌گیری کاربر سیار را تامین و آن را فعال می‌سازد.

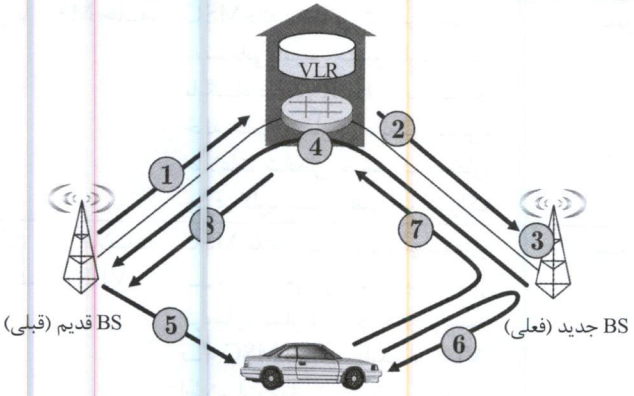
۴- پایانه‌ی BS جدید سیگنالی را به MSC ملاقاتْ شده بازگشت می‌دهد و BS قدیم MSC ملاقات شده را به BS جدید ارتباط می‌دهد و سپس کاربر سیار از جابه‌جایی قریب‌الوقوع آگاه می‌شود. پایانه‌ی BS جدید همه‌ی اطلاعات مورد نیاز گوشی همراه را که برای وابستگی به BS جدید مورد نیاز می‌باشد، تامین می‌نماید.

۵- کاربر سیار از انجام عملیات جابه‌جایی آگاه می‌شود. بدیهی است تا این مقطع گوشی همراه از تمامی زمینه چینی‌های شبکه مانند اختصاص کانال به BS جدید و تعیین مسیر از MSC ملاقات شده به BS جدید به منظور انجام عملیات تعویض بی‌خبر می‌باشد.

۶- در این نقطه از مراحل عملیاتی گوشی همراه و BS جدید به مبادله‌ی یک یا چندین پیام جهت فعال‌سازی کامل کانال جدید در BS تازه تعریف شده می‌پردازند.

۷- کاربر سیار آنگاه یک پیام تکمیل تعویض را به BS جدید ارسال می‌کند. این پیام تا MSC ملاقات شده پیش می‌رود. پس از آن MSC ملاقات شده تغییر مسیر مکالمه‌ی در جریان به گوشی همراه را از طریق BS جدید انجام می‌دهد.

۸- منابع اختصاص یافته در امتداد مسیر به BS قدیم در این مقطع آزاد می‌شوند.



شکل ۶-۳۱ ♦ مراحل تکمیل یک تعویض بین پایانه‌های مبنا با یک **MSC** مشترک

مطالب مربوط به جابه‌جایی با در نظر گرفتن حالت MSC های متفاوت در برابر نوع مشترک آن ادامه می‌یابد. حالتی را فرض کنید که در آن یک کاربر همراه به یک BS جابه‌جا می‌شود که وابسته به MSC متفاوتی نسبت به BS قدیم می‌باشد. مراحل جابه‌جایی بین MSCها بیش از یک مرتبه اتفاق می‌افتد. همانگونه که شکل ۶-۳۲ نشان می‌دهد، GSM یک **MSC مرجع**[1] را تعریف می‌کند. این شبکه‌ی MSC مرجع عبارت است از MSC ملاقات شده از سوی کاربر همراه به هنگام آغاز یک مکالمه که در طول جریان مکالمه بدون تغییر باقی می‌ماند. در طول این مدت بدون توجه به تعداد نقل و انتقال‌های صورت گرفته بین MSCها از سوی کاربر سیار، مکالمه از MSC خانه به MSC مرجع مسیردهی و سپس از آن به MSC ملاقات شده هدایت می‌شود که کاربر سیار در حال حاضر در آن مقیم می‌باشد. هنگامی که کاربر سیار از یک قلمرو در پوشش یک MSC به MSC مرجع جابه‌جا می‌شود، مکالمه‌ی در جریان از MSC مرجع به MSC ملاقات شده‌ی جدید که شامل پایانه‌ی مبنای جدید است تغییر مسیر می‌دهد. بنابراین همواره حداکثر تعداد سه MSC یعنی خانه، مرجع و ملاقات

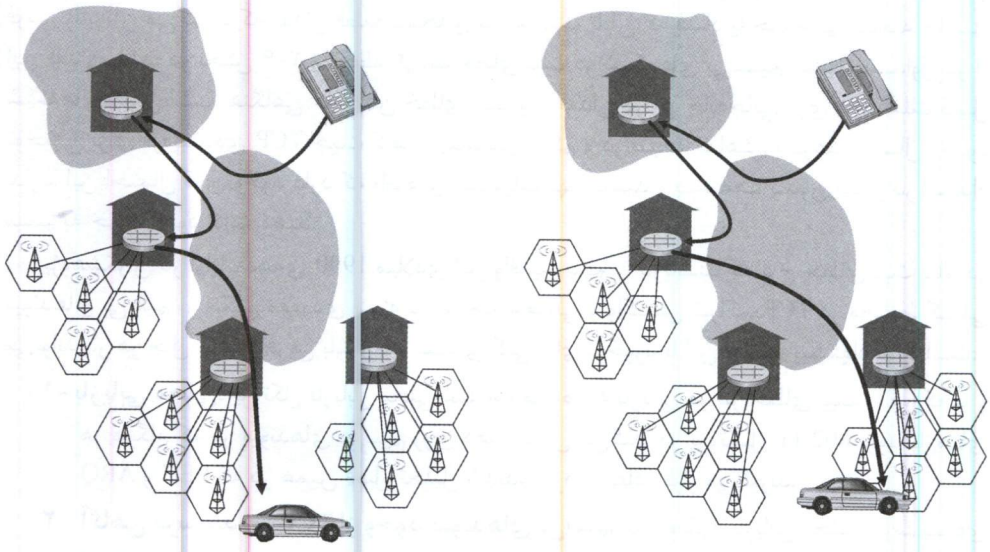
1- Anchor MSC

شده بین تبادل و کاربر سیار وجود دارد. شکل ۶-۳۲ مسیردهی یک مکالمه در بین MSC های ملاقات شده از سوی کاربر سیار را نشان می‌دهد.

شیوه‌ی جایگزین روش بالا، نگهداری یک MSC پرش کوتاه از MSC مرجع به MSC جاری آن است که MSC های ملاقات شده از سوی کاربر سیار (همراه) در پی یکدیگر ردیف (زنجیر) می‌شوند به این مفهوم که یک MSC قدیمی، مکالمه‌ی در جریان را هر مرتبه که کاربر جابه‌جا می‌شود به یک MSC جدید گسیل می‌نماید. جدول ۶-۲ مقایسه‌ی بین مدیریت تحرک در GSM و IP سیار را نشان می‌دهد. اطلاعات این جدول نشان از آن دارد اگر چه IP و شبکه‌های سلولی از لحاظ بنیادی متفاوت می‌باشند اما در تعدادی از عناصر عملکردی و نیز پیاده‌سازی تحرک مشترک می‌باشند.

جدول ۶-۲ ♦ وجوه اشتراک بین **IP** سیار و تحرک **GSM**

عنصر GSM	توضیح عنصر GSM	عنصر IP سیار
سیستم خانه	شبکه‌ای که شماره تلفن دایمی کاربر سیار به آن تعلق دارد.	شبکه‌ی خانه
مرکز راه‌یابی سیار یا MSC خانه، ثبات (HLR)	MSCخانه: نقطه‌ی تماس جهت دست‌یابی آدرس مسیر کاربر سیار پایگاه داده‌ی HLR در سیستم خانه حامل شماره‌ی دایمی تلفن، اطلاعات شخصی مشترک، مکان جاری کاربر سیار، اطلاعات اشتراک	عامل خانه
مرکز راه‌یابی خدمات سیار ملاقات شده، ثبت ملاقات کننده (VLR)	MSC ملاقات شده: عهده‌دار مسئولیت برقراری ارتباط به و یا از گره‌های سیار در سلول‌های وابسته به MSC است. پایگاه VLR: اطلاعات موقت سیستم ملاقات شده شامل اطلاعات اشتراک هرکاربر سیار ملاقات کننده را نگه‌داری می‌کند.	شبکه‌ی عامل بیگانه
سیستم ملاقات شده	شبکه‌ای که کاربر سیار در حال حاضر در آن مقیم است. سیستم خانه شامل این مورد نمی‌شود.	شبکه‌ی ملاقات شده
شماره ردیابی پایانه‌ی سیار (MSRN) یا همان (RN)	آدرس قابل مسیردهی برای مکالمه‌ی تلفنی بین MSC خانه ملاقات شده، پنهان از عامل گزارشگر و کاربر سیار می‌باشد.	آدرس مراقب (COA)



شکل ۶-۳۲ ♦ مسیردهی از طریق **MSC** جایگذاری شده

۶-۱۶ بی‌سیم و تحرک: تاثیر پروتکل‌های لایه‌ی بالاتر

تاکنون در این فصل ملاحظه شد که شبکه‌ها بی‌سیم به گونه‌ی چشمگیری متفاوت از شبکه‌های کابلی در هر دو موردلایه‌ی پیوند و شبکه می‌باشند. اما آیا تفاوت‌هایی نیز در لایه‌های کاربرد و انتقال نیز وجود دارد؟ ظاهراً به نظر می‌رسد اختلاف مهمی بین این دو وجود ندارد زیرا لایه‌ی شبکه بهترین تلاش را در تحویل به لایه‌های بالاتر در هر دو شبکه‌ی بی‌سیم و کابلی از خود نشان می‌دهد. به صورت مشابه اگر پروتکل‌هایی مانند TCP و یا UDP نیز مورد بهره‌برداری جهت تامین خدمات لایه انتقال به کاربردها در هر دو شبکه قرار گیرند، آنگاه لایه‌ی کاربرد نیز بایستی بدون تغییر باقی بماند. از یک نقطه‌نظر TCP و UDP می‌توانند در شبکه‌های شامل پیوندهای بی‌سیم نیز کارایی خود را حفظ نمایند. از سوی دیگر پروتکل‌های لایه‌ی انتقال به طور کلی و TCP به ویژه در اغلب موارد می‌توانند کارایی بسیار متفاوت در شبکه‌های بی‌سیم و کابلی داشته باشند و از نقطه‌نظر کارایی است که تفاوت‌ها خود را نشان می‌دهند. علت آن در ذیل تشریح می‌شود.

یادآوری می‌شود پروتکل TCP در دو حالت به انتقال مجدد یک قطعه اقدام می‌نمود. این دو عبارت بودند از فقدان و یا آلوده شدن قطعه در مسیر بین فرستنده و گیرنده. از نقطه‌نظر کاربران سیار، فقدان یک قطعه داده هنگامی روی می‌دهد که شبکه دچار تراکم باشد یعنی بافر مسیریاب دچار سرریزی شده باشد و یا آن که در اثر تعویض به علت تاخیرها در تغییر مسیر، قطعه‌ها به یک نقطه‌ی جدید اتصال کاربران سیار به شبکه منتقل شوند. در همه‌ی این موارد، یک پیام ACK از گیرنده به فرستنده‌ی TCP فقط به بیان این نکته می‌پردازد که قطعه سالم دریافت نشده است به عبارت دیگر فرستنده آگاه نمی‌شود که آیا قطعه در اثر تراکم هنگام جابه‌جایی مفقود شده است یا به علت شناسایی خطای بیت. در هر حالتی واکنش فرستنده مشابه است، گسیل مجدد قطعه. واکنش کنترل تراکم TCP نیز در همه‌ی موارد یکسان است. پروتکل TCP پنج ره‌ی تراکم را بر اساس آنچه در بخش ۳-۲۲

توضیح داده شد، کاهش می‌دهد. از طریق کاهش غیر مشروط پنجره‌ی تراکم پروتکل TCP انحصاراً فرض را بر آن می‌گذارد که فقدان قطعه نتیجه‌ی تراکم است تا آن که فساد یا جابه‌جایی قطعه حاصل این فقدان باشد. در بخش ۶-۲ ملاحظه گردید خطای بیت در شبکه‌های بی‌سیم سیار متداول‌تر از شبکه‌های کابلی است. هنگامی که چنان خطای بیت و یا فقدان در اثر جابه‌جایی روی دهد، علت قابل توجیهی برای فرستنده‌ی TCP جهت کاهش پنجره‌ی تراکم و در نتیجه کاهش سرعت ارسال وجود ندارد. این احتمال قوی وجود دارد که بافرهای مسیریاب تهی باشند و بسته‌ها بدون مانع در امتداد مسیر به جریان خود ادامه دهند.

پژوهشگران در اوایل دهه‌ی 1990 میلادی این واقعیت مهم را دریافتند که نرخ خطای بیت بالا در پیوندهای بی‌سیم و امکان مفقودی بسته در اثر جابه‌جایی و واکنش تراکم TCP در محیط کابلی می‌تواند در هر حال مشکل‌آفرین باشد. سه شیوه‌ی کلی برای برخورد با این چالش پیشنهاد شده است:

۱- **بازیابی محلی.** پروتکل بازیابی محلی عملیات مربوط به بازیابی در اثر خطای بیت را هر کجا و هر هنگام که در پیوندهای بی‌سیم روی دهد، اعمال می‌کند. در پروتکل 802.11 پروتکل‌های ARQ و FEC که در همین فصل تحلیل شدند، مورد استفاده قرار می‌گیرند.

۲- **آگاهی فرستنده‌ی TCP از وجود پیوندهای بی‌سیم.** در رویکرد بازیابی محلی فرستنده‌ی TCP آگاهی از حرکت قطعه‌های داده بر روی یک پیوند بی‌سیم ندارد. شیوه‌ی جایگزین برای فرستنده و گیرنده‌ی TCP، آگاهی از وجود یک پیوند بی‌سیم جهت تمایز بین فقدان در اثر تراکم در شبکه‌های کابلی و یا فقدان و آلوده شدن در پیوندهای بی‌سیم و تحرک کنترل تراکم فقط در اثر فقدان قطعه‌ها در شبکه‌ی کابلی دچار تراکم می‌باشد.

۳- **رویکرد تقسیم و ممیزی ارتباط.** در این شیوه‌ی ارتباط بین کاربر سیار و طرف مقابل ارتباط لایه‌ی انتقال به دو بخش تقسیم می‌شود. یک ارتباط از میزبان سیار به نقطه‌ی دست‌یابی (AP) بی‌سیم است و دیگری از AP بی‌سیم به دیگر نقاط ارتباطی که ممکن است میزبان‌های کابلی باشند. ارتباط انتها به انتها بنابراین از ترکیب یک بخش بی‌سیم و کابلی تشکیل می‌شود. لایه‌ی انتقال در قطعه‌ی بی‌سیم می‌تواند ارتباط استاندارد TCP باشد و یا آن که یک پروتکل با طراحی ویژه‌ی بازیابی خطا برفراز UDP بررسی‌های انجام شده نشان می‌دهد شیوه‌ی تقسیم ارتباط TCP در شبکه‌های داده‌ی سلولی به صورت گسترده‌ای مورد استفاده قرار گرفته است و بهینه‌سازی چشمگیری را می‌توان در اثر این رویکرد انتظار داشت.

پس از بررسی پروتکل‌های لایه‌ی انتقال نکته‌ی بعدی بررسی و تاثیر وضعیت تحرک و بی‌سیم بر پروتکل‌های لایه‌ی کاربرد می‌باشد. در این حالت اهمیت در آن است که پیوندهای بی‌سیم اغلب پهنای باند (سرعت) نسبتاً کمتری را همانگونه که از شکل ۶-۲ مشاهده می‌شود، اختیار می‌کنند. در نتیجه کاربردهایی که بر روی پیوندهای بی‌سیم و به ویژه پیوندهای بی‌سیم سلولی عمل می‌کنند ارزش بسیاری را برای پهنای باند در نظر می‌گیرند. به عنوان مثال یک وب سرور که در خدمت یک مرورگر وب در حال اجرا بر روی گوشی همراه از نوع 3G می‌باشد به احتمال قوی قادر به تامین تصاویر غنی از لحاظ محتوا در مقایسه با مرورگری که بر روی ارتباط‌های کابلی پیاده‌سازی شده است، نمی‌باشد. اگرچه پیوندهای بی‌سیم چالش‌هایی را در لایه‌ی کاربرد ایجاد می‌کنند اما عامل تحرک و جابه‌جایی که فراهم می‌آورند، نیز مجموعه‌ای غنی از کاربردهای آگاه از موقعیت مکانی و چارچوب محتوا را نیز میّسر می‌سازد به طور کلی شبکه‌های بی‌سیم و سیار نقش بسیار کلیدی و حیاتی را در گسترش سریع

محیط‌های کامپیوتری آینده عهده‌دار خواهند بود. این اظهار نظر منطقی است که گفته شود در حال حاضر قله‌ی کوچکی از یک کوه عظیم که در آینده تاثیر و تحولات عمیقی را به همراه خواهد داشت، ملاحظه نموده‌ایم. این تحولات در مقیاس وسیع در زمینه‌ی پروتکل‌ها و کاربردهای شبکه‌های بی‌سیم و سیار هیجان‌انگیزتر خواهد بود.

۶-۱۷ شبکه‌های سلولی داده‌ی 3G: گسترش خدمات اینترنت به مشترکین ارتباطات همراه

تمرکز بحث در بخش ۶-۹ بر مبنای ارتباط کاربران سلولی صوتی با شبکه‌های تلفنی ثابت پایه‌ریزی شده است. بدیهی است هنگامی که یک کاربر از یک ارتباط سیار بهره‌مند می‌شود مایل به دستیابی صفحات وب، ایمیل‌های دریافتی و خدماتی از قبیل مکان‌یابی موقعیت‌های مورد علاقه نظیر هتل‌ها، رستوران‌ها، جاده‌ها و غیره نیز می‌باشد. حتی ممکن است مایل به تماشای بی‌درنگ تصاویر ویدیویی زنده و در حال پخش نیز باشد. برای رسیدن به این اهداف تلفن هوشمند او نیاز به اجرای کامل پشته‌ی پروتکلی TCP/IP شامل همه‌ی لایه‌های: فیزیکی، پیوند داده، شبکه، انتقال و کاربرد را دارد. علاوه بر همه‌ی این موارد امکان برقراری ارتباط اینترنتی از طریق شبکه‌ی سلولی داده نیز اهمیت بسیار دارد. موضوع شبکه‌های سلولی داده مجموعه‌ای از استانداردهای در حال تکامل و پویا و در عین حال رقابتی را به‌صورت نسل در نسل شامل می‌شود. به سخن دیگر هیچ نهاد مسئولی برای تنظیم نیازهای فناوری‌های نسل‌های 2G، 2.5G، 3G، 3.5G و یا 4G وجود ندارد و فقدان چنین سازمانی گزینش یک استاندارد از میان استانداردهای در حال رقابت را دشوار می‌سازد. در ادامه‌ی مطالب این بخش بیشترین تأکید روی خدمات ارتباطی سیار فراگیر یا UMTS و استانداردهای 3G است که از سوی نهاد **پروژه‌ی مشترک نسل سوم (3Gpp)** با عنوان [3GPP2012] مطرح شده است. یادآوری این نکته ضروری است قبل از گام نهادن به موارد مربوط به 3G ابتدا استاندارد اخیراً تدوین شده از سوی IEEE پیرامون شبکه‌های اختصاصی محلی نسل دوم معروف به Zigbee تحلیل می‌شود.

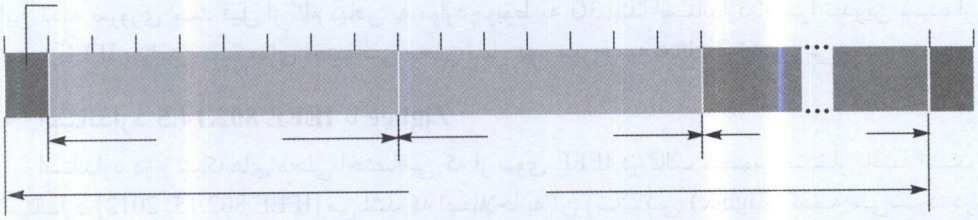
• استاندارد IEEE 802.14.5 یا Zigbee

استاندارد دوم شبکه‌های محلی اختصاصی که از سوی IEEE در قالب مصوب انتشار یافته است استاندارد [IEEE 802.15, 2012] می‌باشد که اصطلاحاً به آن زیگ بی (Zigbee) گفته می‌شود. در حالی که شبکه‌های بلوتوث یک نرخ داده‌ی **جایگزین کابلی** را به‌صورت 1Mbps تأمین می‌کنند. پروتکل Zigbee به‌منظور کاربردهای با سرعت، قدرت مصرف و ضریب کاری کمتری در مقایسه با بلوتوث طراحی شده‌اند. اگرچه به نظر ما هر چیزی که بزرگ‌تر و سریع‌تر است، بهینه و مطلوب به‌نظر می‌رسد اما همه‌ی کاربردهای شبکه نیازمند پهنای باند بیشتر و درنتیجه مخارج بالاتر نیستند. این مهم از هر دو جنبه‌ی مخارج اقتصادی و قدرت مصرفی مورد توجه قرار می‌گیرند. به عنوان مثال حسگرهای دما و روشنایی در خانه‌ها، تجهیزات امنیتی مانند دوربین‌های ریز مدار بسته، کلیدهای الکتریکی روی دیوارها همگی تجهیزات کم‌هزینه، کم‌مصرف، بسیار ساده و با ضریب کاری پایین هستند. زیگ بی (به معنی لغوی خط زنبوری) کانال‌های ارتباطی با سرعت انتقال 20، 40، 100 و 250 کیلو بیت بر ثانیه را بنابر بسامد کانال تعریف می‌کند. به‌عنوان مثال یک کانال با بسامد 915MHz در آمریکا سرعت 40Kbps و در اروپا با بسامد 868MHz سرعت 20 Kbps را پشتیبانی می‌کند.

در مجموعه‌ی پروتکلی زیگ‌بی گره‌ها در یک شبکه‌ی Zigbee در قالب دو معماری تعریف می‌شوند:

تجهیزات اصطلاحاً بهره‌مند از **عملکرد کاهش یافته** که به‌صورت بردگان (Slaves) ذیل کنترل یک وسیله‌ی **با عملکرد کامل** همانند فرمان‌بران بلوتوث که قبلاً در همین فصل به آنها اشاره رفته است، عمل می‌کنند. یک وسیله‌ی بهره‌مند از عملکرد کامل به‌صورت یک فرمانده در بلوتوث عمل می‌کند و کنترل چندین وسیله‌ی فرمان‌بر را برعهده می‌گیرد. چندین وسیله‌ی با عملکرد کامل در قالب یک شبکه‌ی تنیده شده امکان پیکربندی و پیاده‌سازی را دارند. در چنین حالتی تجهیزات بهره‌مند از عملکرد کامل، قاب‌های داده را بین یکدیگر مسیردهی می‌کنند. پروتکل زیگ‌بی با اکثریت پروتکل‌های لایه‌ی پیوند، کاربرد اشتراک‌پذیر دارد به‌عنوان مثال قاب‌های بیکن و پیام‌های تأئید لایه‌ی پیوند مانند پروتکل 802.11 و نیز پروتکل‌های دستیابی تصادفی حسی حامل (CSRA) مانند 802.11 و اترنت از جمله‌ی این موارد هستند.

شبکه‌های Zigbee را می‌توان در قالب‌های متفاوت بی‌شماری پیاده‌سازی کرد. به‌عنوان نمونه حالتی را درنظر بگیرید که یک ابزار با عملکرد کامل چندین وسیله‌ی دارای خاصیت عملکرد کاهش یافته را با بهره‌گیری از قاب‌های بیکن به‌صورت حفره‌های زمان‌بندی شده کنترل می‌کند. شکل ۶-۳۳ حالتی را نشان می‌دهد که شبکه‌ی Zigbee زمان را به قاب‌های بزرگ تکرارشونده‌ای که هر کدام با یک قاب بیکن آغاز می‌شوند تقسیم می‌کند. هر قاب بیکن، قاب بزرگ اشاره شده را به یک دوره‌ی زمانی فعال که در فاصله‌ی آن وسیله امکان انتقال داده را دارد و نیز یک فاصله‌ی زمانی خنثی تقسیم می‌کند در طول جریان قاب همه‌ی تجهیزات حتی کنترل‌کننده اقدام به صرفه‌جویی در مصرف قدرت می‌کند و به حالت خواب می‌روند. در محدوده‌ی فاصله‌ی زمانی فعال، 16 حفره‌ی زمانی وجود دارد که پاره‌ای از آنها در قالب دستیابی تصادفی CSMA/CA مورد بهره‌برداری قرار می‌گیرند و برخی دیگر به‌وسیله‌ای خاص از سوی کنترل‌کننده اختصاص می‌یابند. در این رویکرد دسترسی ضمانت‌شده‌ای به کانال برای همه‌ی تجهیزات میسر می‌شود.



شکل ۶-۳۳ ♦ ساختار اَبَر قاب زیگ‌بی 802.14.4

- **هسته‌ی شبکه‌ی 3G**

هسته‌ی شبکه‌ی سلولی داده‌ی 3G از اتصال شبکه‌های دسترسی رادیویی به اینترنت عمومی تشکیل می‌شود. هسته‌ی شبکه با عناصر موجود در شبکه‌های سلولی صدا و به‌ویژه با MSCها که قبلاً به آنها اشاره رفته است ترکیب می‌شود. با توجه به گستردگی زیرساخت‌های موجود سلولی و سودآوری اقتصادی قابل توجه آنها در شبکه‌های تلفنی همراه رویکرد طراحان خدمات داده‌ی 3G کاملاً آشکار است: **عدم تغییر در ساختار شبکه‌های سلولی صدا یعنی هسته‌ی GSM و افزودن کارکردها و توانمندی داده‌ی سلولی به شبکه‌ی سلولی موجود.** رویکرد جایگزین می‌تواند یکپارچه‌سازی مستقیم خدمات جدیدداده به درون هسته‌ی شبکه‌ی سلولی صوتی باشد که سبب افزایش چالش‌های اشاره شده در ارتباط با یکپارچه‌سازی دو فناوری IPV4 و IPV6 خواهد شد. در هسته‌ی یک شبکه‌ی

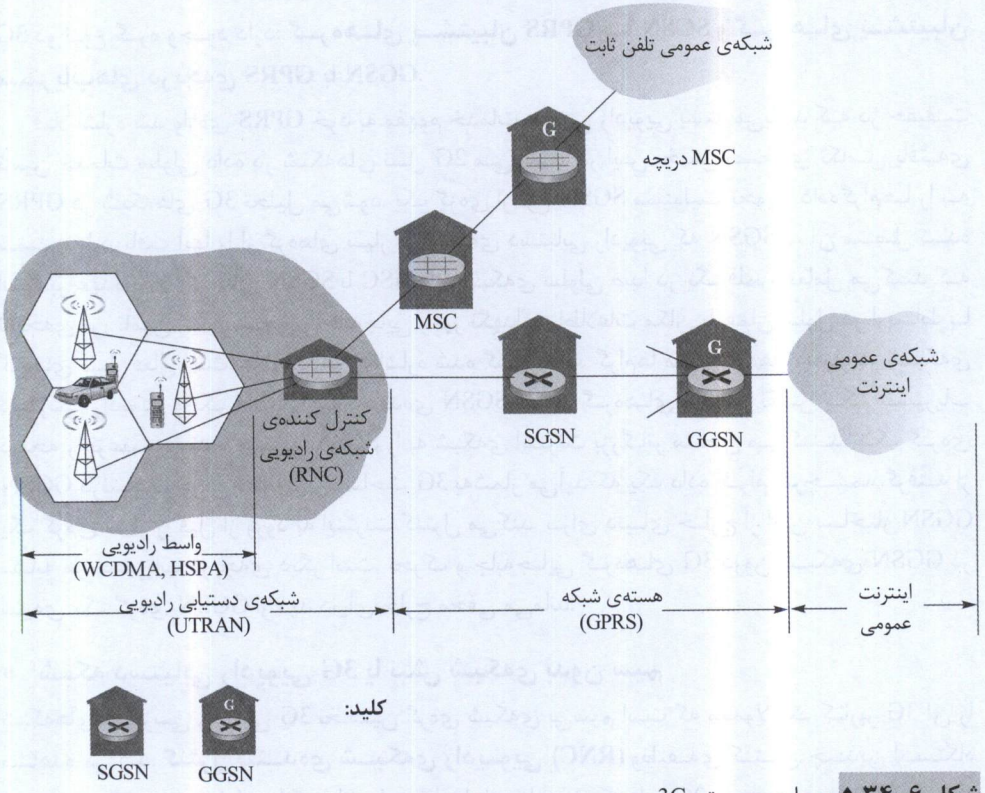
3G دو نوع گره وجود دارد: **گره‌های پشتیبان GPRS یا SGSN و گره‌های پشتیبان مسیریاب‌های دریچه‌ی GPRS یا GGSN.**

قبلاً اشاره شد واژه‌ی GPRS خود به مفهوم خدمات عمومی رادیویی بسته می‌باشد که در حقیقت تأمین خدمات سلولی داده در شبکه‌های نسل 2G می‌باشد. در این بخش نسخه‌ی تکامل یافته‌ی GPRS در شبکه‌های 3G تحلیل می‌شود. یک گره‌ی از نوع SGSN مسئولیت تحویل داده‌گرام‌ها را به سمت و یا دریافت آنها را از گره‌های سیار در شبکه‌ی دستیابی رادیویی که SGSN به آن متصل شده است بر عهده دارد. گره‌های SGSN با MSCهای شبکه‌ی سلولی صدا در یک قلمرو تعامل می‌کنند که نتیجه‌ی آن تأمین اعتبارسنجی و جابه‌جایی و نیز نگهداری اطلاعات مکان یا همان سلول در ارتباط با گره‌های سیار فعال است. علاوه بر موارد اشاره شده گسیل داده گرام‌ها مابین گره‌های سیار در شبکه‌ی دسترسی رادیویی و یک GGSN نیز برعهده‌ی SGSN است. گره‌های GGSN نقش یک مسیریاب دریچه را برعهده دارند و چندین SGSN را به شبکه‌ی اینترنت بزرگ‌تر متصل می‌کنند. یک گره‌ی GGSN درنتیجه آخرین قطعه‌ی زیرساخت 3G به‌شمار می‌آید که یک داده گرام سرچشمه گرفته از یک گره‌ی سیار را قبل از ورود به اینترنت کنترل می‌کند. برای دنیای خارج از این ساختار GGSN مشابه هر مسیریاب دریچه‌ی دیگر است. تحرک و جابه‌جایی گره‌های 3G درون شبکه‌ی GGSN در سایه‌ی یک گره‌ی GGSN از دید دنیای خارج مخفی می‌ماند.

- **شبکه دستیابی رادیویی 3G یا نبش شبکه‌ی بدون سیم**

شبکه‌های دسترسی رادیویی 3G نخستین گره‌ی شبکه‌ی بی‌سیم است که معمولاً یک کاربر 3G آن را مشاهده می‌کند. **کنترل‌کننده‌ی شبکه‌ی رادیویی (RNC)** وظیفه‌ی کنترل چندین ایستگاه فرستنده و گیرنده‌ی سلولی را که مانند ایستگاه‌های مبنا در شبکه‌های 2G هستند برعهده دارد. معمولاً این کنترل‌کننده‌ها بیشتر با نام **3GUMTS** شناخته می‌شوند. هر پیوند بی‌سیم در یک سلول بین گره‌های سیار و پایانه‌ی فرستنده و گیرنده‌ی مبنا عمل می‌کند این عملکرد عیناً شبیه 2G می‌باشد. معمولاً RNC به هر دو شبکه‌ی سلول راهیابی مداری صدا و اینترنت که راهیابی بسته است، به‌ترتیب از طریق یک MSC و یک SGSN متصل می‌شود. بنابراین اگر چه شبکه‌ی سلولی 3G و خدمات سلولی داده‌ی آن از هسته‌ی شبکه‌های مختلف استفاده می‌کنند اما هردو از شبکه‌ی دستیابی رادیویی مشترکاً بهره می‌گیرند.

تغییر چشمگیر در 3GUMTS در مقایسه با شبکه‌های 2G آن است که به‌جای بهره‌گیری از الگوی FDMA/TDMA در GSMها، سیستم‌های 3GUMTS از یک روش CDMA که اصطلاحاً به آن **زنجیره‌ی مستقیم باند گسترده‌ی CDMA یا WCDMA** گفته می‌شود در درون حفره‌های TDMA بهره می‌گیرد. بدیهی است حفره‌های زمانی TDMA خود از طریق چندین بسامد در دسترس قرا می‌گیرند و این مهم بهره‌گیری مناسبی از هر سه رویکرد اشتراک کانال‌های اختصاص یافته را میسر می‌سازد که در فصل ۵ به آنها اشاره شد. این مورد همچنین مشابه رویکرد شبکه‌های دستیابی کابل‌های فلزی است. تغییرات اظهار شده نیازمند یک شبکه‌ی جدید دسترسی بی‌سیم سلولی 3G است که به موازات شبکه‌ی رادیویی 2G-BSS در شکل ۶-۳۴ بایستی عمل نماید. خدمات داده‌ی مربوط به مشخصات WCDMA اصطلاحاً **دستیابی سریع بسته (HSP)** نام گرفته است. این نوع دستیابی امکان سرعت داده‌ی تا 14Mbps را در سمت دریافت (Downlink) تأمین می‌کند. جزئیات بیشتر پیرامون 3G از طریق سامانه‌ی رسمی 3GPP قابل دسترسی می‌باشد.



شکل ۶-۳۴ ♦ معماری سیستم 3G

۶-۱۸ گام به سوی 4G و تکامل دراز مدت (LTE)

از آنجا که سیستم‌های 3G در سراسر گیتی موفق به گسترش تأثیر خود شده‌اند آیا می‌توان انتظار داشت که نسل 4G از این قافله عقب بماند؟ پاسخ قطعاً خیر است. طرح‌های آزمایشی اولیه و بهره‌برداری مقدماتی از سیستم‌های 4G هم‌اکنون در حال پی‌گیری است. استاندارد **تکامل درازمدت 4G** که معروف به **LTE** می‌باشد از سوی کار گروه 3GPP مطرح شده است و از دو نوآوری مهم نسبت به سیستم‌های 3G برخوردار است:

۱- هسته‌ی تکامل یافته بسته (EPC)[1]

استاندارد EPC یک شبکه‌ی اصلی تمام IP است که شبکه‌ی سلولی صوتی راهیابی مدار را با شبکه‌ی سلولی داده‌ی راهیابی بسته براساس شکل ۶-۳۵ یکپارچه می‌کند. پروتکل EPC یک شبکه‌ی تمام IP به‌شمار می‌آید زیرا اطلاعات صوتی، تصویری و متنی (داده) را در قالب داده‌گرام‌های IP ارسال و دریافت می‌کند. از آنجا که الگوی خدماتی بهترین تلاش IP که در فصل‌های ۴ و ۷ مورد مطالعه قرار گرفته است نمی‌تواند مطلوب نیازهای پیچیده و کارآمد ترافیک VOIP باشد درنتیجه مسئولیت کلیدی EPC مدیریت منابع شبکه به‌منظور تأمین خدمات با کیفیت بالا می‌باشد. علاوه بر این مهم EPC تمایز آشکاری را بین موارد کنترل شبکه و محتوای داده‌ها اعمال می‌کند. زیرا بسیاری از موضوعات که

1- Evolved packet core

مربوط به تحرک در شبکه‌های سیار می‌شود به قلمرو عملیاتی کنترل شبکه واگذار می‌شود. پروتکل EPC امکان تنوع در شبکه‌های دستیابی رادیویی مانند پیوند شبکه‌های دسترسی رادیویی 2G و 3G را به هسته‌ی شبکه فراهم می‌کند. جزئیات عمیق پیرامون EPC را می‌توان از دو منبع:[2009 Motorla2007, Alcatel-lucent] مطالعه کرد.

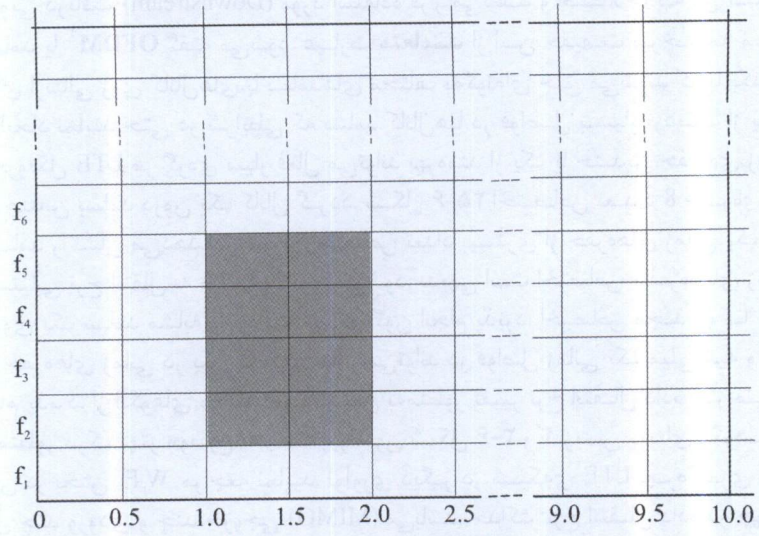
۲- شبکه‌ی دسترسی رادیویی LTE

پروتکل LTE ترکیبی از FDM (تسهیم تقسیم بسامد) و TDM (تسهیم تقسیم زمان) را در سمت کانال رادیویی دریافت (Downstream) مورد استفاده قرار می‌دهد و اصطلاحاً به آن تسهیم تقسیم بسامد متعامد یا **OFDM**[1] گفته می‌شود. عبارت **متعامد** از این حقیقت سرچشمه می‌گیرد که سیگنال‌های ارسالی روی کانال‌های با بسامدهای مختلف به‌گونه‌ای خلق می‌شوند که با یکدیگر کمترین تداخل را ایجاد نمایند حتی در شرایطی که بسامد کانال‌ها در فواصل بسیار نزدیک از یکدیگر قرار دارند. در پروتکل LTE هر گره‌ی سیار فعال می‌تواند بهره‌مند از یک یا چندین حفره‌ی زمانی 0.5ms در یک یا چندین بسامد درون یک کانال گردد. شکل ۶-۳۵ اختصاص تعداد 8 حفره‌ی زمانی روی تعداد 4 بسامد را نشان می‌دهد. در صورت اختصاص تعداد بیشتری از حفره‌های زمانی یک گره‌ی سیار قادر به دستیابی نرخ انتقال به مراتب بالاتری می‌شود. بدیهی است اختصاص حفره‌های زمانی بیشتر می‌تواند روی یک بسامد مشابه یا بسامدهای گوناگون انجام پذیرد. اختصاص مجدد و یا تغییر مکان اختصاص حفره‌های زمانی در بین گره‌های سیار می‌تواند در فواصل زمانی یک میلی‌ثانیه و به هر تعداد مرتبه انجام پذیرد. از الگوهای مختلف مدولاسیون به‌منظور تغییر نرخ انتقال داده نیز می‌توان بهره گرفت. به‌منظور درک بهتر موضوع به بحث پیرامون شکل ۶-۳ و گزینش پویای الگوهای تغییر یا مدولاسیون در بخش W_iF_i مراجعه نمایید. نوآوری دیگر در شبکه‌ی LTE بهره‌گیری از آنتن‌های پیشرفته‌ی چند ورودی و چند خروجی (MIMO) می‌باشد. حداکثر نرخ انتقال داده در جهت دریافت یک کاربر LTE برابر 100Mbps و در سمت ارسال 50Mbps در شرایطی است که طیف شبکه‌ی بی‌سیم استفاده شده پهنایی معادل 20MHZ را شامل گردد. اجباری در اختصاص الگوی ویژه‌ای از حفره‌های زمانی به گره‌های سیار از سوی استاندارد LTE مشخص نشده است و در عوض تصمیم‌گیری در مورد این مهم که کدامیک از گره‌های سیار اجازه‌ی انتقال در یک حفره‌ی زمانی مفروض را در یک بسامد خاص دارند برعهده‌ی الگوریتم‌های زمانبندی گذارده شده است که از سوی فروشندگان تجهیزات LTE و یا راهبران شبکه (اپراتورهای شبکه) تأمین می‌شوند. یکی از الگوریتم‌های مهم این عرصه الگوریتم زمانبندی موقعیت مناسب است که از سوی [Kulkarni 2005] پیشنهاد شده است این الگوریتم وظیفه‌ی نگاشت پروتکل‌های لایه‌ی فیزیکی را، به شرایط کانال بین فرستنده و گیرنده برعهده دارد و علاوه بر آن مسئولیت گزینش گیرنده‌هایی را نیز برعهده دارد که ارسال بسته‌ها به سوی آنها بر مبنای وضعیت کانال و نیز تأمین بهترین شرایط استفاده از رسانای بی‌سیم از سوی کنترل‌کننده شبکه‌ی رادیویی محقق می‌شود. همچنین اولویت کاربران و سطوح قراردادی خدمات آنها از قبیل نقره‌ای، طلایی یا پلاتینی در ارتباط با زمانبندی نرخ انتقال بسته‌ها در شرایط دریافت را از طریق همین الگوریتم می‌توان کنترل و برنامه‌ریزی نمود.

علاوه بر توانمندی‌های توصیف شده در بالا پروتکل‌های LTE پیشرفته‌تر امکان تعریف پهنا باندهای صد مگابیت بر ثانیه را از طریق اختصاص کانال‌های انباشتی به یک گره‌ی سیار بی‌سیم در شرایط دریافت اطلاعات میسر می‌سازند.

1- Orthogonal FDM

فناوری دیگر در ارتباط با شبکه‌های بی‌سیم 4G عبارت است از WiMAX به مفهوم عملیات ترکیب جهانی برای دستیابی مایکرویو. این پروتکل خانواده‌ای از استانداردهای [IEEE 802.16] می‌باشد که تفاوت عمده‌ای با LTE دارد. پاسخ به این پرسش که کدامیک از پروتکل‌های LTE یا WiMAX به عنوان فناوری 4G گزینش می‌شوند حاکی از این واقعیت است که تا بهار سال 2012 ظاهراً LTE از رقیب خود WiMAX پیشتاز بوده است. به‌منظور آشنایی با جزئیات بیشتر در زمینه‌ی WiMAX به سامانه‌ی وب همین اثر مراجعه نمایید.



شکل ۶-۳۵ ♦ تعداد 20 حفره‌ی زمانی 0.2ms که در قالب قاب‌های 10ms در هر بسامد سازماندهی شده است. در بخش سایه زده‌ی شکل ۶-۳۵ یک اختصاص 8 حفره‌ی زمانی نشان داده شده است.

تمرین‌های فصل ششم

۱- چهار نوع شبکه‌ی بی‌سیم عنوان شده در بخش ۶-۱ فصل کدامند؟ کدامیک از آن‌ها را شما مورد استفاده قرار داده‌اید؟

۲- هنگامی که گفته می‌شود یک شبکه‌ی بی‌سیم در روش زیرساخت عمل می‌کند، منظور چیست؟ اگر شبکه‌ای در حالت زیرساخت نباشد چه الگوهای عملیاتی دیگری را می‌تواند اختیار کند؟ تفاوت بین آن‌ها کدامند؟

۳- هرچه یک گره‌ی سیار از پایانه‌ی مبنا فاصله می‌گیرد پایانه‌ی مبنا دو عملیات انجام می‌دهد تا اطمینان از کاهش احتمال از دست رفتن قاب‌های انتقال یافته پیدا کند، این دو عملیات کدامند؟

۴- چه تفاوت‌هایی را بین انواع اختلال‌های ذیل در کانال‌های بی‌سیم می‌شناسید؟ توضیح دهید: فقدان مسیر، انتشار چند مسیری، تداخل از منابع یکدیگر.

۵- آیا دو پروتکل اترنت و 802.11 از یک ساختار مشترک قاب داده بهره می‌گیرند؟

۶- نقش قاب‌های ردیابی (بیکن) در پروتکل 802.11 چیست؟
۷- به چه علت از پیام‌های ACK در 802.11 استفاده می‌شود؟ چرا در اترنت کابلی کاربرد ندارد؟
۸- توضیح دهید آستانه‌ی RTS چگونه عمل می‌کند؟
۹- بخش ۶-۷ به توصیف تحرک در 802.11 می پردازد که در آن یک پایانه‌ی بی‌سیم بین دو BSS در همان زیرشبکه جابه‌جا می‌شود. هنگامی که نقاط دست‌یابی (AP) از طریق یک راه‌یاب به یکدیگر متصل شده باشند، یک AP ممکن است نیازمند ارسال یک قاب با یک آدرس MAC مجعول برای الزام راه‌یاب به ارسال مطلوب قاب باشد، علت آن چیست؟
۱۰- در زیربخش ۶-۵ با دو استاندارد مهم 3G آشنا شدیم. این دو عبارت بودند از UMTS و CDMA 2000. هر یک از این دو با کدام یک از دو استاندارد 2G و 2.5G وابستگی داشتند؟
۱۱- مفهوم زمان‌بندی فرصت طلبانه در WiMAX چیست؟
۱۲- تفاوت‌های مابین یک وسیله‌ی فرمانده در شبکه‌ی بلوتوث و یک پایانه‌ی مبنا در یک شبکه‌ی 802.11 چیست؟
۱۳- فرض کنید فریم‌های IEEE 802.11 RTS و IEEE 802.11 CTS از لحاظ طول برابر قاب‌های استاندارد DATA و ACK باشند. آیا می‌توان ویژگی قابل ملاحظه‌ای را در استفاده از RTS و CTS برشمرد؟ علت را توضیح دهید.
۱۴- آیا در شبکه‌ی WiMAX یک پایانه‌ی مبنا بایستی به همه‌ی گره‌های شبکه با سرعت یکسان انتقال اطلاعات نماید؟
۱۵- تفاوت بین آدرس‌های دایمی و COA کدام است؟ آدرس COA از سوی چه عاملی اختصاص می‌یابد؟
۱۶- اگر یک گره حامل یک ارتباط بی‌سیم به اینترنت باشد. آیا آن گره بایستی سیار باشد؟ علت را توضیح دهید.
۱۷- در پرسش ۱۶ اگر یک کاربر سیار در اطراف محل سکونت خود حرکت نماید و همیشه از طریق یک AP به اینترنت دسترسی داشته باشد آیا این کاربر از نقطه‌نظر شبکه سیار است؟ چرا؟
۱۸- یک ارتباط TCP را از طریق یک IP سیار در نظر بگیرید. آیا مرحله‌ی ارتباط TCP بین عامل گزارشگر و میزبان سیار از درون شبکه‌ی خانه سیار عبور می‌کند یا خیر؟ آیا مرحله‌ی انتقال داده مستقیماً بین گزارشگر و میزبان سیار بدون عبور از شبکه‌ی خانه صورت می پذیرد یا خیر؟
۱۹- نقش MSC مرجع در شبکه‌های GSM چیست؟
۲۰- اهداف HLR و VLR در شبکه‌های GSM کدام است؟ چه عناصری از IP سیار مشابه HLR و VLR می‌باشند؟
۲۱- سه رویکردی که مانع از تضعیف کارایی پیوندهای بی‌سیم در یک ارتباط TCP انتها به انتها می‌باشد کدامند؟ توضیح دهید.
۲۲- فرض کنید گیرنده‌ی شکل ۶-۶ مایل به دریافت داده‌ی ارسالی از سوی فرستنده 2 می‌باشد.

از طریق محاسبه نشان دهید گیرنده بایستی قادر به بازیابی داده فرستنده‌ی 2 از انباشت سیگنال کانال با بهره‌گیری از کد فرستنده‌ی 2 باشد.

۲۳- فرستنده‌ی CDMA در مثال ۶-۵ را در نظر بگیرید. برای دو بیت داده‌ی نشان داده شده خروجی فرستنده در صورتی که کد CDMA به صورت (1,-1,1,-1,1,-1,1,-1) باشد چیست؟

۲۴- برای دو فرستنده و دو گیرنده که از دو کد CDMA شامل 1 و 1- استفاده می‌کنند و اجازه نمی‌دهند دو گیرنده به جداسازی بیت‌های ارسالی اصلی از دو فرستنده‌ی CDMA بپردازند، یک مثال بیان کنید.

۲۵- فرستنده‌ی 2 در شکل ۶-۶ را در نظر بگیرید خروجی فرستنده به کانال قبل از افزایش آن به سیگنال خارج شده از فرستنده‌ی 1 یعنی $\sum_{i,m}^{2}$ چیست؟

۲۶- دوISP را در نظر بگیرید که قادر به تأمین دست‌یابی WiFi در یک مکان می‌باشند. اگر هر گره‌ی ISP نقطه‌ی دست‌یابی خویش را به همراه توده آدرس IP آن در اختیار داشته باشد آنگاه:

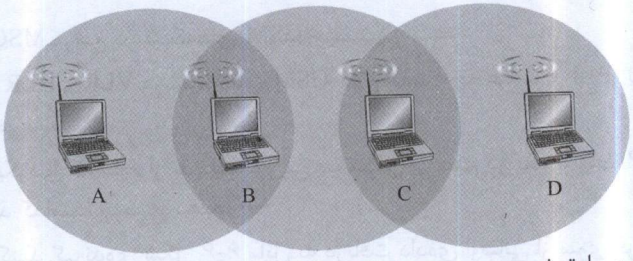
الف) اگر به صورت تصادفی هر ISP پیکربندی AP را بر اساس کانال 11 انجام داده باشد آیا پروتکل 802.11 به شکل کامل دچار اختلال خواهد شد؟ توضیح دهید چه اتفاقی روی می‌دهد هنگامی که دو پایانه هر کدام با وابستگی به یک ISP مختلف تلاش در انتقال همزمان دارند؟

ب) اگر فرض شود یک APبر روی کانال 1 و دیگری بر روی کانال 11 عمل نماید پاسخ به بخش الف چگونه تغییر می‌نماید؟

۲۷- فرض کنید یک پایانه‌ی 802.11b پیکربندی شده است که همیشه به ذخیره‌ی کانال با زنجیره‌ی RTS/CTS بپردازد. اگر فرض شود این پایانه ناگهانی مایل به ارسال 1000 بایت داده باشد و همه‌ی پایانه‌های دیگر در این زمان آزاد باشند. به عنوان تابعی از SIFS و DIFS و بدون در نظر گرفتن تاخیر انتشار و با فرض عدم خطای بیت، زمان لازم جهت انتقال قاب و دریافت پیام تائید آن را محاسبه نمایید.

۲۸- در گام ۴ پروتکل CDMA/CA یک پایانه با موفقیت قابی را ارسال می‌کند و سپس پروتکل CDMA/CA را برای قاب دوم از گام ۲ به جای گام ۱ آغاز می‌کند. چه استدلالی ممکن است طراحان CSMA/CA در ذهن خود داشته‌اند که اجازه داده‌اند یک پایانه بی‌درنگ قادر به گسیل قاب دوم خود نباشد حتی اگر حس نماید کانال آزاد است؟

۲۹- قاب فریم بلوتوث در پروتکل 802.15.1 را تشریح کنید. آیا نکته‌ای در قاب فریم وجود دارد که تعداد گره‌های فعال را در یک شبکه‌ی 802.15.1 فقط به 8 گره محدود نماید؟ توضیح دهید. از شکل ۶-۳۶ کمک بگیرید.



شکل ۶-۳۶♦ سناریوی بلوتوث

۳۰- دو گره‌ی سیار در یک شبکه‌ی خارجی حامل یک عامل بیگانه را در نظر بگیرید. آیا این امکان وجود دارد که دو گره‌ی سیار از یک آدرس COA مشابه در شبکه‌ی IP سیار بهره گیرند؟ توضیح دهید.

۳۱- در بحث به روزرسانی HLR از طریق VLR نقاط قوت و ضعف حضور MRSN در برابر آدرسی از VLR به HLR چیست؟

۳۲- در شبکه IP سیار تاثیر تحرک بر تاخیرهای انتها به انتهای داده‌گرام‌ها بین مبدا و مقصد چیست؟

۳۳- مثال زنجیره‌ای مورد بحث در انتهای زیر بخش ۶-۱۵ را در نظر بگیرید. فرض کنید یک کاربر سیار شبکه‌های خارجی A و B و C را ملاقات می‌نماید و نیز عامل گزارشگر ارتباطی را با کاربر سیار هنگامی که وی در شبکه‌ی خارجی A مقیم است برقرار می‌سازد. فهرستی از پیام‌های بین عوامل خارجی و عوامل بیگانه و خانه را به هنگام جابه‌جایی کاربر سیار از شبکه‌ی A به B و سپس C فهرست نمایید.

۳۴- در پرسش ۳۳ اگر فرض شود عملیات زنجیره شدن روی ندهد و گزارشگر و عامل خانه انحصاراً بایستی از تغییرات COA در کاربر سیار آگاه شوند فهرست پیام‌های مورد نیاز جهت مبادله چه خواهد بود؟

۳۵- فرض کنید گزارشگر شکل ۶-۲۱ سیار باشد. زیرساخت اضافه‌تر لایه‌ی شبکه را بری حالتی رسم کنید که داده‌گرام بایستی از کاربر سیار اصلی به گزارشگر سیار فعلی مسیردهی شود. ساختار داده‌گرام بین کاربر اصلی و گزارشگر سیار فعلی را بر اساس شکل ۶-۲۲ نشان دهید.

۳۶- به عنوان یک کاربر IEEE 802.11 چه نوع اشکالاتی را تاکنون ملاحظه کرده‌اید؟ چگونه 802.11 ار می‌توان طراحی نمود که بر این دشواری‌ها چیره شود؟

۳۷- آیا خدمات بی‌سیم 3G در محدوده‌ی شما وجود دارد؟ قیمت‌گذاری آن چیست؟ چه کاربردهایی را پشتیبانی می‌کند؟



فصل ۷

شبکه‌های چندرسانه‌ای

در حال حاضر مردمان در چهارگوشه‌ی گیتی تصاویر ویدیویی و برنامه‌های تلویزیونی زنده را با بهره‌گیری از اینترنت تماشا می‌کنند. شرکت‌های توزیع تصاویر و نمایش‌های اینترنتی و نیز رسانه‌های تصویری مانند Netfix و Hulu در آمریکای شمالی و نیز Youku و Kankan در چین شهرت و آوازه‌ی وسیعی را برای خود کسب کرده‌اند. مردمان نه‌تنها ویدوهای اینترنتی را تماشا می‌کنند بلکه با بهره‌گیری از سایت‌هایی مانند یوتیوب محتوای تولید شده از جانب خود را نیز بارگذاری یا بارگیری می‌کنند و به عبارتی خود تبدیل به تولیدکنندگان اینترنتی ویدیو و نیز مصرف کننده‌ی آنها شده‌اند. علاوه بر آن کاربردهای شبکه مانند اسکایپ، GoogleTalk و QQ که سامانه‌ی اخیر به شدت مورد علاقه‌ی چینی‌ها می‌باشد این امکان را برای مردمان فراهم آورده‌اند تا علاوه بر ارتباط‌های تلفنی متداول از طریق اینترنت این‌گونه ارتباط‌ها را به ابعاد تصویری و کنفرانس‌های چندنفره نیز ارتقا دهند. در حقیقت با اطمینان می‌توان پیش‌بینی کرد که تا پایان دهه‌ی جاری (2020) تقریباً تمامی محتوای تصویری و مکالمات صوتی در یک قالب انتها به انتها از طریق شبکه‌ی اینترنت انجام خواهد گرفت. در بیشتر موارد تجهیزات بی‌سیم از طریق شبکه‌هایی مانند 4G و WiFi به اینترنت متصل خواهند شد. موضوعات جدید پیرامون فناوری موارد اشاره شده در بالا در ویراست ششم این اثر شامل عناوین بخش‌های ۷-۱۷ به بعد خواهند شد.

موضوعات این فصل مانند گذشته با معرفی واژگان ویژه‌ی موضوع آغاز و به دنبال آن تقسیم‌بندی کاربردهای چند رسانه‌ای به دو گروه اصطلاحاً **جریان صوتی و تصویری ذخیره‌سازی شده و پخش زنده‌ی صدا و تصویر** و به عبارت دیگر **تعامل بی‌درنگ صوتی و تصویری** انجام می‌گیرد. در ادامه‌ی مطالب فصل نیازهای خدماتی هریک از این گروه‌ها معرفی می‌شوند. پروتکل‌های گوناگون چند رسانه‌ای در کاربردهای امروزی اینترنت و نیز راهکارهای تمایز انواع ترافیک قابل تحمل در این بخش از شبکه‌های کامپیوتری از جمله موارد تکمیلی این فصل را تشکیل می‌دهند. کارایی کاربردها به هنگام

دریافت محتوی اطلاعاتی آن‌ها در قالب مسایل چندرسانه‌ای مانند بهره‌گیری از شبکه راهیابی مدار تلفنی و یا شبکه‌های مبتنی بر بسته‌های تلفنی IP و پروتکل‌های وابسته به آن‌ها از جمله‌ی موارد پایانی این فصل را تشکیل می‌دهد.

۷-۱ کاربردهای چندرسانه‌ای شبکه

در بحث مربوط به نیازهای خدماتی کاربردها در فصل ۲ به شناسایی تعدادی از این نیازهای محوری پرداختیم. دو گروه مهم از این نیازها تحمل‌پذیری در برابر اتلاف داده‌ها و زمان‌بندی ارسال داده‌ها بودند که برای کاربردهای چندرسانه‌ای نیز از اهمیت ویژه‌ای برخوردار می‌باشند. عامل زمان‌بندی برای کاربردهای چندرسانه‌ای از این جهت مهم می‌باشد که این گونه کاربردها بسیار در برابر **تاخیر حساس**[1] می‌باشند. در بسیاری از کاربردهای چندرسانه‌ای بسته‌های داده که تحت تأثیر تأخیر فرستنده و گیرنده برای یک مدت زمان بیش از چند صد میلی ثانیه قرار گیرند از نقطه نظر گیرنده بی‌فایده به‌شمار می‌آیند. از طرف دیگر کاربردهای چند رسانه‌ای در اغلب موارد در برابر فقدان بسته‌ها دارای چند ویژگی **تحمل‌پذیری فقدان**[2] می‌باشند. از دست رفتن اتفاقی تعداد اندکی بسته باعث وقفه‌های تقریباً بی‌اثر در پخش صدا و تصویر می‌شود. در اغلب موارد این آسیب‌ها را می‌توان به گونه‌ی نسبی و یا کامل پنهان نمود. این ویژگی‌های حساس در برابر تأخیر و مقاوم در برابر اتلاف داده تفاوت آشکاری را در برابر آن دسته از کاربردهای انعطاف‌پذیر مانند وب، FTP و ایمیل دارند. برای کاربردهای انعطاف‌پذیر تأخیرهای طولانی ناگوار اما آسیب‌رسان نمی‌باشند برعکس یکپارچگی و تمامیت داده‌ی انتقال یافته اهمیت حیاتی چشمگیر دارد.

- **نمونه‌هایی از کاربردهای چندرسانه‌ای**

شبکه‌ی اینترنت قادر به پشتیبانی انواع بی‌شماری از کاربردهای چندرسانه‌ای موجود می‌باشد. در این زیربخش سه گروه وسیع از این کاربردهای چندرسانه‌ای بررسی و تحلیل می‌شوند. این سه گروه عبارتند از:

- **پخش ذخیره‌سازی شده‌ی فایل‌های صوتی و تصویری.**
- **پخش زنده‌ی سیال فایل‌های صوتی و تصویری.**
- **تعامل بی‌درنگ صوتی و تصویری.**

در این فصل موضوعات مربوط به کاربردهای دانلود و سپس پخش مانند دریافت یک فایل کامل MP3 از طریق بهره‌گیری از پروتکل‌های کاربردی مانند P2P قبل از پخش MP3 آن‌ها بررسی نمی‌شود. بدیهی است کاربردهای دانلود و پخش از نوع انعطاف‌پذیر به‌شمار می‌آیند زیرا کاربردهای انتقال فایل بدون نیاز به تأخیر ویژه می‌باشند. در فصل ۲ به‌همراه جزئیات کامل این گونه کاربردها مانند HTTP، FTP و نیز سیستم‌های اشتراک فایل P2P تجزیه و تحلیل شدند.

۱- پخش ذخیره‌سازی شده‌ی فایل‌های صوتی و تصویری

در این گروه از کاربردها برنامه‌های مشتری نسبت به درخواست فایل‌های صوتی و یا تصویری

1- Delay sensitive
2- Loss tolerant

فشرده‌سازی شده برروی سرورها اقدام می‌کنند. هزاران سامانه در حال حاضر امکانات این گونه را برروی اینترنت در اختیار کاربران قرار می‌دهند. ازجمله می‌توان از سامانه‌هایی مانند CNN، ویدیو مایکروسافت، یوتیوپ و غیره نام برد. این گروه از کابردها دارای سه ویژگی متمایز کلیدی می‌باشند:

- **رسانه‌ی ذخیره‌سازی شده**. محتوای یک فایل چندرسانه‌ای از پیش ضبط شده در یک سرور ذخیره‌سازی می‌شود. از آن‌جائیکه رسانه از پیش ضبط شده است، کاربرد سمت برنامه‌ی مشتری ممکن است قادر به **توقف پخش**[1]، **برگشت عقب**[2]، **هدایت سریع به جلو**[3] و یا شاخص‌گذاری درون محتوای فایل چندرسانه‌ای و گزینش بخشی از آن باشد. زمان مورد نیاز از لحظه‌ای که کاربر چنین تقاضایی را اعمال می‌کند تا هنگام انجام عملیات مطلوب تقاضا در سمت برنامه‌ی مشتری بایستی در محدوده‌ی یک یا چندین ثانیه به طول انجامد در غیر این صورت از نقطه نظر کارایی مطلوب نخواهد بود.
- **پخش در قالب جریان لحظه‌ای سیال (Streaming)**. در پخش سیال یک کاربرد صوتی و تصویری برنامه‌ی مشتری پس از دریافت چند ثانیه از فایل چندرسانه‌ای از طریق سرور آغاز به بازپخش آن می‌کند. این به مفهوم آن است که برنامه‌ی مشتری پخش فایل صوتی و یا تصویری را از یک نقطه از فایل آغاز می‌کند در حالی که بخش‌های بعدی فایل را همزمان از سرور دریافت می‌کند. این روش در واژگان چندرسانه‌ای شبکه **پخش سیال**[4] نامیده شده است زیرا با این رویکرد نیازی به دانلود تمامیت اطلاعات نمی‌باشد و پخش و دریافت تقریباً همزمان انجام می‌شود و نتیجه‌ی آن ممانعت از تأخیر طولانی در دریافت و پخش اطلاعات چندرسانه‌ای است. تعداد انبوهی از برنامه‌های مشتری به روش پخش سیال در حال حاضر در اختیار کاربران می‌باشد. از جمله می‌توان از: پخش‌کننده رسانه‌ای مایکروسافت [Microsoft media player 2009]، [Apple Quick Time 2009]، [Real player 2009] و غیره نام برد.
- **پخش مستمر زنده**. در این الگو هنگامی که پخش اطلاعات فایل چندرسانه‌ای آغاز می‌شود، چگونگی پخش آن بایستی برمبنای زمان‌بندی اولیه ذخیره‌سازی آن در سرور صورت پذیرد. به بیان دیگر اطلاعات دریافتی از سرور بایستی در قالب زمانی معین دریافت شده از سرور اخذ گردد تا برنامه‌ی مشتری امکان پخش آن را داشته باشد در غیر این صورت کاربران با تأخیرهای ذخیره‌سازی بسیار طولانی روبه‌رو می‌شوند. اگرچه کاربردهای رسانه‌ی ذخیره‌سازی شده (روش نخست) نیز نیازمند پخش مستمر می‌باشند اما تأثیر تأخیر انتها به انتهای آن‌ها کمتر از کاربردهای زنده محسوس می‌باشد. کاربردهای تعاملی مانند تلفن اینترنتی (IP phone) و همایش تصویری از جمله‌ی این موارد می‌باشند.

1- Pause
2- Rewind
3- Fast forward
4- Streaming

۲- پخش سیال زنده‌ی فایل‌های صوتی و تصویری

این گروه از کاربردها تقریباً مشابه پخش برنامه‌های رسانه‌های جمعی صدا و سیما می‌باشند با این تفاوت که انتقال آن‌ها از طریق شبکه‌ی اینترنت صورت می‌پذیرد. این کاربردها به یک کاربر اجازه‌ی دریافت یک برنامه‌ی **زنده‌ی** پخش رادیویی و یا تلویزیونی از هر نقطه‌ی گیتی که سرچشمه گرفته است را می‌دهند. این کاربردها اغلب به رادیوهای اینترنتی و یا IPTV معروف می‌باشند. در حال حاضر هزاران ایستگاه رادیویی و تعداد قابل توجهی IPTV با بهره‌گیری از شبکه‌ی جهانی اینترنت برنامه‌های زنده‌ی خود را به گوشه و کنار کره‌ی زمین انتقال می‌دهند.

از آن‌جایی که پخش سیال زنده‌ی صدا و تصویر، ذخیره‌سازی نشده است یک برنامه‌ی مشتری قادر به جلو راندن سریع (FF) اطلاعات رسانه نمی‌باشد، اما داده‌های دریافتی اگر به گونه‌ی محلی ذخیره‌سازی شوند، دیگر عملیات تعاملی توقف و بازگشت به عقب امکان‌پذیر می‌باشد. کاربردهای پخش زنده و شبیه آن‌ها اغلب دارای برنامه‌های مشتری فراوانی می‌باشند که برنامه‌های صوتی و یا تصویری مشابهی را دریافت می‌کنند. توزیع فایل صوتی و تصویری زنده به سوی بسیاری از دریافت کننده‌ها می‌تواند با بهره‌گیری از روش های چند پخش IP که در زیربخش ۴-۲۲ توضیح داده شدند به صورت کارآمد، میسر گردد. در حال حاضر توزیع زنده‌ی فایل‌های صوتی و تصویری از طریق لایه‌ی کاربرد چند پخشی با بهره‌گیری از P2P و یا CDN و یا بهره‌گیری از چندین جریان تک پخش مشتری – سرور مجزا انجام می‌شود. مشابه رسانه‌ی ذخیره‌سازی شده، پخش مستمر از جمله نیازهای این الگوی پخش می‌باشد اگرچه تأخیر زمان تأخیر آن کمتر از کاربردهای تعاملی بی‌درنگ می‌باشد. تأخیر تا حدود ده‌ها ثانیه از لحظه‌ی درخواست کاربر جهت تحویل و یا پخش یک انتقال زنده تا هنگامی که پخش آن آغاز می‌گردد، کاملاً تحمل‌پذیر می‌باشد.

- **سیمای اینترنتی (IPTV)**

محتوای برنامه‌های تلویزیونی به صورت سنتی از طریق پیوندهای سطحی مایکروویو، ترکیبی نوری و هم‌محور (HFC) و کانال‌های ماهواره‌ای از لحاظ جغرافیایی ثابت براساس مطالب بخش ۱-۲ توزیع می‌شوند. اما در عصر اینترنت امروزی علاقه‌مندی گسترده‌ای به سیمای پروتکل اینترنت یا همان IPTV در حال ظهور است.

یکی از چالش‌های IPTV ملاحظه‌ی نیاز گسترده به پهنای باند، به ویژه در منبع سرور می‌باشد. به عنوان مثال توزیع یک رویداد ورزشی مهم مانند فوتبال جام جهانی را از طریق یک سرور برروی اینترنت به 100 میلیون کاربر همزمان در ذهن تجسم نمائید. اگر سرعت انتقال ویدیویی 1Mbps درنظر گرفته شود آن‌گاه پهنای باند لازم در خروجی سرور بایستی در حدود 100 Tbps یعنی 100 میلیون برابر 1Mbps باشد! بدیهی است روش توزیع سرور – مشتری سنتی کاملاً بی‌فایده و بی‌اثر می‌باشد. اگر از روش IP چند پخش در سراسر اینترنت به‌صورت گسترده بهره‌گیری شود، آن‌گاه واقعیت IPTV به عمل نزدیک‌تر می‌شود. جایگزین دیگر توزیع ویدیویی از طریق یک شبکه‌ی فراگذر مانند CDN می‌باشد که در زیربخش ۷-۵ در همین فصل بررسی شده است.

جایگزین سوم، بهره‌گیری از توزیع نظیر به نظیر (P2P) است که در آن هر مانند یک کانال تلویزیونی را دریافت می‌دارد و نیز در توزیع مجدد کانال به دیگر نظیرها کمک می‌کند. شاید جذابیت مهم این رویکرد هزینه‌ی پائین توزیع آن می‌باشد. اگر هریک از نظیرها مجموعاً پهنای باند لازم جهت بارگذاری تصاویر را فراهم آورند آن‌گاه پهنای باند نسبتاً اندکی از جانب سرور بایستی تأمین شود، شاید

چندین برابر نرخ انتقال ویدیویی، با چنین هزینه‌ی اندک هر فردی با بهره‌گیری از یک دوربین وب (Web cam) می‌تواند یک برنامه‌ی زنده را بین میلیون‌ها کاربر با هزینه‌ی بسیار ناچیزی توزیع نماید.

در حال حاضر برخی از سیستم‌های این گونه مانند P2P IPTV از این به‌کارگیری با موفقیت استفاده نموده‌اند، یکی از پیشتازان این عرصه یعنی Cool Streaming پشتیبانی بیش از 4000 کاربر همزمان در سال 2002 میلادی را گزارش نموده است. در این اواخر سیستم‌های PPLive و PPStream موفقیت چشم‌گیر بیش‌تری را در نرخ بین 300 kbps و 1Mbps جهت ارائه‌ی خدمات چندرسانه‌ای برای ده‌ها هزار نفر از کاربران فراهم آورده‌اند. در این سیستم‌های شبیه بیت ترنت، نظیرها تشکیل یک شبکه‌ی جایگذار پویا را می‌دهند و به مبادله‌ی قطعه‌های اطلاعاتی با مجاوران جایگزین خود می‌پردازند. رشد IPTV از لحاظ ضریب نفوذ آن در جهان در 5 تا 10 سال آینده توجه پژوهشگران را به خود معطوف نموده است. نکات مهم دیگر این تحول نوع فنآوری می‌باشد که مورد استفاده‌ی کاربران قرار خواهد گرفت. آیا این فناوری CDN یا P2P و یا آن که ترکیبی از این دو می‌باشد؟ آیا درصد چشمگیری از علاقه‌مندان جام جهانی 2014 و المپیک 2016 این بازی‌ها را از طریق IPTV تماشا خواهند نمود یا از طریق رسانه‌های متداول فعلی؟

۳- تعامل بی‌درنگ صدا و تصویر

این گروه از کاربردها بهره‌گیری از ارتباطات صوتی و تصویری بی‌درنگ را در اختیار کاربران قرار می‌دهند. تعامل بی‌درنگ صوتی از طریق اینترنت اصطلاحاً **تلفن اینترنتی** نامیده می‌شود، زیرا از نقطه نظر کاربر بسیار مشابه خدمات تلفنی شبکه راهیابی مدار (PSTN) می‌باشد. خدمات تلفن اینترنتی هر سه نوع تبادل انشعاب خصوصی (PBX)، محلی و راه دور را با هزینه‌ی بسیار پائین امکان‌پذیر می‌سازد. بهره‌گیری از خدمات نوین مانند شناسایی حضور، ارتباط جمعی یا گروهی، تعریف محدودیت ارتباط و غیره که شبکه‌های تلفنی سنتی قادر به پشتیبانی از آن‌ها نمی‌باشند، در اینترنت تلفنی حضور پررنگی را نشان می‌دهند. در حال حاضر ده‌ها برنامه‌ی کاربردی ویژه‌ی تلفن اینترنتی از قبیل اسکایپ، اووو و غیره وجود دارند که کاربران با بهره‌گیری از آن‌ها قادر به ارتباط‌های از نوع PC به تلفن، مکالمه‌های صوتی PC به PC و غیره می‌باشند. تعامل بی‌درنگ تصویری که کنفرانس تصویری نیز گفته می‌شود به کاربران اجازه می‌دهد ارتباط صوتی و تصویری تقریباً همزمان را با یکدیگر برقرار نمایند. از جمله برنامه‌های کاربردی در این زمینه، می‌توان از Net Meeting مایکروسافت، ویدیو اسکایپ و غیره نام برد. بدیهی است در کاربردهای تعاملی صوتی و تصویری بی‌درنگ یک کاربر در هنگام مکالمه و گفتگو می‌تواند در هر لحظه از زمان حرکت‌های فیزیکی مانند نگارش، جابه جا شدن و غیره را نیز اختیار نماید. برای یک مکالمه‌ی تعامل‌گرا در بین چندین شرکت کننده زمان تأخیر قابل قبول از لحظه‌ی آغاز سخن و یا حرکت کاربر تا هنگام دریافت آن از سوی برنامه‌های میزبان در مقیاس کم‌تر از چندصد میلی ثانیه می‌باشد. برای تعامل‌های صوتی میزان قابل قبول بین 150 تا 400 میلی ثانیه است. تأخیر بیش از 400 میلی ثانیه برای مکالمه بسیار ناخوشایند و در اغلب موارد غیرقابل درک از لحاظ مفهوم مکالمه می‌باشد.

۷-۲ موانع فعلی چندرسانه‌ای در شبکه‌ی اینترنت

یادآوری می‌شود پروتکل IP استفاده شده در اینترنت در حال حاضر برمبنای **بهترین تلاش خدماتی** به همه‌ی داده گرام‌های حمل شده است. به بیان دیگر شبکه‌ی اینترنت بهترین تلاش خود را جهت انتقال و جابه جایی هریک از داده گرام‌ها از فرستنده به گیرنده با بهترین سرعت ممکن می‌کند. با این

حال هیچ گونه تدبیری را در رابطه با احتمال تأخیر انتها به انتها برای هریک از بسته‌ها و یا اعمال تغییرات تأخیر بسته در درون یک جریان انتقال اتخاذ نمی‌کند. از آن‌جایی که TCP و UDP براساس پروتکل IP اجرا می‌شوند بنابراین هیچ یک از این پروتکل‌های انتقال ضمانت تأخیر به وجود آمده در اثر کاربردهای اشاره شده را نمی‌کنند. به دلیل غیبت تلاش ویژه برای تحویل بسته‌ها در یک قالب زمان‌بندی ویژه، طراحی موفقیت آمیز کاربردهای چندرسانه‌ای تحت شبکه‌ی اینترنت بسیار دشوار و همراه با چالش‌های فراوان است. با توجه به همه‌ی این محدودیت‌ها پیاده‌سازی و اعمال پروتکل‌های چندرسانه‌ای از طریق اینترنت موفقیت قابل توجهی را تا امروز داشته است. به عنوان مثال پخش صدا و تصویر ذخیره سازی شده در حال حاضر در اینترنت با تأخیری بین 5 تا 10 ثانیه انجام می‌شود. بدیهی است در اوج ترافیک، کارایی ممکن است چندان رضایت‌بخش نباشد به ویژه هنگامی که پیوندهای عمل کننده مانند پیوندهای بین قاره‌ای از لحاظ ترافیک متراکم باشند.

تلفن اینترنتی و تعامل صدا و تصویر بی‌درنگ نیز کاربردهای گسترده‌ای در حال حاضر دارند. به عنوان مثال در هر لحظه از زمان براساس آخرین آمارها بیش از 7میلیون کاربر اسکایپ از خدمات آنلاین این کاربرد بهره می‌گیرند. کاربردهای تعامل بی‌درنگ صدا و تصویر محدودیت‌های ثابت و انعطاف‌ناپذیری را برروی تأخیر و ناپایداری بسته تحمیل می‌کنند. **ناپایداری بسته**[1] تغییراتی از تأخیرهای بسته در درون یک جریان مشابه انتقال بسته‌ها می‌باشد. انتقال بی‌درنگ صدا و تصویر هنگامی عملکرد مطلوب از خود نشان می‌دهد که پهنای باند در دسترس چشمگیر و تأخیر و ناپایداری بسته‌ها حداقل باشد. اما هنگامی که جریان بی‌درنگ بسته‌های صدا و تصویر با یک پیوند نسبتاً متراکم روبه‌رو می‌شود کیفیت دچار وضعیت بسیار نامطلوب می‌شود.

طراحی کاربردهای چندرسانه‌ای قطعاً در صورت تعریف نوعی خدمات اینترنتی درجه یک و یا دو می‌تواند به مراتب ساده گردد. از این نقطه نظر بسته‌های از نوع درجه یک در صف یک مسیریاب از خدمات اولویتی بهره‌مند می‌شوند در حالی که از لحاظ تعداد جمعیتی محدود خواهند شد. بدیهی است چنین خدمات درجه یک، مطلوب کاربردهای حساس در برابر تأخیر است. در حال حاضر در شبکه‌ی اینترنت از یک رویکرد برابرنگر زمان‌بندی بسته در صف‌های روتر استفاده می‌شود. همه‌ی بسته‌ها از نقطه نظر خدماتی برابرند و هیچ یک از آن‌ها حتی بسته‌های صوتی و تصویری حساس در برابر تأخیر بر دیگران اولویت ندارند. در نیمه‌ی آخر این فصل معماری‌هایی جهت برطرف نمودن این محدودیت‌ها پیشنهاد شده است.

بدیهی است در شرایط فعلی تلاش ارایه‌ی بهترین خدمات را بایستی پذیرا بود اگرچه می‌توان با اعمال تدابیری محدودیت‌های اعمال شده را تا اندازه‌ای برطرف نمود، به عنوان مثال با ارسال داده‌های صوتی و تصویری از طریق UDP از محدودیت‌های اعمال شده در TCP مانند کاهش نهفتگی عبور نمود و یا ناپایداری بسته‌ها در اثر تأثیرات شبکه را با اعمال زمان تأخیر پخش درگیرنده‌ها به اندازه 100 میلی ثانیه و یا بیش‌تر پنهان نمود. قالب‌بندی بسته‌ها در فرستنده را می‌توان به گونه‌ای انجام داد که گیرنده از زمان پخش بسته‌ها آگاه گردد. در کاربردهای صدا و تصویر ذخیره سازی شده، داده‌ها را می‌توان در هنگام عملیات پخش و یا هنگامی که ظرفیت ذخیره‌سازی و سرعت در دسترس مشتری مطلوب می‌باشد از قبل واکشی نمود، حتی می‌توان اطلاعات افزونه را به منظور کاهش تأثیر فقدان بسته‌ها ارسال نمود. بررسی همه‌ی این موارد در ادامه انجام خواهد شد.

1- Packet jitter

- **چگونگی تکامل اینترنت به منظور پشتیبانی بهینه‌ی امکانات چندرسانه‌ای**

در حال حاضر گفتمان‌های فراوانی پیرامون چگونگی تکامل اینترنت در جریان است تا آن را مناسب کاربردهای چندرسانه‌ای، با توجه به محدودیت‌های انعطاف ناپذیری از لحاظ زمان‌بندی نماید. در یک سوی این گفتمان برخی از پژوهشگران تغییرات بنیادی را پیشنهاد می‌دهند که بایستی به شبکه‌ی اینترنت اعمال گردد تا کاربردها قادر به گزینش انحصاری پهنای باند انتها به انتها و کارایی ضمانت شده باشند. براین مبنا از دو نوع ضمانت نام برده می‌شود این دو اصطلاحاً **ضمانت سخت و ضمانت نرم**[1] نامیده شده‌اند. یک ضمانت سخت به مفهوم آن است که کاربرد کیفیت خدمات (QoS) مورد تقاضای خود را با اطمینان کامل دریافت می‌دارد. یک ضمانت نرم به مفهوم آن است که کاربرد با احتمال نسبتاً قابل قبولی کیفیت خدمات مورد تقاضای خود را دریافت می‌کند. پژوهشگرانی که طرفدار این گفتمان می‌باشند براین باورند که اگر یک کاربر به عنوان مثال مایل به انجام یک مکالمه‌ی تلفن اینترنتی است آن‌گاه برنامه‌ی کاربردی تلفن اینترنتی کاربر بایستی قادر به ذخیره‌ی انحصاری پهنای باند مورد نیاز خود در هر پیوند امتداد یک مسیر بین دو میزبان A و B باشد. اما اجازه‌ی رزرو پهنای باند از سوی کاربردها و مجبور نمودن شبکه به رعایت این نیازها تغییرات وسیعی را باعث می‌شود. نخست آن که یک پروتکل به نیابت از کاربردها جهت ذخیره‌ی پهنای باند پیوندها در امتداد مسیر از فرستنده‌ها به گیرنده‌ها لازم است. دوم سیاست‌های زمان‌بندی در صف مسیریاب‌ها بایستی به‌گونه‌ای تغییر نماید که رعایت ذخیره‌سازی پهنای باند امکان‌پذیر باشد. با این سیاست گذاری‌های جدید زمان‌بندی همه‌ی بسته‌ها از امکانات برابر بهره‌مند نخواهند شد بلکه آن گروه از آن‌ها که توانایی رزرو دارند امکانات بیش‌تری نیز در اختیار خواهند گرفت. سوم جهت رعایت نیازهای مربوط به عملیات رزرو کاربردها بایستی اطلاعاتی از ترافیک که قصد ارسال آن به شبکه را دارند در اختیار شبکه قرار دهند. در این حالت شبکه بایستی ترافیک هر کاربرد را کنترل نماید تا مطمئن شود که برمبنای آن‌چه توصیف شده است عمل می‌نماید. سرانجام شبکه بایستی دارای امکاناتی جهت اطمینان از پهنای باند کافی و در دسترس جهت پشتیبانی از هر تقاضای ذخیره‌ی پهنای باند جدید باشد. هنگامی که این مکانیزم‌ها با یکدیگر ترکیب شوند نرم افزار جدید و پیچیده‌ای در میزبان‌ها و مسیریاب‌ها به همراه خدمات جدید نیاز می‌باشد. این مکانیزم‌ها در بخش ۷-۱۶ پوشش داده می‌شوند.

در سوی دیگر این گفتمان، گروه دیگری از پژوهشگران براین باورند که لزومی به اعمال تغییرات بنیادی در تلاش به منظور ارایه‌ی بهترین خدمات و پروتکل‌های بنیادی آن نمی‌باشد و به جای آن پیشنهاد یک رویکرد **عدم مداخله** را می‌دهند، این موارد عبارتند از:

۱- هرچه برتعداد تقاضاها افزوده می‌شود، ISP های سطوح پائین و یا بالا شبکه‌های خود را برمبنای نیاز مقیاس گذاری می‌کنند. معمولاً ISPها ظرفیت‌های راهیابی و پهنای باند کافی جهت تأمین رضایت‌بخش تأخیر و فقدان کارایی بسته در درون شبکه‌های خود را فراهم می‌آورند. در نتیجه می‌توانند خدمات بهتری را برای کاربران و مشترکین ISP تامین کنند و از این جهت درآمد بیش‌تری را نیز برای خود کسب می‌کنند. یک ISP ممکن است جهت رسیدن به این هدف پهنای باند و ظرفیت راهیابی خود را بیش از حد نیاز افزایش دهد تا خدمات کافی برای کاربردهای چندرسانه‌ای در مواقع افزایش بیش از حد بار ترافیکی را امکان پذیر نماید. این مهم از طریق پیش‌بینی مناسب ترافیک و پهنای باند ضمانت نرم QoS امکان‌پذیر است.

1- Hard and soft guarantee

۲- **شبکه‌های توزیع محتوا (CDN)**[1] شبیه‌سازی حالت ذخیره سازی شده‌ی صدا و تصویر می‌باشد و این محتوای شبیه سازی شده در لبه‌ها یا نبش اینترنت قرار می‌گیرد. با توجه به این که جریان نسبتاً بالای ترافیک از درون اینترنت قالب محتوای ذخیره‌سازی شده مانند صفحه‌های وب، MP3 و ویدیو را دارد، شبکه‌های CDN می‌توانند بار ترافیک برروی یک ISP را در حجم مؤثری حمل نموده و همانند واسطهای نظیر بین ISP ها عمل نمایند. علاوه بر آن شبکه‌های CDN کاملاً مجزا از تامین کنندگان اطلاعات محتوا عمل می‌کنند. تأمین کنندگان اطلاعات محتوایی به یک شبکه‌ی CDN هزینه پرداخت می‌کنند تا به گونه‌ای سریع و مؤثر محتوا را تحویل آن‌ها نمایند. موارد مربوط به شبکه‌های CDN در بخش ۷-۵ مورد مطالعه قرار می‌گیرند.

۳- برای تأمین نیازهای ترافیک پخش زنده مانند رویدادهای ورزشی که همزمان به میلیون‌ها کاربر گسیل می‌شود، از **شبکه‌های جایگذار چندپخش**[2] می‌توان بهره گرفت. یک شبکه‌ی جایگذار چندپخش شامل میزبان‌های کاربران و احتمالاً سرورهای انحصاری پراکنده در سراسر اینترنت می‌باشد. این میزبان‌ها، سرورها و پیوندهای منطقی بین خود یک شبکه‌ی جایگذار را تشکیل می‌دهند که به صورت چندپخش ترافیک را از مبدأ به میلیون‌ها کاربر مقصد انتقال می‌دهند. این رویکرد عکس IP می‌باشد که مسیریاب‌ها از عملیات چندپخش برای توزیع کنندگان IP استفاده می‌کنند. شبکه‌های جایگذار عملیات چندپخش را در لایه‌ی کاربرد انجام می‌دهند. به عنوان مثال میزبان مبدأ ممکن است جریانی را به سه سرور جایگذار گسیل نماید، هریک از این سه سرور جایگزین ممکن است جریان را به دیگر سرورها و میزبان‌ها گسیل نمایند، ادامه‌ی این فرآیند یک درخت توزیع برفراز شبکه‌ی IP ایجاد می‌کند. در مقایسه با توزیع تک پخش، روش گسیل چندپخش ترافیک زنده‌ی پرطرفدار از درون شبکه‌های جایگذار سبب می‌شود بار ترافیک کلی در اینترنت کاهش قابل توجهی را به همراه داشته باشد.

در بین دو گروه طرفدار عدم مداخله و رزرو پهنای باند، گروه سوم از پژوهشگران قرار می‌گیرند که معروف به گروه طرفدار **خدمات متمایز** می‌باشند. این گروه مایل به اعمال تغییرات نسبتاً اندک در لایه‌های شبکه و انتقال می‌باشند. علاوه برآن طرفدار روش‌های سیاست‌گذاری و قیمت‌گذاری ساده در نبش شبکه یعنی در واسط بین کاربر و ISP آن نیز می‌باشند. هدف اصلی این گروه تعریف حداقل دو گروه ترافیک و اختصاص هریک از آن‌ها به داده گرام‌ها می‌باشد. در نتیجه سطوح خدمات مختلف به داده گرام‌ها براساس اهمیت گروهی آن‌ها در صف‌های مسیریاب و نیز هزینه‌ی پرداختی از سوی کاربران برمبنای نوع درجه‌ی گروهی از بسته‌های وارد شده به شبکه می‌باشد. این گونه خدمات متمایز در بخش ۷-۱۲ تحلیل می‌شوند.

این سه رویکرد متفاوت در برخورد با ترافیک چندرسانه‌ای یعنی: QoSهای گوناگون، خدمات برمبنای بهترین تلاش و QoS ضمانت شده در جدول ۷-۱ خلاصه شده‌اند و در زیربخش‌های ۷-۱۲، ۷-۵ و ۷-۱۶ به ترتیب تحلیل شده‌اند.

1- Content distribution networks

2- Multicast overlay networks

جدول ۷-۱ ♦ مقایسه‌ی سه رویکرد پشتیبان کاربردهای چندرسانه‌ای

رویکرد	یکای اختصاص	ضمانت	بکارگیری در حال حاضر	پیچیدگی	الگوها
خدمات بهترین تلاش	-	هیچ و یا نرم	هرجا و هر نقطه	حداقل	پشتیبانی لایه‌ی کاربرد، CDN و تأمین بیش از حد
QoS متمایز	گروه‌های جریان	هیچ و یا نرم	برخی از مکان‌ها	متوسط	سیاست‌گذاری و زمان‌بندی
QoS ضمانت شده	جریان فردی	نرم، سخت هنگام قبول جریان	اندکی از مکان‌ها	بسیار فراوان	سیگنالینگ، پذیرش مکالمه، زمان‌بندی و غیره

۷-۳ فشرده‌سازی صدا و تصویر

قبل از انتقال اطلاعات صوتی و تصویری از طریق یک شبکه‌ی کامپیوتری، این اطلاعات بایستی ابتدا تبدیل به کدهای دیجیتال شده و سپس فشرده‌سازی گردد. نیاز به دیجیتال نمودن سیگنال‌های صوتی و تصویری کاملاً روشن است زیرا شبکه‌های کامپیوتری برمبنای انتقال بیت عمل می‌کند و بنابراین همه‌ی اطلاعات انتقال یافته بایستی به گونه‌ی زنجیره‌ای از بیت‌ها تعریف شوند. به دلیل آن‌که فایل‌های صوتی و تصویری فشرده‌سازی نشده مصرف کننده‌ی مقدار زیادی از ظرفیت ذخیره سازی و پهنای باند می‌باشند. بنابراین فشرده‌سازی این گونه فایل‌ها اهمیت فراوانی دارد. به عنوان مثال اگر فرض شود یک تصویر که شامل 1024 نقطه‌ی رنگی می‌باشد و هر پیکسل نیز در قالب یک کد 24 بیت که هر بیت آن اختصاص به سه رنگ قرمز، سبز و آبی دارد، تعریف گردد، آن‌گاه بدون بهره‌گیری از امکانات فشرده سازی این تصویر به تنهایی نیازمند 3 مگابایت ظرفیت حافظه خواهد بود. زمان لازم برای ارسال این تصویر از طریق یک پیوند 64 kbps معادل 7دقیقه می‌باشد. در صورتی که اگر همین تصویر با ضریب 10:1 فشرده سازی گردد فضای لازم جهت ذخیره‌سازی آن معادل 300 کیلو بایت و زمان انتقال آن نیز کاهش با ضریب 10 را در همان سرعت 64 kbps به همراه دارد.

موضوع مربوط به فشرده سازی صدا و تصویر نزدیک به نیم قرن است که مورد مطالعه و پژوهش قرار دارد و صدها استاندارد و مقاله در این زمینه پیشنهاد و انتشار یافته است. بسیاری از دانشگاه‌ها دروس اختصاصی ویژه پیرامون این موضوع ارایه می‌دهند. در این اثر معرفی خلاصه‌ای از روش‌های فشرده‌سازی انجام می‌گیرد:

۱- فشرده سازی صدا در اینترنت

یک سیگنال صوتی آنالوگ در حال تغییر معمولاً به صورت ذیل تبدیل به یک سیگنال دیجیتال می‌شود:

- از سیگنال صوتی آنالوگ نخست در یک سرعت تعریف شده‌ای مانند 8000 در هر ثانیه نمونه‌گیری می‌شود. بدیهی است ارزش هر نمونه یک عدد حقیقی تصادفی خواهد بود.

- ارزش هر نمونه سپس به نزدیک‌ترین ارزش کامل آن گرد می‌شود. این عملیات اصطلاحاً **کمیت دهی**[1] نامیده می‌شود. ارزش‌های گرد شده معمولاً در قالب توانی از 2 مانند 8، 16، 256 و غیره تعریف می‌شوند.
- هر ارزش کمیت دهی شده بیان کننده‌ی تعداد ثابتی از بیت می‌باشد. به عنوان مثال اگر تعداد ارزش‌های کمیت دهی شده معادل 256 ارزش باشد، آن‌گاه هر ارزش و بنابراین هر نمونه از طریق **یک بایت** قابل تعریف است. هریک از نمونه‌ها تبدیل به معادل تعداد بیت مربوطه می‌گردد. به عنوان مثال در حالت 256 ارزش کمیت دهی هر کمیت در قالب 8 بیت قابل تعریف می‌باشد. بدیهی است چون ارزش‌ها می‌توانند مثبت و یا منفی باشند طبیعی است تعریف هر نمونه بنابر علامت (+) و یا (-) آن می‌تواند در محدوده‌ی 0 تا 127+ و یا 1- تا 128- قرار گیرد. نمایش علامت‌دار یک نمونه با ارزش 65+ در قالب تعریف تعداد 8 بیت آن به صورت: (01000001) می‌باشد. فرض شده است بیت واقع در سمت چپ نمایش بیت‌ها معرف بیت علامت است.

در این الگو اگر سیگنال آنالوگ صوتی با سرعت 8000 نمونه بر ثانیه در معرض نمونه‌برداری قرار گیرد و اگر هر نمونه نیز تبدیل به یک رشته‌ی 8 بیت مانند مورد اشاره شده در بالا گردد آن‌گاه سیگنال دیجیتال حاصل از این فرآیند سرعتی برابر $8000 \times 8 = 64000\,bps$ را اختیار می‌کند. بدیهی است این سیگنال رمزگذاری شده سپس در محل گیرنده هنگام پخش، رمزگشایی شده و تقریباً تبدیل به سیگنال آنالوگ اولیه می‌شود. بدیهی است یک ارتباط مستقیم بین کیفیت سیگنال رمزگشایی شده و ظرفیت ذخیره سازی و پهنای باند مورد نیاز سیگنال دیجیتال وجود دارد.

اصول اساسی تعریف شده در بالا معروف به **مدولاسیون پالس به کد (PCM)**[2] می‌باشد. رمزگذاری صدا معمولاً در **PCM** با سرعت نمونه برداری 8000 برثانیه و 8 بیت در هر نمونه انجام می‌شود و نتیجه‌ی آن سرعت 64kbps می‌باشد. لوح فشرده (CD) نیز از روش PCM با سرعت نمونه‌گیری 44100 نمونه در ثانیه و 16 بیت برای هر نمونه استفاده می‌کند در نتیجه سرعتی معادل 705.6kbps برای پخش معمولی و 1.411 Mbps برای حالت استریو را تعریف می‌کند.

بدیهی است سرعت دستیابی 1.411 Mbps بسیار فراتر از نیاز فایل‌های صوتی اینترنت می‌باشد همان‌گونه که سرعت 64 kbps نیز برای دستیابی کاربرانی که از مودم شماره‌گیر سنتی بهره می‌گیرند نسبتاً زیاد می‌باشد. به همین دلایل از روش PCM جهت رمزگذاری فایل‌های صوتی مکالمه و یا موسیقایی در اینترنت به ندرت استفاده می‌شود و به جای آن از روش‌های فشرده سازی جهت کاهش سرعت انتقال جریان داده‌ها استفاده می‌شود. معروف‌ترین روش‌های فشرده‌سازی برای مکالمه عبارتند از:

GSM با سرعت 13kbps، **G72q** با سرعت 8 kbps، **G723.3** با دو سرعت 5.3 و 6.4 کیلو بیت بر ثانیه و تعداد قابل توجه دیگری از روش‌های ویژه‌ی خصوصی دیگر و علاوه بر همه‌ی این موارد روش معروف دیگری که از لحاظ کیفیت موسیقایی استریو به CD نزدیک‌تر است، **MPEG1 لایه‌ی سوم** و یا به اختصار MP3 می‌باشند. سرعت فشرده‌سازی رمز کننده‌های MP3 به ترتیب 96، 128 و 160 کیلو بیت بر ثانیه است و تولید تضعیف صدا نیز در آن‌ها بسیار اندک است. هنگامی که یک فایل MP3

1- Quantization
2- Pulse code modulation

رمزگشایی می‌شود، آن فایل تقسیم به قطعه‌هایی می‌شود که هر بخش آن قابلیت پخش مستقل را دارد. این فایل بدون سرپیام، اجازه می‌دهد فایل‌های موسیقایی MP3 در اینترنت مشروط بر سرعت ارتباط سازگار با آن‌ها قابلیت بازپخش را داشته باشند. استاندارد فشرده سازی MP3 نسبتاً پیچیده است و از سه روش کاهش افزونگی، منبع ذخیره‌ی بیت و پوشش **ذهنی انسان از صدا**[1] بهره می‌گیرد.

۲- فشرده‌سازی تصویر در اینترنت

یک فایل ویدئویی زنجیره‌ای از تصاویر است که در سرعت ثابتی به نمایش درمی‌آید، به عنوان مثال در سرعت متعارف 24 و یا 30 تصویر در ثانیه. یک تصویر رمزگذاری شده‌ی دیجیتال که فشرده سازی نشده است شامل آرایه‌ای از نقاط تصویری (پیکسل) می‌باشد که هریک از این نقاط به تعدادی بیت در قالب تعریف روشنایی و رنگ آن تصویر تعریف می‌شوند. دو گونه افزونگی در تصاویر ویدیویی وجود دارد که از هر دوی آن‌ها به منظور اعمال فشرده سازی می‌توان بهره گرفت. افزونگی فضایی (فاصله‌ای)، خاصیتی است در درون هر تصویر که از آن می‌توان به صورت کارآمد جهت فشرده سازی استفاده نمود. به عنوان مثال نقاط رنگ سفید در یک تصویر، بخشی از افزونگی فضایی است که به آسانی آن‌ها را می‌توان فشرده نمود. افزونگی دوم زمانی است، این افزونگی بیان کننده‌ی سرعت و یا زمان تکرار یک تصویر به تصویر بعدی می‌باشد. به عنوان مثال اگر یک تصویر و تصویر بعدی آن مشابه یکدیگر باشند دلیلی وجود ندارد که تصویر دوم مجدداً رمزگذاری شود. اگر هنگام رمزگذاری به نوعی مشخص گردد که تصویر بعدی عیناً مشابه قبلی است، کارآمدی رمزگذاری افزایش می‌یابد.

استانداردهای فشرده سازی MPEG از جمله کاربردی‌ترین روش‌های اعمال شده در فشرده‌سازی تصویر می‌باشند. این استانداردها عبارتند از:

- **MPEG1** برای تصاویر با کیفیت نمایش در لوح فشرده (CD) با سرعت 1.5 Mbps.
- **MPEG2** جهت تصاویر با کیفیت بسیار بالا در لوح فشرده دیجیتال (DVD) با سرعت مابین 3 تا 6 مگابیت در ثانیه.
- **MPEG4** برای فشرده سازی تصویری شیی‌گرا.

استاندارد MPEG اصول بنیادی خود را از مجموعه استانداردهای JPEG گرفته است. این استانداردها علاوه بر بهره‌گیری از افزونگی فضایی JPEG از افزونگی زمانی موجود در تصاویر نیز بهره می‌گیرند. نوع دیگر استانداردهای متداول در شبکه‌ی اینترنت جهت فشرده سازی تصویر **H.261** می‌باشد به این مجموعه روش‌های فشرده سازی رمزگذاری خصوصی مانند Quick Time و Real Networks را نیز بایستی اضافه نمود.

۷-۴ پخش فایل‌های صدا و تصویر ذخیره سازی شده

در سال‌های اخیر از جمله کاربردهای پرطرفدار پخش تصاویر، صدا و تصویر تواماً از طریق اینترنت می‌باشد. این کاربرد نیازمند بهره‌گیری از سرعت و پهنای باند نسبتاً وسیعی از شبکه‌ی اینترنت است. معمولاً در جریان پخش تصویر و صدا برنامه‌های مشتری درخواست فایل‌های تصویری و صوتی فشرده سازی شده‌ای را می‌کند که برروی سرورها مقیم می‌باشند. این گونه سرورها از لحاظ نوع می‌توانند وب سرورهای معمولی و یا سرورهای ویژه‌ی ذخیره و پخش این گونه کاربردها باشند. برمبنای تقاضای

1- Psychoacoustic

برنامه مشتری، برنامه‌ی سرور یک فایل صوتی و یا تصویری را به درون یک سوکت جهت گسیل به مشتری، گسیل می‌کند. اگرچه از هر دو پروتکل TCP و UDP می‌توان جهت انجام این مهم استفاده نمود اما در حال حاضر بیش‌ترین ترافیک پخش تصاویر از طریق TCP انتقال می‌یابد. علت این موضوع آن است که برنامه‌های دیواره‌های آتش (فایروال) به گونه‌ای پیکربندی شده‌اند که ترافیک حمل شده از سوی UDP را مسدود می کنند و دوم آن که با بهره‌گیری از TCP انتقال مطمئن تمامیت فایل به مشتری بدون اتلاف بسته روی می‌دهد و در نتیجه پخش مجدد آن از حافظه‌ی نهان محلی امکان پذیر می‌شود. هنگامی که یک فایل صوتی و تصویری وارد می‌شود، برنامه‌ی مشتری آغاز به بازیابی فایل در فاصله‌ی چندثانیه می‌نماید، برخی از سیستم‌ها هم‌چنین امکانات تعامل کاربران مانند توقف موقت پخش مجدد و پرش موقتی از یک فایل به دیگری را نیز تعیین می‌کنند. **پروتکل بی‌درنگ پخش (RTSP)**[1] در انتهای این بخش تحلیل می‌شود، این پروتکل یک قلمرو عمومی تعامل بین کاربران را تعریف می‌کند.

کاربران اغلب تقاضای پخش صدا و تصویر را از درون یک برنامه‌ی مشتری وب مانند مرورگر می‌کنند اما نمایش و کنترل فایل صدا و تصویر با بهره‌گیری از یک برنامه‌ی **بازپخش رسانه** مانند پخش کننده‌ی رسانه‌ی ویندوز و یا Flashplayer انجام می‌شود. معمولاً یک بازپخش رسانه چندین عملکرد را شامل می‌شود که عبارتند از:

- **بازفشردگی**. جهت صرفه‌جویی در فضای ذخیره‌سازی و نیز پهنای باند معمولاً فایل‌های صوتی و تصویری فشرده سازی می‌شوند. یک برنامه‌ی پخش رسانه هنگام انجام عملیات بازپخش فایل آن را باز فشرده سازی می‌نماید.
- **حذف عوامل ناپایداری**[2]. ناپایداری بسته متغیری است از تأخیرهای مبدأ به مقصد بسته‌ها در درون یک جریان از بسته‌ها، از آن جائیکه صدا و تصویر بایستی با قالب زمان‌بندی مشابه ضبط آن‌ها، پخش شوند معمولاً یک گیرنده بسته‌های دریافتی را به منظور حذف ناپایداری در آن‌ها برای یک زمان کوتاه ذخیره سازی می‌نماید.

۱- دستیابی فایل‌های صدا و تصویر از طریق یک وب سرور

فایل‌های صوتی و تصویری ذخیره شده می‌توانند بر روی یک وب سرور که آن‌ها را از طریق پروتکل HTTP به برنامه‌ی مشتری انتقال می‌دهد قرار گیرند و یا آن که بر روی یک سرور ویژه‌ی پخش بارگذاری شوند که از پروتکل‌های HTTP و یا دیگر پروتکل‌ها بهره می‌گیرد. در این زیربخش چگونگی تحویل فایل‌های صوتی و تصویری از یک وب سرور و در بخش بعد تحویل همین فایل‌ها از طریق یک سرور ویژه‌ی پخش بررسی می‌شوند. تحویل جریان سیال فایل‌های چندرسانه‌ای از طریق پروتکل HTTP به دلیل آن که فایروال‌ها اجازه‌ی عبور ترافیک HTTP را از درون خود می‌دهند علاقه‌مندان فراوانی دارد. بدیهی است بیش‌تر پروتکل‌های خصوصی و وابسته به شرکت‌ها از سوی دیواره‌های آتش مسدود می‌شوند.

ابتدا تحویل جریان سیال فایل‌های صوتی تحلیل می‌شوند، هنگامی که یک فایل صوتی بر روی یک وب سرور مقیم می‌شود، از نقطه نظر چارچوب فایل‌ها در آن سرور این فایل صوتی یک موجودیت معمولی مانند HTML و یا فایل‌های JPEG به شمار می‌آید. هنگامی که کاربر مایل به شنیدن فایل

1- Real time streaming protocol
2- Jitter removal

صوتی می‌باشد، میزبان کاربر یک ارتباط TCP با وب سرور برقرار می‌کند و یک تقاضای HTTP برای دستیابی فایل مورد نظر گسیل می‌کند. بی‌درنگ پس از دریافت تقاضا وب سرور فایل صوتی را در قالب یک پیام پاسخ HTTP بسته‌بندی و آن را از طریق ارتباط TCP به میزبان بازگشت می‌دهد. این موضوع در مورد فایل‌های تصویری اندکی پیچیده‌تر می‌باشد. بخش‌های صدا و تصویر فایل ویدیویی ممکن است **دو** فایل ذخیره شده باشند و یا آن که هر دو بخش در یک فایل مشابه جایگذاری شده و در نتیجه فقط لازم باشد یک موجودیت به مشتری ارسال گردد. به منظور ساده شدن بحث برای حالت ویدیو فرض می‌شود هر دو بخش صوت و تصویر در یک فایل قرار دارند.

در بسیاری از پیاده‌سازی‌های HTTP به منظور انتقال جریان سیال صدا و تصویر، عملیات مربوطه در سمت مشتری به دو بخش تقسیم می‌شود. بخش نخست مربوط به مرورگر است که وظیفه‌ی آن درخواست یک **متافایل**[1] است که حامل اطلاعاتی از قبیل URL و نوع رمزگذاری جهت شناسایی رسانه‌ی پخش مناسب فایل است و بایستی از طریق HTTP تحویل گردد و بخش دوم انتقال این متافایل از مرورگر به برنامه‌ی بازپخش رسانه‌ی مناسب آن است که وظیفه‌ی آن تماس با سرور HTTP و سپس گسیل فایل چند رسانه به پخش رسانه از طریق HTTP می‌باشد. این مراحل از طریق شکل ۷-۱ نشان داده شده است:

۱- کاربر یک اُبَر پیوند (هایپرلینک) را برای یک فایل صدا و تصویر کلیک می‌کند. این فراپیوند مستقیماً به فایل صوتی – تصویری اشاره نمی‌کند بلکه اشاره‌ی آن به متافایل است. این فایل شامل URL فایل صوتی و تصویری اصلی است. پیام پاسخ HTTP که آن متافایل را ببسته‌بندی می‌کند شامل یک خط سرپیام نوع محتوا می‌باشد که کاربر ویژه‌ی آن فایل صوتی، تصویری را تعریف می‌کند.

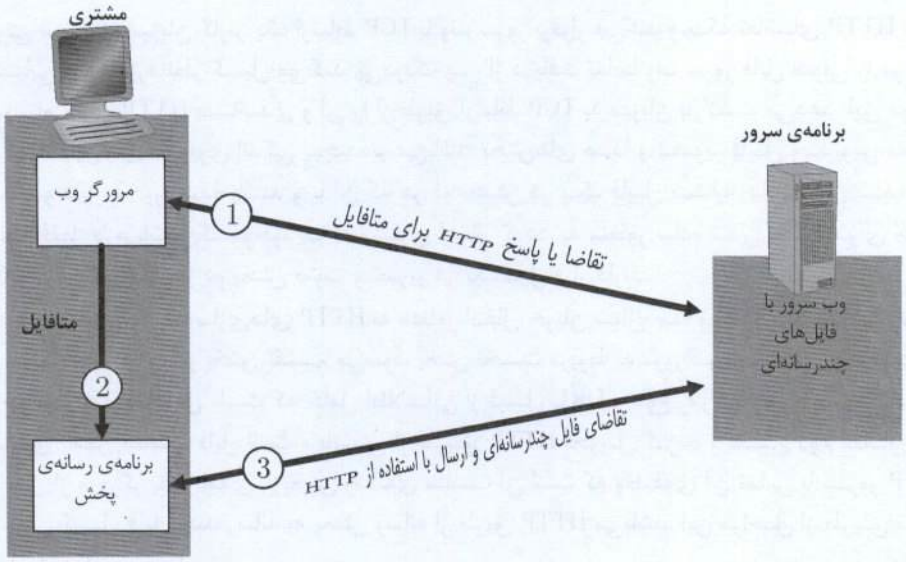
۲- برنامه‌ی مرورگر مشتری، خط سرپیام نوع محتوا را در پیام پاسخ بررسی و برنامه‌ی رسانه‌ی پخش مربوط به آن را فعال می‌سازد. آن‌گاه تمامی بدنه‌ی پیام پاسخ که همان متافایل است به برنامه‌ی رسانه‌ی پخش تحویل می‌گردد. لازم به توضیح است نقش یک متافایل آشکارسازی فایل‌های دیگر است.

۳- برنامه‌ی پخش رسانه یک ارتباط TCP را مستقیماً با سرور HTTP برقرار می‌کند و پیام تقاضای HTTP خود را برای فایل صوتی و تصویری مورد نیاز به درون ارتباط TCP وارد می‌نماید. فایل صوتی و تصویری سپس در قالب یک پیام پاسخ HTTP به رسانه‌ی پخش ارسال می‌شود. سرانجام برنامه‌ی رسانه‌ی پخش فایل صوتی و تصویری را انتشار می‌دهد.

اهمیت مراحل میانی جهت دستیابی متافایل آشکار است. هنگامی که مرورگر نوع محتوای فایل را ملاحظه می کند آن‌گاه می‌تواند رسانه‌ی پخش مناسب آن را فعال و در نتیجه باعث شود رسانه‌ی پخش مستقیماً با سرور مورد نظر ارتباط برقرار نماید.

در این زیربخش ملاحظه شد چگونه یک متافایل اجازه‌ی ارتباط مستقیم یک رسانه‌ی پخش با وب سروری که فایل برروی آن ذخیره شده است را امکان‌پذیر می‌سازد. بدیهی است بسیاری از شرکت‌ها محصولاتی را برای پخش صدا و تصویر ارایه داده‌اند که از معماری تشریح شده در این زیربخش بهره نمی‌گیرند و به‌جای آن پیشنهاد پخش فایل‌های صوتی و تصویری ذخیره‌سازی شده را از طریق سرورهای ویژه‌ی پخش این گونه فایل‌ها می‌کنند که به‌همین منظور بهینه‌سازی شده‌اند.

1- Metafile



شکل ۷-۱ ♦ وب سرور فایل صدا و تصویر را مستقیماً به برنامه‌ی پخش رسانه گسیل می‌کند.

۲- ارسال فایل چند رسانه‌ای از یک سرور پخش به یک مددکار کاربردی

یک سرور پخش جریان پیوسته می‌تواند یک سرور ویژه‌ی اختصاصی مانند Real Networks و یا مایکروسافت و نظایر آن و یا یک سرور جریان پخش قلمرو عمومی باشد. از طریق سرورهای پخش سیال، فایل‌های صوتی و تصویری را می‌توان از طریق HTTP/TCP و یا پروتکل HTTP/UDP با استفاده از پروتکل‌های لایه‌ی کاربرد که از لحاظ طراحی بهتر از HTTP می‌باشند به منظور پخش صدا و تصویر به کار گرفت.

این معماری نیازمند **دو** سرور برمبنای شکل ۷-۲ می‌باشد. یکی از این دو وب سرور است که صفحه‌های وب و نیز متافایل‌ها را شامل می‌شود. دیگری **سرور پخش سیال**[1] می‌باشد که حامل فایل‌های صوتی و تصویری است. هر دو سرور می‌توانند از طریق یک سیستم انتهایی و یا بر روی دو سیستم انتهایی مجزا از یکدیگر به اجرا درآیند. مراحل اجرایی این معماری مشابه آن‌چه در زیربخش قبل توصیف شد، می‌باشد. تنها تفاوت در آن است که رسانه‌ی پخش، فایل موردنظر خود را از یک سرور پخش سیال به جای وب سرور درخواست می‌کند و هر دو برنامه‌ی سرور و رسانه‌ی پخش از طریق پروتکل‌های اختصاصی خویش با یکدیگر تعامل برقرار می‌سازند. این پروتکل‌ها باعث تعامل غنی‌تری از لحاظ پخش فایل‌های صدا و تصویر بین کاربران می‌شوند.

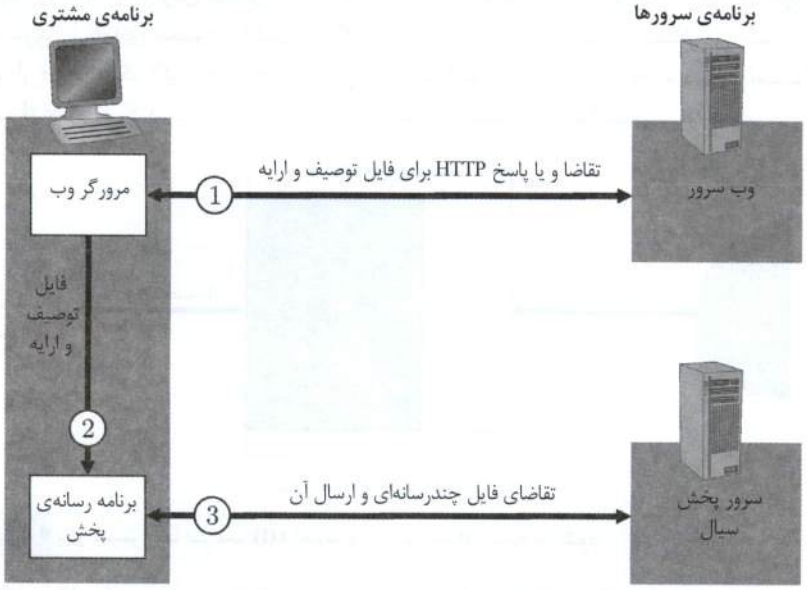
در معماری شکل ۷-۲ امکانات فراوانی جهت تحویل صدا و تصویر از یک سرور پخش به رسانه‌ی پخش وجود دارد. فهرستی از این امکانات به شرح ذیل می‌باشد:

- صدا و تصویر در یک سرعت ثابت برابر نرخ دریافت درگیرنده که سرعت رمزگذاری صدا وتصویر است از طریق پروتکل UDP ارسال می‌گردد. به عنوان مثال اگر فشرده سازی صدا از طریق بهره‌گیری از GSM در سرعت 13 kbps باشد آن‌گاه سرور، فایل صوتی فشرده

1- Streaming file sever

سازی شده را نیز با سرعت 13 kbps گسیل می‌نماید. برنامه‌ی مشتری پس از دریافت فایل صوتی و تصویری فشرده سازی شده از سوی شبکه، آن را باز فشرده‌سازی نموده و سپس پخش می‌کند.

- این وضعیت مشابه حالت قبل است با این تفاوت که رسانه‌ی پخش عملیات بازپخش فایل صدا و تصویر را با تأخیری برابر 2 تا 5 ثانیه انجام می‌دهد و در نتیجه‌ی آن ناپایداری القایی از سوی شبکه را حذف می‌کند. برنامه‌ی مشتری این مهم را از طریق قرار دادن رسانه‌ی فشرده سازی شده که از شبکه دریافت نموده است به درون یک **بافر مشتری**[1] برمبنای شکل ۷-۳ به انجام می‌رساند. هنگامی که مشتری چندثانیه از اطلاعات رسانه را واکشی نمود آن‌گاه آغاز به تهیه ی بافر می‌کند. برای دو مورد اول و دوم اگر فرض شود نرخ وارد کردن اطلاعات x(t) و برای خارج نمودن d باشد آنگاه معمولاً این دو سرعت با یکدیگر برابرند به جز هنگامی که فقدان بسته روی می‌دهد که در این حالت x(t) زمانی کم‌تر از d می‌باشد.



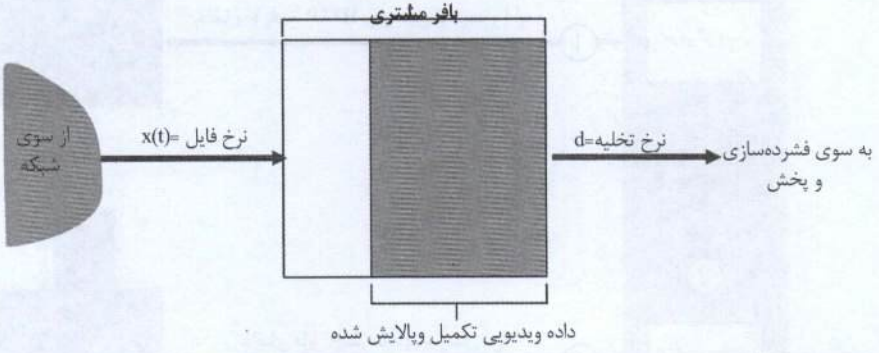
شکل ۷-۲ ♦ پخش پیوسته از یک سرور پخش به یک رسانه‌ی پخش

- مورد سوم ارسال رسانه از طریق TCP است. برنامه‌ی سرور فایل رسانه را بی‌درنگ به درون سوکت TCP تزریق می‌کند و سپس برنامه‌ی مشتری که همان رسانه‌ی پخش است اطلاعات را از سوکت TCP خوانده وتصویر فشرده سازی شده را وارد بافر رسانه‌ی پخش می‌کند. پس از تأخیر 2 تا 5 ثانیه، رسانه‌ی پخش اطلاعات فایل را با سرعت d از بافر خویش خوانده و آن را پس از باز فشرده سازی پخش می‌کند. از آن‌جایی که TCP بسته‌های مفقود شده را مجدداً انتقال می‌دهد در نتیجه توان‌مندی آن را دارد که کیفیت صوتی بهتری را نسبت به UDP تأمین نماید. از طرف دیگر سرعت x(t) می‌تواند باگذر زمان

1- Client buffer

تغییر نماید. این دگرگونی در اثر کنترل تراکم TCP و کنترل جریان پنجره‌ی TCP می‌باشد که در فصل ۳ تشریح شده است. در واقع پس از تلف شدن بسته، پروتکل کنترل تراکم TCP ممکن است سرعت لحظه‌ای را به میزان کمتر از d در فواصل زمانی نسبتاً طولانی کاهش دهد. این موضوع سبب تخلیه‌ی بافر مشتری یعنی فرآیندی می‌شود که اصطلاحاً به آن **گرسنگی**[1] گفته می‌شود و در نتیجه توقف نامطلوبی را در خروجی مشتری برای فایل رسانه فراهم می‌آورد. پژوهش‌های انجام گرفته نشان داده است هنگامی که میانگین نهفتگی TCP تقریباً دوبرابر سرعت رسانه است فرآیند پخش TCP باعث حداقل مقدار گرسنگی و تأخیر اندک در آغاز عملیات پخش می‌شود.

در مورد سوم رفتار تابع x(t) بسیار وابسته به ظرفیت بافر مشتری می‌باشد. لازم به توضیح است بافر مشتری را نبایستی با بافر گیرنده‌ی TCP یکسان به شمار آورد. اگر این بافر یعنی بافر مشتری به اندازه‌ی کافی بزرگ طراحی شود که قادر به نگهداری همه‌ی فایل‌های رسانه باشد، آنگاه TCP از همه‌ی ظرفیت پهنای باند ارتباطی در آن لحظه بهره می‌گیرد تا x(t) به مراتب بزرگ‌تر از d شود. اگر x(t) بسیار بزرگ‌تر از d برای فواصل نسبتاً طولانی از زمان باشد آن‌گاه بخش قابل توجهی از حجم رسانه به درون برنامه‌ی مشتری واکشی می‌گردد و در نتیجه از فرآیند گرسنگی مشتری جلوگیری می‌نماید. از طرف دیگر اگر حجم بافر مشتری کوچک باشد آن‌گاه x(t) از لحاظ میزان تغییرات تقریباً در اطراف d عمل می‌کند، در این حالت ریسک گرسنگی مشتری به مراتب بیش‌تر است.



شکل ۷-۳ ♦ بافر مشتری با سرعت **x(t)** تغذیه و در سرعت **d** تخلیه می‌شود

- **بازنگری تاریخی: پخش صدا و تصویر ذخیره‌سازی شده از Real Network به یوتیوب**

یکی از طلایه‌داران پخش پیوسته‌ی فایل صوتی وتصویری شرکت Real Network می‌باشد که نخستین امکان پخش رسانه را از طریق اینترنت میّسر نمود. نخستین محصول این شرکت با نام Realaudio در سال 1995 به بازار عرضه گردید. در این نرم افزار پخش رسانه یک برنامه‌ی رمزگذار، یک سرور و پخش صدا طراحی شده بود. این نرم افزار به کاربران اجازه می‌داد از طریق اینترنت به گزینش و پخش فایل‌های صوتی براساس تقاضا بپردازند. با اقبال روزافزون کاربران به این سیستم توزیع فایل، دیگران نیز تشویق به ورود در این عرصه شدند.

در حال حاضر پخش سیال صدا و تصویر یکی از محبوب‌ترین خدمات اینترنت به شمار می‌آید.

1- Starvation

تعداد قابل توجهی از شرکت‌ها به طراحی و تولید انواع رسانه‌های پخش، سرورها و پروتکل‌های گوناگون در این زمینه نموده‌اند. تعدادی از این امکانات چندرسانه‌ای پخش صدا و تصویر ذخیره سازی شده بر روی سرورها تا انتهای سال 2009 میلادی شامل موارد ذیل می‌شدند:

- **سیستم چندرسانه‌ای Rhapsody (به مفهوم حماسی) از شرکت Real Networks**- این نرم افزار خدمات اشتراک، بارگیری و پخش فایل‌های چندرسانه‌ای را برای کاربران فراهم می‌کند، این نرم افزار به کمک برنامه‌ی مشتری ویژه‌ی خود آهنگ‌های صوتی را از یک سرور خصوصی ویژه‌ی آن از طریق HTTP بازیابی می‌کند. هنگامی که فایل صوتی از طریق HTTP وارد برنامه‌ی مشتری Rhapsody می‌شود آن فایل پس از باز فشرده سازی انتشار می‌یابد. جهت بارگیری (دانلود) فایل‌های صوتی از یک سیستم **مدیریت حق نشر دیجیتالی**[1] **(DRM)** جهت دستیابی به آن‌ها استفاده می‌شود.
- **سیستم ویدیویی (MSN)**. ازطریق این نرم افزار اختصاصی مایکروسافت، کاربران قادر به دریافت و پخش انواع گوناگونی از امکانات چندرسانه‌ای مانند قطعه‌های صوتی و تصویر موسیقایی و خبری می‌باشند. تصاویر ویدیویی از طریق نرم افزار پخش چند رسانه‌ای ویندوز (WMP) پخش می‌گردد که تقریباً در دسترس همه‌ی میزبان‌های سیستم عامل ویندوز می‌باشد. ارتباط بین WMP و سرورهای مایکروسافت با بهره‌گیری از پروتکل انحصاری **سرور رسانه‌ی مایکروسافت**[2] **(MMS)** صورت می‌پذیرد. این پروتکل پخش فایل‌های چند رسانه‌ای را از طریق دو پروتکل RTSP و RTPانجام می‌دهد. چنانچه این پروتکل‌ها به دلیل وجود فایروال مسدود شوند آن‌گاه از طریق HTTP تلاش بر پخش فایل‌ها می‌کند.
- **سیستم چندرسانه‌ای Muze**. یک خدمات نمونه‌ی صوتی برای فروشندگان از طریق این نرم افزار می‌باشد. عرضه کننده‌هایی مانند یاهو و BestBuy از جمله بهره‌وران این نرم افزار می‌باشند. نمونه‌های گزینش شده‌ی موسیقایی در سامانه‌های این شرکت‌ها در حقیقت از طریق Muze دستیابی می‌شود و با بهره‌گیری WMP نواخته و پخش می‌شوند. سیستم‌های چندرسانه‌ای مانند Muze، یوتیوب، Rhapsody و بسیاری دیگر، از شبکه‌های توزیع محتوی (CDN) جهت توزیع و پخش خود بهره می‌گیرند که در زیربخش ۷-۵ مورد مطالعه قرار می‌گیرد.
- **سیستم چندرسانه‌ای یوتیوب (YouTube)**. این نرم افزار اشتراک فایل‌های تصویری بسیار پرطرفدار، بر مبنای پروتکل پخش Flash جایگذاری شده در درون صفحه‌ی وب عمل می‌کند. ارتباط بین برنامه‌های مشتری و سرورهای یوتیوب از طریق HTTP انجام می‌پذیرد.

اکنون پرسش آن است، در آینده چه سیستم‌هایی ظهور می‌کنند؟ در حال حاضر اکثریت سیستم‌های چندرسانه‌ای پخش ویدیو از کیفیت پائین و سرعت رمزگذاری 500 Kbpsو یا کمتر بهره می‌گیرند. بدیهی است با گسترش دستیابی به اینترنت از طریق پهنای باند گسترده و FTTH کیفیت فایل‌های ویدیویی نیز قطعاً بهینه خواهند شد، به احتمال بسیار قوی جایگزین بسیاری از پخش کننده‌های صوتی و تصویری فعلی که فایل را در خود ذخیره و سپس پخش می‌کنند این امکانات از طریق کانال‌های بی‌سیم و براساس تقاضا در آینده‌ی نزدیک در اختیار کاربران قرار خواهد گرفت.

1- Digital rights Management
2- Microsoft media service

۳- پروتکل پخش سیال بی‌درنگ (RTSP)

بسیاری از کاربران چند رسانه‌ای اینترنت به ویژه آن گروه از آن‌ها که با سیستم کنترل از راه دور TV کار کرده‌اند و با بهره‌گیری از امکانات آن به ویژگی‌های آن خو گرفته‌اند بسیار مایل هستند که از امکانات مشابهی مانند پخش، تغییر مکان پخش، بازگشت کند و سریع عقب و جلو بازپخش گذشته و آینده‌ی فایل‌های چندرسانه‌ای برروی اینترنت و غیره استفاده نمایند. این عملکرد مشابه حالتی است که یک کاربر در حال تماشای پخش تصاویر از یک DVD یا شنیدن یک فایل موسیقی از طریق CD می‌باشد. به منظور تأمین امکانات مختلف کنترل پخش نیاز به یک پروتکل جهت تبادل اطلاعات مربوط به این عملیات بین سرور و پخش رسانه می‌باشد. پروتکل پخش سیال بی‌درنگ (RTSP) در مستند [RFC2326] برای ایجاد این امکانات تعریف شده است. قبل از گام نهادن در جزئیات RTSP نخست چگونگی عملکرد آن بررسی می‌شود:

- پروتکل RTSP روش‌های فشرده سازی تصویر و صدا را در درون خود تعریف نمی‌کند.
- چگونگی بسته‌بندی اطلاعات صوتی و تصویری و تعریف بسته جهت انتقال از طریق یک شبکه نیز از سوی RTSP تعریف نمی‌شود. بسته‌بندی پخش رسانه از طریق RTP و یا پروتکل خصوصی ویژه انجام می‌گیرد. به عنوان مثال سرورهای صدا و تصویر در Real Networks از RTSP جهت ارسال اطلاعات کنترل به یکدیگر بهره می‌گیرند، اما جریان رسانه می‌تواند در قالب بسته‌های RTP بسته‌بندی گردد و یا آن که از قالب داده‌های خصوصی استفاده نماید.
- پروتکل RTSP چگونگی انتقال پخش رسانه را محدود نمی‌کند و بنابراین انتقال می‌تواند از طریق TCP یا UDP انجام گیرد.
- علاوه برآن ظرفیت بافرهای پخش رسانه از لحاظ صدا و تصویر نیز از طریق پروتکل RTSP محدود نمی‌گردد. پخش صدا و تصویر می‌تواند بی‌درنگ به هنگام آغاز ورود آن در برنامه‌ی مشتری انجام گیرد. پخش فایل صدا و یا تصویر می‌تواند پس از یک تأخیر چندثانیه‌ای انجام گیرد و یا آن که قبل از پخش فایل به صورت کامل دانلود و آن‌گاه پخش انجام گیرد.

بنابراین اگر پروتکل RTSP هیچ یک از موارد اشاره شده را انجام نمی‌دهد، پرسش قابل طرح آن است که چه عملی را انجام می‌دهد؟ پاسخ آن است که پروتکل RTSP اجازه می‌دهد یک رسانه‌ی پخش از نقطه نظر انتقال کنترل شود. قبلاً اشاره گردید عملیات کنترل شامل توقف و پخش مجدد، تغییر موقعیت مکان محتوای در حال پخش، عقب و جلو راندن کند و سریع فایل و غیره می‌باشد. پروتکل RTSP را اصطلاحاً یک **پروتکل خارج از محدوده (باند)** می‌نامند. پیام‌های RTSP خارج از باند ارسال می‌شوند در حالی که پخش رسانه‌هایی که از RTSP به منظور ساختار بسته‌های خود بهره نمی‌گیرند درون باند یا محدوده تلقی می‌شوند. پیام‌های پروتکل RTSP از شماره پورت 544 بهره می‌گیرند که با شماره درگاه رسانه‌ی پخش متفاوت می‌باشد. مشخصات RTSP در مستند [RFC2326] امکان ارسال پیام‌های RTSP از طریق TCP و یا UDP را امکان‌پذیر می‌سازد.

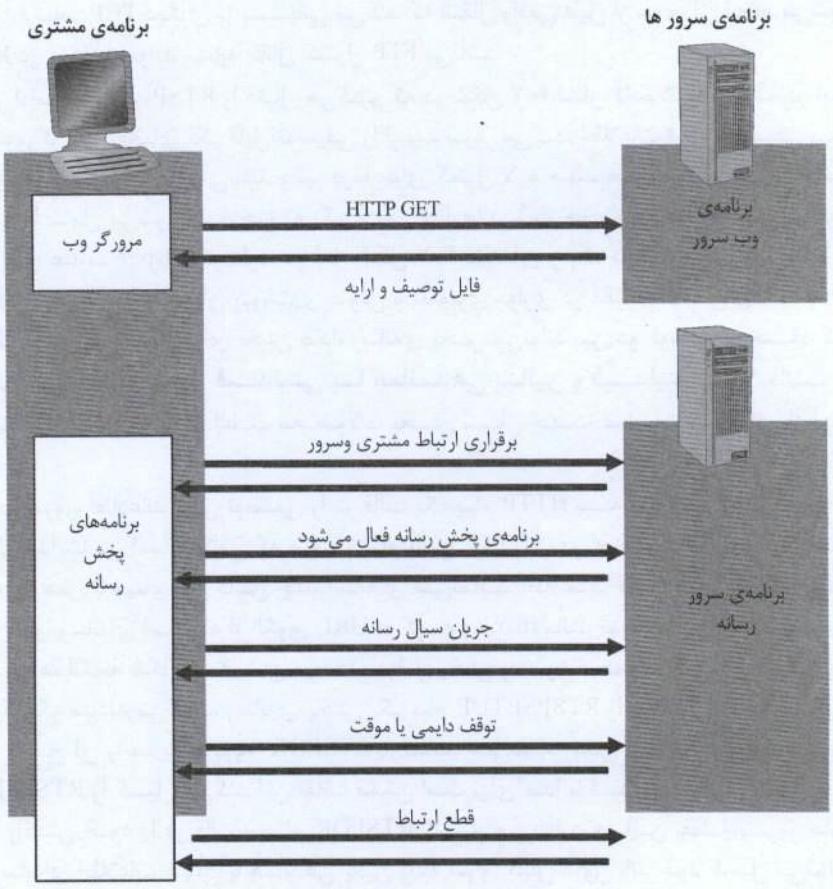
با یادآوری مطالب فصل ۲ که در آن عنوان شد پروتکل انتقال فایل (FTP) از فرآیند خارج از باند بهره می‌گیرد، به مفهوم دیگر پروتکلی مانند FTP از دو جفت سوکت مشتری و سرور که هر جفت، شماره پورت ویژه‌ی خود را اختیار می‌کند، استفاده می‌برد. یک جفت سوکت سرور و مشتری از این دو یک ارتباط TCP را پشتیبانی می‌کند که اطلاعات کنترل را انتقال می‌دهد. جفت سوکت دیگر مشتری

و سرور ارتباط TCP دیگری را پشتیبانی می‌کند که انتقال واقعی فایل از سوی آن انجام می‌شود. کانال RTSP در بسیاری موارد مشابه کانال کنترل FTP می‌باشد.

در ادامه مثالی از RTSP را دنبال می‌کنیم که در شکل ۷-۴ نشان داده شده است. در این شکل ابتدا مرورگر وب تقاضای یک فایل توصیفی را از وب سرور می‌کند. اطلاعات فایل توصیفی به چندین فایل رسانه‌ای، قابل ارجاع می‌باشد و نیز فرمان‌های کنترل لازم جهت همزمانی فایل‌های چندرسانه‌ای پیوسته را شامل می‌شود. هر ارجاع به یکی از این فایل‌ها در آغاز خود از یک URL بهره می‌گیرد که در ابتدای آن عبارت rtsp:// قرار دارد. در ذیل شکل ۷-۴ نمونه‌ای از یک فایل توصیفی قید شده است. در این فایل توصیفی یک جریان پیوسته‌ی صوتی و تصویری موازی در قالب همزمان (به نام lipsyn) با یکدیگر پخش می شوند. برای پخش صدا، رسانه‌ی پخش می‌تواند بین دو گونه روش ضبط کلید زنی نماید. این دو عبارتند از **فیدلیتی یا هماندهی پائین** و **فیدلیتی بالا**. قالب فایل به صورت SMIL می‌باشد که اغلب محصولات پخش سیال جهت عملیات همزمانی از آن استفاده می‌کنند.

وب سرور، اطلاعات فایل توصیفی را در قالب یک پیام HTTP بسته‌بندی می‌کند و آن را به سوی مرورگر هدایت می‌کند. هنگامی که مرورگر پیام پاسخ HTTP را دریافت نمود اقدام به فعال‌سازی یک رسانه‌ی پخش برمبنای نوع فایل چندرسانه‌ای می‌نماید. اطلاعات فایل توصیفی از جمله شامل جریان‌های رسانه‌ای است که از الگوی URL به گونه‌ی: rtsp:// مانند نمونه‌ی اشاره شده در بالا بهره می‌گیرد. همانگونه شکل ۷-۴ نشان می‌دهد، عوامل پخش و سرور مجموعه‌ای از پیام‌های RTSP را بین یکدیگر مبادله می‌کنند. رسانه‌ی پخش یک پیام RTSPSETUP را به عنوان تقاضا ارسال می‌کند و سرور پاسخ آن را با یک پیام RTSPOK بازگشت می‌دهد. پس از آن رسانه‌ی پخش تقاضای RTSPPLAY را گسیل می‌کند این تقاضا ممکن است برای صدا با فیدلیتی پائین باشد و در نتیجه سرور واکنش خود را در قالب پیام RTSPOK گسیل می دارد. در این نقطه سرور حامل فایل چندرسانه‌ای اطلاعات صوتی با هماندهی پائین را به سوی کانال درون باند خود گسیل می‌نماید. پس از آن رسانه‌ی پخش یک تقاضای RTSPPAUSE گسیل می‌کند و سرور پاسخ آن را در قالب RTSPOK فراهم می‌آورد. هنگامی که کاربر به انتهای کاربرد خود نزدیک می‌شود، رسانه‌ی پخش پیام RTSPTEARDOWN را به عنوان تقاضا به سوی سرور گسیل می‌کند و پاسخ RTSPOK را از سوی سرور دریافت می‌دارد.

```
<title>Twister</title>
<session>
    <group language=en lipsync>
       <switch>
          <track type=audio
                e="PCMU/8000/1"
                src="rtsp://audio.example.com/twister/audio.en/lofi">
          <track type=audio
                e="DVI4/16000/2" pt="90 DVI4/8000/1"
                src="rtsp://audio.example.com/twister/audio.en/hifi">
             </switch>
          <track type=video/jpeg"
                src="rtsp://video.example.com/twister/video">
    <group>
<session>
```



شکل ۷-۴ ♦ تعامل بین برنامه‌های مشتری و سرور با بهره‌گیری از RTSP

اکنون به بررسی پیام‌های RTSP می‌پردازیم. در ذیل یک نمونه‌ی ساده شده از یک تعامل RTSP بین برنامه‌ی مشتری (C:) و سرور (S:) بررسی شده است.

```
C: SETUP rtsp: // audio.example.com/twister/audio RTSP/1.0
   Cseq: 1
   Transport: rtp/udp; compression; port=3056; mode=PLAY
S: RTSP/1.0 200 OK
   Cseq: 1
   Session: 4231
C: PLAY rtsp: //audio.example.com/twister/audio.en/lofi RTSP/1.0
   Range: npt=0-
   Cseq: 2
   Session: 4231
S: RTSP/1.0 200 OK
   Cseq: 2
   Session: 4231
C: PAUSE rtsp: //audio.example.com/twister/audio.en/lofi RTSP/1.0
   Range: npt=37
```

```
   Cseq: 3
   Session: 4231
S: RTSP/1.0 200 OK
   Cseq: 3
   Session: 4231
C: TEARDOWN rtsp: //audio.example.com/twister/audio.en/lofi RTSP/1.0
   Cseq: 4
   Session: 4231
S: RTSP/1.0 200 OK
   Cseq: 4
   Session: 4231
```

نکته‌ی جالب توجه در این نمونه آن است که شباهت آشکاری بین دو پروتکل HTTP و RTSP وجود دارد. همه‌ی پیام‌های پرسش و پاسخ در قالب متن اسکی (ASCII) می‌باشند و برنامه‌ی مشتری از روش‌های استاندارد SETUP، PLAY، PAUSE و غیره بهره می‌گیرد و نیز واکنش برنامه‌ی سرور در قالب کدهای استاندارد است. تنها تفاوت مهم در این نکته است که سرور RTSP به رهگیری همه‌ی حالت‌های برنامه‌ی مشتری برای هریک از تعامل‌های در جریان RTSP می‌پردازد. به عنوان مثال برنامه‌ی سرور وضعیت مشتری را از لحاظ این نکته که آیا در حالت آغازین، پخش و یا توقف قرار دارد، زیرنظر می‌گیرد. یک تعامل و شمارگان ردیف که بخشی از هر تقاضا و پاسخ RTSP می‌باشند به برنامه‌ی سرور کمک می‌کنند که وضعیت تعامل را رهگیری نماید. معمولاً شماره شناسه‌ی تعامل در سراسر انجام آن تعامل ثابت می‌باشد، برنامه‌ی مشتری ارزش شماره‌ی ردیف را هر مرتبه که یک پیام جدید گسیل می‌شود، افزایش می‌دهد. برنامه‌ی سرور پس از آن شماره شناسه‌ی تعامل را به همراه شماره‌ی ردیف جاری آن بازگشت می‌دهد.

همان‌گونه که در مثال نشان داده شده است، برنامه‌ی مشتری تعامل را با پیام درخواست SETUP آغاز می‌کند. این مهم را از طریق فراهم آوردن URL متعلق به فایل چندرسانه‌ای که بایستی پخش گردد و نوع نسخه‌ی پروتکل RTSP انجام می‌دهد. پیام تقاضای SETUP شامل شماره پورت برنامه‌ی مشتری است که رسانه بایستی به آن گسیل گردد. این پیام هم چنین مشخص می‌کند که فایل چندرسانه بایستی از طریق UDP با بهره‌گیری از پروتکل بسته‌بندی RTP انجام گیرد. بدیهی است در این مثال پخش رسانه نسبت به بازپخش کامل آنچه ارایه شده است، نمی‌پردازد بلکه فقط پخش با همانده‌ی پائین را بازپخش می‌کند.

یادآوری این نکته ضروری است که پروتکل RTSP قادر به انجام عملیات بیشتری در مقایسه با آنچه در این مختصر توصیف شده است می‌باشد. به ویژه پروتکل RTSP به برنامه‌ی مشتری امکان می‌دهد با بهره گیری از امکاناتی نظیر پخش جریان به سوی سرور جهت انجام عملیات ضبط اقدام نماید.

۵-۷ بهره‌گیری از امکان خدمات بهترین تلاش

پروتکل لایه‌ی شبکه اینترنت (IP) خدمات برمبنای بهترین تلاش را که به مفهوم انتقال هر داده گرام از منبع به مقصد با سرعت هرچه بیش‌تر می‌باشد فراهم می‌آورد. با این حال ضمانت لازم در مورد میزان تأخیر انتها به انتها که به هر بسته ممکن است تحمیل گردد و یا میزان ناپایداری و فقدان بسته‌ها در درون یک جریان حامل آنها را تأمین نمی‌کند. این مهم سبب چالش‌های قابل توجه برای

طراحی کاربردهای بی‌درنگ چندرسانه‌ای مانند تلفن اینترنتی و همایش ویدیویی بی‌درنگ خواهد شد زیرا این موارد بسیار حساس در برابر تأخیر، ناپایداری و فقدان بسته‌ها می‌باشند.

در این بخش بهینه‌سازی کارایی کاربردهای چندرسانه‌ای در شبکه‌های با امکانات خدمات بهترین تلاش تحلیل می‌شود. بیش‌ترین تمرکز بر روش‌های لایه‌ی کاربرد یعنی رویکردهایی که نیازمند هیچ گونه تغییری در هسته‌ی شبکه و یا حتی لایه‌ی انتقال در میزبان‌های انتهایی می‌باشند، قرار گرفته است. ابتدا تأثیرهای ناشی از فقدان، ناپایداری تأخیر و نیز تأخیرها در کاربردهای چندرسانه‌ای بررسی می‌شوند و آن‌گاه روش‌های بازیابی از تأثیر این اختلال‌ها توصیف می‌شود و در انتهای بخش نیز توضیحاتی پیرامون شبکه‌های توزیع محتوا و منابع با هدف جلوگیری از وقوع چنین اختلال‌هایی ارایه می‌گردد.

۱- محدودیت‌های اعمال خدمات با بهترین تلاش

قبلاً اشاره شد خدمات با بهره گیری از بهترین تلاش می‌تواند سبب فقدان بسته، تأخیر چشمگیر انتها به انتها و ناپایداری بسته گردد. تأثیرهای مورد اشاره را می‌توان در چارچوب یک **کاربرد تلفنی در اینترنت (iphone)** تحلیل نمود. این وضعیت بسیار مشابه یک کاربرد ویدیویی بی‌درنگ است.

در یک مکالمه‌ی تلفنی از طریق IP، معمولاً مکالمه یک سیگنال صوتی شامل بخش‌های مربوط به صدای مکالمه کننده و فواصل سکوت وی هنگام شنیدن طرف مقابل می‌باشد. به منظور صرفه جویی در پهنای باند، کاربرد تلفنی، IP بسته‌های داده‌ای را تولید می‌کند که فقط شامل بخش صدا و نه سکوت آن می‌باشد. در هنگام انجام مکالمه این بخش از اطلاعات در قالب بایت‌های با سرعت 8000 بایت در ثانیه تولید می‌شوند و در فاصله‌ی زمانی هر 20 میلی ثانیه فرستنده بایت‌ها را در قالب گروهی جمع‌آوری می‌کند. بنابراین تعداد بایت‌های صوتی در یک گروه برابر است با 160 بایت، به عبارت دیگر:

$$\text{تعداد بایت‌های صدا در هر گروه} = 20\text{ms} \times 8000\,\text{Byte/S}$$

به هر گروه از بایت یک سر پیام ویژه نیز افزوده می‌شود که اطلاعات درون آن در ادامه تشریح خواهد شد، پس از آن گروه بایت به همراه سرپیام آن در قالب یک قطعه‌ی UDP بسته‌بندی و به واسط سوکت تحویل می‌شود. بنابراین در فاصله‌های زمانی 20 میلی ثانیه یک قطعه‌ی UDP هنگام انجام مکالمه تلفنی IP ارسال می‌شود.

اگر هر بسته موفق به ورود در گیرنده گردد وتأخیر انتها به انتهای ثابتی نیز داشته باشد، آن‌گاه بسته به گونه‌ی متناوب هر 20 میلی ثانیه هنگام انجام مکالمه وارد گیرنده می‌شود. در این وضعیت ایده‌آل گیرنده فقط می‌تواند به بازپخش هر گروه از بسته‌ها بی‌درنگ پس از وارد شدن آن‌ها بپردازد. با این حال برخی از بسته‌ها ممکن است تلف شوند و اغلب آن‌ها نیز تأخیر انتها به انتهای مشابهی را ندارند حتی در هنگامی که تراکم ترافیک اینترنت چندان هم سنگین نمی‌باشد. به همین منظور گیرنده بایستی موارد ذیل را مورد توجه بیش‌تر خود قرار دهد:

- چه هنگام بازپخش یک گروه بسته را انجام دهد؟
- چه تدبیری در مورد فقدان یک گروه بسته بایستی صورت پذیرد؟

- **فقدان بسته**

یکی از قطعه‌های UDP تولید شده از یک کاربرد اینترنت تلفنی را در نظر بگیرید. هر قطعه‌ی UDP در قالب یک داده گرام IP بسته بندی می‌شود. هنگامی که داده در شبکه حرکت می‌کند ناگزیر از درون

بافرها و به عبارت دیگر صف‌ها در مسیریاب عبور خواهد نمود تا به پیوندهای خروجی وارد گردد. این احتمال وجود دارد که یک و یا چندین بافر در مسیر بین فرستنده و گیرنده به علت تکمیل ظرفیت قادر به پذیرش داده گرام‌های IP نبوده و در نتیجه از این داده گرام‌ها صرف‌نظر می‌شود و هرگز آن‌ها وارد کاربرد دریافت کننده نخواهند شد.

به منظور جلوگیری از فقدان بسته‌ها می‌توان از ارتباط‌های TCP بهره گرفت، زیرا TCP قادر به انجام عملیات انتقال مجدد داده گرام‌ها در صورت مفقودی آن‌ها می‌باشد. روش‌های انتقال مجدد بسته‌ها در این حالت به علت آن که کاربردهای صوتی و تصویری نیازمند تعامل بدون درنگ می‌باشند غیرقابل قبول به نظر می‌رسد. به عنوان مثال هنگام انجام یک مکالمه‌ی تلفن اینترنتی انجام عملیات انتقال مجدد باعث افزایش زمان تأخیر انتها به انتها خواهد شد. علاوه بر آن به علت کنترل تراکم TCP سرعت انتقال در فرستنده نیز به دنبال اتلاف بسته‌ها به سطح پائین‌تر از سرعت دریافت در گیرنده کاهش پیدا می‌کند. این رخداد تأثیر جدی بر کیفیت صدا و تصویر درگیرنده از نقطه نظر تشخیص دارد. به همین سبب اغلب کاربردهای تلفن اینترنتی از طریق ارتباط‌های UDP انجام می‌گیرند و بنابراین انتقال مجدد داده گرام‌ها را در صورت فقدان آنها انجام نمی‌دهند. لازم به یادآوری است کاربردی مانند اسکایپ به صورت عادی از UDP بهره می‌گیرد مگر آن که کاربر پشت NAT و یا یک دیواره‌ی آتش (فایروال) که قطعه‌های UDP را مسدود می‌نمایند، پنهان شده باشد، در این حالت از TCP استفاده می‌شود.

بدیهی است اتلاف بسته‌ها در کاربردهای چندرسانه‌ای آن چنان هم نگران کننده نمی‌باشد، زیرا میزان فقدان بسته بین یک تا 20 درصد قابل تحمل است. میزان تحمل پذیری اتلاف داده گرام با استفاده از UDP در کاربردهای چندرسانه‌ای بستگی به چگونگی و نوع رمزگذاری و انتقال اطلاعات صوتی و نیز مراتب پنهان سازی قابل قبول اتلاف درگیرنده دارد. به عنوان مثال عامل تصحیح خطا (FEC) می‌تواند اتلاف بسته‌ها را به نوعی از دید گیرنده پنهان نماید. در ادامه‌ی مطالب همین فصل ملاحظه خواهد شد چگونه به کمک FEC اطلاعات افزونه به همراه اطلاعات اصلی گسیل می‌شوند تا بازیابی برخی از اطلاعات اصلی مفقود شده از طریق همان اطلاعات افزونه امکان‌پذیر گردد. در هر حال اگر یک یا چندین پیوند بین فرستنده و گیرنده دچار تراکم شدید باشد و نرخ اتلاف بسته‌ها فراتر از 10 تا 20 درصد آن‌گاه عملاً نمی‌توان دریافت اطلاعات صوتی با کیفیت قابل قبول را انتظار داشت. آشکار است خدمات با رویکرد بهترین تلاش محدودیت‌های ویژه‌ی خود را دربردارد.

- **تأخیر انتها به انتها**

زمان تأخیر انتها به انتها عبارت است از تجمیع تأخیرهای انتقال، پردازش و صف‌بندی در روترها، تأخیر انتشار در پیوندها و تأخیرهای حاصل از پردازش در سیستم‌های انتهایی. برای کاربردهای تعامل‌گرای صوتی مانند تلفن اینترنتی تأخیرهای انتها به انتهای کمتر از 150 میلی ثانیه قابل تشخیص از سوی انسان شنونده نمی‌باشد. از طرف دیگر تأخیرهای بین 150 و 400 میلی ثانیه اگرچه مطلوب نیستند اما قابل قبول می‌باشند و بدیهی است تأخیرهای فراتر از 400 میلی ثانیه تعامل واقعی در یک مکالمه‌ی صوتی را دچار تهدید جدی می‌کند. طرف دریافت کاربرد تلفن اینترنتی تقریباً همه‌ی بسته‌هایی که دچار تأخیر بیش از آستانه‌ی معینی باشند را حذف می‌کند. به عنوان مثال اگر تأخیر بیش از 400 میلی ثانیه باشد بسته‌های با تأخیر بیش از این آستانه به صورت مؤثر تلف می‌شوند.

- **ناپایداری بسته (Packet jitter)**

تأخیرهای تصادفی در اثر تشکیل صف در مسیریاب‌ها یکی از عناصر حیاتی تأخیر انتها به انتها به شمار می‌آید. به علت وجود این تأخیرهای متغیر در درون شبکه فاصله‌ی زمانی بین تولید یک بسته در منبع تا زمان دریافت آن از سوی مقصد با توجه به طبیعت ساختاری هر بسته متغیر می‌باشد. این تغییرات تصادفی در زمان را اصطلاحاً **ناپایداری** می‌نامند.

برای درک بهتر موضوع دو بسته‌ی صوتی در پی یکدیگر را در یک مکالمه تلفنی از طریق شبکه‌ی اینترنت درنظر بگیرید. فرستنده بسته دوم خود را در فاصله‌ی زمانی 20 میلی ثانیه پس از گسیل بسته‌ی اول انجام می‌دهد. اما در گیرنده فاصله زمانی بین دریافت این دو بسته ممکن است فراتر از 20 میلی ثانیه باشد. جهت مشاهده‌ی این موضوع فرض کنید بسته‌ی اول وارد یک صف تقریباً تهی در یک مسیریاب می‌شود در حالیکه قبل از ورود بسته‌ی دوم در صف تعداد انبوهی از بسته‌ها از منابع دیگر در حال ورود به همان صف می‌باشند. ملاحظه می‌شود نخستین بسته با تأخیر صف نسبتاً اندکی روبه رو می‌شود اما بسته دوم برعکس متحمل یک زمان تأخیر نسبتاً طولانی خواهد شد و در نتیجه فاصله‌ی زمانی بین دو بسته از 20 میلی ثانیه بیش‌تر خواهد شد. بدیهی است فاصله‌ی زمانی بین بسته‌ها می‌تواند کوتاه‌تر از 20 میلی ثانیه نیز باشد. برای ملاحظه‌ی این موضوع بار دیگر دو بسته‌ی صوتی در قالب داده گرام IP پی در پی را درنظر بگیرید. اگر فرض شود بسته‌ی اول وارد انتهای یک صف طولانی در روتر شده است و بسته‌ی دوم نیز قبل از بسته‌های ارسال شده از سوی منابع دیگر وارد همان صف گردد آن‌گاه دو بسته‌ی اول و دوم اگرچه در پی یکدیگر در صف قرار می‌گیرند، اما اگر زمان لازم جهت انتقال یک بسته بر روی پیوند خروجی مسیریاب نبایستی کمتر از 20 میلی ثانیه باشد آن‌گاه بسته‌های اول و دوم در فاصله‌ی زمانی کمتر از 20 میلی ثانیه از یکدیگر قرار می‌گیرند.

این حالت‌ها را می‌توان با خودروهای در حال حرکت در جاده‌ها شبیه‌سازی نمود. فرض کنید شما و دوستتان هرکدام در حال رانندگی با خودروی خویش از تهران به مقصد اصفهان در حرکت می‌باشید. هم‌چنین فرض کنید هر دو از لحاظ مهارت در رانندگی نیز مشابه یکدیگر و با سرعت 100 کیلومتر در ساعت مشروط بر شرایط ترافیکی مناسب در حال حرکت می‌باشید. سرانجام فرض کنید دوست شما به فاصله‌ی یک ساعت قبل از شما رانندگی خود را آغاز می‌کند. برمبنای وقوع شرایط ترافیکی ممکن است شما با زمان تأخیر کمتر و یا بیش‌تر از یک ساعت پس از دوست خود وارد اصفهان شوید.

اگر گیرنده‌ای از حضور ناپایداری صرف‌نظر نماید و گروه بسته‌ی صوتی وارد شده را بی‌درنگ پس از دریافت بازپخش نماید آن‌گاه کیفیت صدای دریافت شده به آسانی می‌تواند در گیرنده غیرقابل شنود باشد. بدیهی است در اغلب موارد ناپایداری را می‌توان با بهره‌گیری از **شماره‌های ردیف**[1] و یا **تأخیر بازپخش**[2] و سرانجام از طریق **برچسب‌های زمانی**[3] حذف نمود. این موارد همگی در ادامه تشریح شده‌اند.

۲- حذف ناپایداری در گیرنده برای صدا

در یک کاربرد صوتی مانند تلفن اینترنتی و مانند آن گیرنده بایستی امکانات لازم جهت بازپخش همزمان گروه بسته‌های صوتی را در حضور ناپایداری تصادفی شبکه تامین کند. کاربردهای تصویری نیز

1- Sequence number
2- Playout delay
3- Time stamps

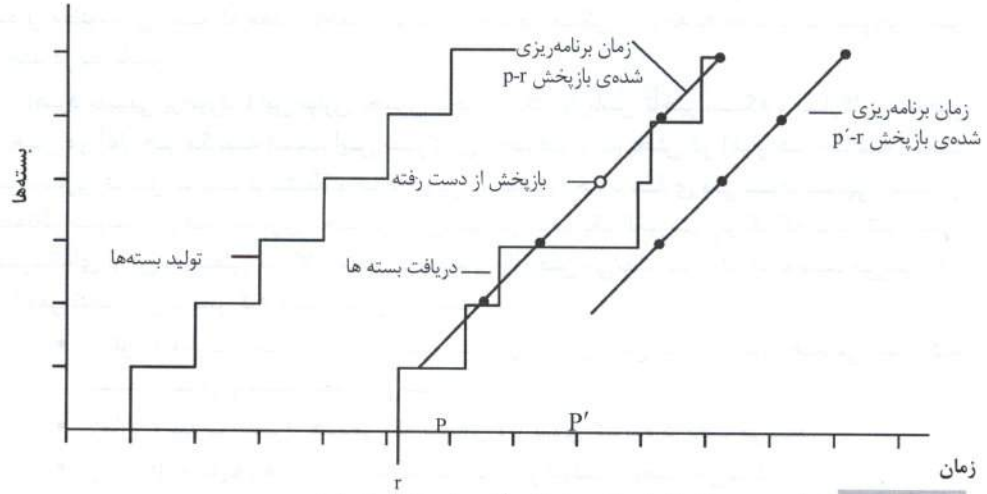
از نیازهای مشابه برخوردار می‌باشند. تأمین این نیاز در قالب مکانیزم‌های ذیل انجام می‌شود:

- دیباچه گذاری هر گروه بسته با یک **شماره‌ی ردیف**. فرستنده با تولید هر بسته ارزش این شناسه را به مقدار یک افزایش می‌دهد.
- سرآغاز گذاری هر گروه بسته با یک **برچسب زمان**. فرستنده هر گروه از بسته‌های صوتی را با یک برچسب زمانی که مشخص کننده‌ی زمان تولید آن گروه بسته است شناسه‌گذاری می‌کند.
- **تأخیر در بازپخش**، گروه بسته‌های صوتی در گیرنده. تأخیر بازپخش اطلاعات صوتی دریافتی بایستی به اندازه کافی طولانی باشد تا اغلب بسته‌ها قبل از زمان تعریف شده برای بازپخش دریافت شوند. این زمان تأخیر در بازپخش می‌تواند در تمام مدت انجام تعامل صوتی ثابت و به گونه‌ی قابل سازگار متغیر در طول زمان حیات تعامل باشد. بسته‌هایی که قبل از زمان تعیین شده جهت بازپخش دریافت نگردند، فراموش و یا مفقود شده به شمار می‌آیند. بدیهی است گیرنده ممکن است از نوعی جایگذاری صوتی برای پنهان‌سازی اتلاف بسته‌ها بهره گیرد.

در ادامه‌ی مطلب ترکیب سه مکانیزم اشاره شده در بالا و چگونگی عملکرد آن‌ها به منظور حذف تأثیرهای ناپایداری تحلیل می‌شود. دو استراتژی بازپخش یعنی تأخیر ثابت و تأخیر خودسازگار بازپخش به صورت ذیل قابل توصیف می‌باشند:

الف) تأخیر ثابت بازپخش

در استراتژی تأخیر ثابت، گیرنده تلاش می‌کند هر گروه بسته را دقیقاً q میلی ثانیه پس از تولید آن بازپخش نماید. بنابراین اگر گروه بسته دارای یک برچسب در زمان t شود، گیرنده گروه بسته را در زمان t+q بازپخش می‌نماید مفروض برآن که گروه داده‌ی مورد اشاره در همان زمان قید شده وارد گیرنده شده باشد. بسته‌هایی که پس از زمان مشخص شده جهت بازپخش دریافت شوند حذف و به عنوان مفقود شده به شمار می‌آیند.



شکل ۷-۵ ♦ فقدان بسته برای اندازه‌های مختلف تأخیرهای ثابت بازپخش

گزینش مناسب برای q چیست؟ تلفن اینترنتی قادر به پشتیبانی تأخیرهای تا 400 میلی ثانیه می‌باشد، اگرچه ارزش‌های کوچکتر q احتمال دستیابی به یک تعامل رضایت‌بخش را افزایش می‌دهد. از طرف دیگر اگر q کمتر از 400 میلی ثانیه درنظر گرفته شود آنگاه بسیاری از بسته‌ها ممکن است زمان بازپخش اختصاص یافته به خود را در اثر ناپایداری القایی شبکه برای بسته‌ها از دست دهند. تقریباً اگر تغییرات تأخیر انتها به انتها بسیار چشمگیر باشد، آن‌گاه بهینه است که q ارزش بزرگی را اختیار نماید. از طرف دیگر اگر تأخیر و تغییرات آن نیز کوچک باشد مطلوب آن است که مقدار q نیز کوچک طراحی شود، شاید کمتر از 150 میلی ثانیه.

توازن بین تأخیر و اتلاف بازپخش در شکل ۷-۵ نشان داده شده است. شکل ۷-۵ زمان تولید بسته‌ها و بازپخش آن‌ها را برای یک قطعه‌ی مکالمه‌ی صوتی نشان می‌دهد. دو تأخیر بازپخش اولیه و متمایز از یکدیگر تحلیل می‌شوند. آن‌چه نمودار سمت چپ شکل ۷-۵ نشان می‌دهد تولید بسته‌ها در فواصل زمانی معین مانند هر 20 میلی ثانیه از سوی فرستنده می‌باشد، نخستین بسته در این گروه در زمان r دریافت شده است. همان‌گونه که در شکل ۷-۵ نشان داده شده است ورود بسته‌های بعدی به علت ناپایداری شبکه در فواصل زمانی یکسان انجام نمی‌گیرد.

نخستین زمان بازپخش دارای تأخیر ثابت اولیه‌ی p-r می‌باشد. با توجه به این زمان‌بندی چهارمین بسته در زمان اختصاص یافته به آن جهت بازپخش وارد نمی‌شود و گیرنده آن را به عنوان مفقود شده قلمداد می‌کند. در زمان برنامه‌ریزی شده‌ی دوم تأخیر بازپخش ثابت اختصاص یافته p'-r می‌باشد، برای این زمان‌بندی همه‌ی بسته‌ها قبل از زمان بازپخش خود وارد می‌شوند و بنابراین مفقودی بسته وجود ندارد.

ب) تأخیر بازپخش خودسازگار

مثال اشاره شده در بالا بیان کننده‌ی نکات مهم در طراحی استراتژی یک بازپخش با زمان تأخیرهای ثابت می‌باشد. از طریق افزایش زمان تأخیر بازپخش اغلب بسته‌ها به هدف زمانی تعریف شده‌ی خود دستیابی دارند و بنابراین دچار اتلاف ناچیزی می‌شوند. به منظور دستیابی به خدمات تعامل‌گرا مانند تلفن اینترنتی تأخیرهای طولانی می‌توانند مشکل آفرین شوند اگرچه قابل تحمل می‌باشند. به صورت ایده‌آل مطلوب آن است که مقدار تأخیر بازپخش در حداقل ممکن با توجه به محدودیت مفقودی کمتر از چند درصد باشد.

راهبرد طبیعی برخورد با این توازن تخمین تأخیر شبکه، واریانس تأخیر شبکه و سازگاری تأخیر بازپخش در آغاز هر مکالمه است. این سازگاری تأخیرهای بازپخش در آغاز هر مکالمه سبب فشرده‌سازی فواصل سکوت فرستنده و صرفه‌جویی در فضای ذخیره سازی می‌شود. بدیهی است از نقطه‌نظر شنونده این فشرده‌سازی محسوس نمی‌باشد. در ادامه یک الگوریتم ژنریک که یک گیرنده‌ی چندرسانه‌ای از آن به منظور سازگار نمودن تأخیرهای بازپخش می‌تواند بهره گیرد، توصیف می‌شود. در ابتدا مفروضات ذیل درنظر گرفته شده بیان می‌شوند:

- زمان t_i عبارت است از برچسب زمانی بسته‌ی i و این زمان بیان کننده‌ی هنگامی است که بسته از سوی فرستنده تولید می‌شود.
- زمان r_i عبارت است از هنگامی که بسته‌ی i از سوی گیرنده دریافت می‌شود.
- زمان p_i به مفهوم هنگامی است که بسته‌ی i درگیرنده بازپخش می‌شود.

بدیهی است تأخیر انتها به انتها در شبکه برای بسته‌ی i معادل r_i-t_i می‌باشد. به علت ناپایداری

شبکه این تأخیر از یک بسته نسبت به بسته‌ی دیگر متفاوت است. اگر d_i برآوردی از **میانگین** تأخیر شبکه هنگام دریافت بسته‌ی i باشد، آن‌گاه این برآورد براساس رابطه‌ی (۷-۱) با بهره‌گیری از برچسب زمانی بسته تعریف می‌شود:

$$d_i = (1-u)d_{i-1} + u(r_i - t_i) \qquad (۷-۱)$$

در این رابطه u یک ارزش ثابت مانند $u = 0.01$ می‌باشد. به عبارت دیگر d_i میانگین تأخیرهای یکنواخت شبکه در قالب $r_1-t_1,\ldots,r_i-t_i$ می‌باشد. برآورد محاسبه شده بیش‌ترین وزن خود را بر تأخیرهای ملاحظه شده‌ی شبکه قرار می‌دهد تا آن‌چه در گذشته‌ی دور ملاحظه نموده است. مشابه این رویکرد برآورد زمان دوره‌ی رفت و برگشت در TCP است که در فصل ۳ بررسی شد. اگر u_i برآوردی از انحراف میانگین تأخیر نسبت به یک میانگین تأخیر محاسبه شده باشد، آنگاه این تأخیر نیز برمبنای رابطه‌ی (۷-۲) با بهره‌گیری از برچسب زمانی چنین محاسبه می‌شود:

$$u_i = (1-u)v_{i-1} + u|r_i - r_i - d_i| \qquad (۷-۲)$$

بدیهی است برآوردهای d_i و v_i برای هر بسته‌ی دریافتی محاسبه می‌شود، اگرچه از آن‌ها فقط به منظور تعیین نقطه‌ی بازپخش نخستین بسته در هر گروه بسته‌ی مکالمه استفاده می‌شود. هنگامی که این برآوردها محاسبه شوند، گیرنده از الگوریتم ذیل جهت بازپخش بسته‌ها بهره می‌گیرد. اگر بسته‌ی i نخستین از گروه بسته‌های مکالمه باشد، بازپخش آن زمانی معادل p_i را اختیار می‌کند و این زمان از سوی رابطه‌ی (۷-۳) محاسبه می‌شود:

$$p_i = t_i + d_i + kv_i \qquad (۷-۳)$$

در این رابطه k یک ثابت مثبت مانند k=4 می‌باشد. هدف از عبارت kv_i تنظیم زمان بازپخش کافی در آینده است به گونه‌ای که فقط بخش کوچکی از بسته‌های وارد شده از یک مکالمه در اثر ورود تأخیری مفقود شوند. نقطه‌ی بازپخش برای هر بسته‌ی بعدی برمبنای فاصله‌ی زمانی از نقطه‌ی بازپخش نخستین بسته می‌باشد. به عبارت دیگر اگر:

$$q_i = p_i - t_i \qquad (۷-۴)$$

طول زمان از هنگام تولید نخستین بسته در مکالمه تا هنگام بازپخش آن می‌باشد و اگر بسته‌ی i نیز متعلق به همان مکالمه باشد آن‌گاه زمان بازپخش آن برمبنای رابطه‌ی (۷-۵) تعریف می‌شود:

$$p_j = t_j + q_j \qquad (۷-۵)$$

الگوریتم توصیف شده مشروط بر آن که گیرنده قادر به تشخیص نخستین بسته در بین گروه بسته‌های مکالمه باشد از مفهوم کاملی برخوردار است. در صورت عدم فقدان بسته آن‌گاه گیرنده قادر به تعیین این موضوع خواهد بود که آیا بسته‌ی i نخستین بسته است. این مهم از طریق مقایسه‌ی برچسب زمانی بسته‌ی i با معادل آن برای بسته‌ی با برچسب (i-1) انجام می‌گیرد. اگر اندازه‌ی: $t_i - t_{i-1} > 20$ میلی ثانیه باشد آنگاه گیرنده تشخیص می‌دهد که بسته‌ی i آغازگر یک گروه بسته در یک بخش از مکالمه است. حالت دیگرهنگامی است که اتلاف بسته به‌گونه‌ی نامنظم و در پاره‌ای اوقات

صورت می‌پذیرد. در این حالت دو بسته‌ی دریافتی پی‌درپی در مقصد ممکن است برچسب‌های زمانی متفاوتی را اختیار نمایند که فاصله‌ی آن بیش از 20 میلی ثانیه می‌باشد و هر دو بسته نیز متعلق به یک گروه بسته در بخشی از مکالمه می‌باشند. در این شرایط است که بهره‌گیری از شماره‌ی ردیف بسیار سودمند می‌باشد. گیرنده با بهره‌گیری از شماره‌ی ردیف تعیین می‌کند آیا تفاوت بیش از 20 میلی ثانیه در برچسب زمان به علت وجود یک بخش جدید از مکالمه است و یا اتلاف بسته‌ها این زمان طولانی‌تر از 20 میلی ثانیه را سبب شده است.

در انتهای این بخش به گفتاری کوتاه در ارتباط با پخش سیال صدا و تصویر ذخیره سازی شده می‌پردازیم. کاربردهای این گونه از ویژگی‌های شماره ردیف، برچسب زمان و تأخیر بازپخش جهت کاهش و حتی حذف تأثیر ناپایداری شبکه بهره می‌گیرند. تفاوت مهم بین تعامل بی‌درنگ صدا و تصویر و پخش سیال صدا و تصویر ذخیره سازی شده آن است که پخش سیال قادر به تحمل تأخیرهای طولانی‌تر نسبت به تعامل بی‌درنگ می‌باشد. هنگامی که یک کاربر یک قطعه‌ی صدا و تصویر را درخواست می‌کند برای وی انتظار 5 ثانیه و یا اندکی بیش‌تر قبل از آغاز بازپخش آن قطعه قابل تحمل و پذیرش می‌باشد و اغلب کاربران تأخیرهای مشابه را می‌توانند پس از وقوع یک تعامل مانند پرش موقتی در هنگام پخش سیال تحمل نمایند. این موضوع سبب انعطاف‌پذیری برای طراحان جهت کاربردهای چندرسانه‌ای ذخیره شده می‌باشد.

۳- بازیابی تأثیر فقدان بسته

برخورد یک کاربرد تلفنی اینترنتی در برابر ناپایداری بسته‌ها را با جزئیات آن در مطالب گذشته تحلیل نمودیم. اکنون به اختصار چندین روش به منظور دستیابی به یک کیفیت مطلوب صدا در حضور عوامل اتلاف بسته تحلیل می‌شود. این گونه روش‌ها معروف به **رویکردهای بازیابی فقدان بسته** می‌باشند. در این زیربخش اتلاف بسته از یک دیدگاه گسترده بررسی می‌شود. فقدان بسته ممکن است به علت عدم تحویل آن در گیرنده باشد و یا آن که بسته پس از گذشت زمان معین و تعریف شده‌ای جهت بازپخش وارد گیرنده شود و در این صورت گیرنده نسبت به حذف آن اقدام نماید. از مثال کاربرد تلفن اینترنتی در چارچوب روش‌های بازیابی فقدان بسته بهره‌گیری می‌شود. در آغاز این بخش اشاره شد انتقال مجدد بسته‌ها در اغلب موارد در یک کاربرد تعامل‌گرای بی‌درنگ مانند تلفن اینترنتی مطلوب نمی‌باشد. یقیناً انتقال مجدد یک بسته که زمان تعریف شده‌ی باز پخش آن گذشته باشد هیچ گونه بهره‌ای را از نقطه نظر هدف مطلوب آن کاربرد در پی ندارد. از طرف دیگر انتقال مجدد بسته‌ای که در اثر سرریزی صف در یک مسیریاب حذف شده است نیز با سرعت مطلوب انجام‌پذیر نمی‌باشد. به علت رعایت این موارد کاربردهای تلفن اینترنتی غالباً از نوعی روش پیش‌بینی اتلاف بهره می‌گیرند. دو نوع از روش‌های پیش‌بینی فقدان بسته عبارتند از:

- **تصحیح خطا به جلو[1] (FEC)**
- **جایگذاری[2]**

• روش تصحیح خطا به جلو (FEC)

نقطه‌ی اتکا FEC پیوست اطلاعات افزونه به جریان بسته‌های اصلی می‌باشد. به منظور افزایش سرعت

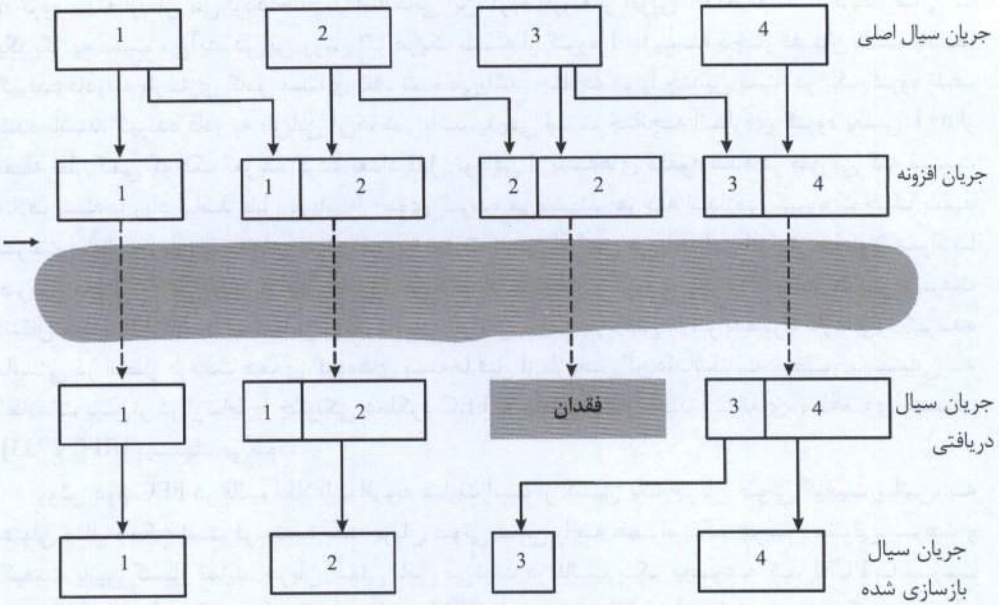
1- Forward error correction
2- Interleaving

انتقال جریان بسته‌های صوتی در این روش از اطلاعات افزونه جهت بازسازی تقریبی و یا واقعی نسخه‌ای از بسته‌های تلف شده بهره‌گیری می‌شود. در ادامه دو رویکرد کاربردی FEC تحلیل می‌شود. در روش نخست مجموعه‌ای از بسته‌های رمزگذاری شده در قالب گروهی به صورت افزونه پس از گسیل n گروه بسته ارسال می‌گردد. ساختار اطلاعاتی این گروه افزونه از طریق XOR تعداد n گروه ارسالی با یکدیگر به دست می‌آید. در این روش اگر هریک بسته از گروه n+1 بسته دچار فقدان شده باشد، گیرنده قادر به بازسازی کامل بسته‌ی تلف شده می‌باشد. چنانچه دو یا چندین بسته در یک گروه تلف شده باشند، گیرنده قادر به بازیابی آن‌ها نمی‌باشد. بدیهی است چنانچه اندازه‌ی گروه یعنی n+1 از نقطه نظر کمی کوچک تعریف گردد تعداد قابل توجهی از بسته‌های مفقود شده در صورتی که نسبت اتلاف بسته‌ها زیاد نباشد قابل بازیابی از سوی گیرنده می‌باشند. هرچه اندازه‌ی گروه کوچک باشد سرعت انتقال جریان بسته‌های صوتی نیز به صورت نسبی افزایش می‌یابد. این افزایش معمولاً همراه با ضریبی معادل $\frac{1}{n}$ می‌باشد به عنوان مثال اگر n تعداد بسته‌ها در گروه و برابر n=3 باشد افزایش سرعت انتقال معادل 33 درصد است. این رویکرد ساده افزایش تأخیر بازپخش را نیز به‌همراه دارد، زیرا گیرنده بایستی در انتظار دریافت همه‌ی گروه‌های بسته‌ها قبل از بازپخش آن‌ها باشد. به منظور دستیابی به اطلاعات بیش‌تر در ارتباط با چگونگی عملکرد FEC به منظور انتقال چند رسانه‌ای، مطالعه‌ی مستند [RFC 2733] پیشنهاد می‌شود.

روش دوم FEC در قالب اطلاعات افزونه عبارت است از گسیل یک جریان صوتی کیفیت پائین، به عنوان مثال ممکن است فرستنده یک جریان صوتی نامی را به همراه یک جریان صوتی سرعت و کیفیت پائین گسیل نماید. جریان صوتی نامی می‌تواند در قالب یک مجموعه کد PCM با سرعت 64 kbps باشد که یک جریان کیفیت پائین GSM با سرعت 13 kbps آن را همراهی می‌کند. در بیان شبکه‌های کامپیوتری جریان سرعت و کیفیت پائین معروف به **جریان افزونه** می‌باشد. در شکل ۷-۶ فرستنده بسته‌ی شماره‌ی n رااز طریق گزینش گروه n ازجریان نامی وسپس پیوست آن به گروه (n-1) از جریان افزونه بازسازی می‌کند. در این رویکرد هرگاه اتلاف بسته‌ها حالت پیاپی نداشته باشد، گیرنده می‌تواند رویداد اتلاف را از طریق بازپخش با سرعت کند گروه‌های بسته که به همراه بسته‌ی مورد نظر وارد می‌شوند پنهان نماید. بدیهی است گروه‌های کند سرعت از کیفیت نسبتاً اندکی در مقایسه با گروه‌های نامی برخوردار می‌باشند. یک گروه سیال با کیفیت بالا چنانچه در پاره‌ای از اوقات با گروه‌های از لحاظ کیفی پائین و بدون فقدان بسته همراه گردد در حالت کلی کیفیت صوتی بهینه‌ای را به ارمغان می‌آورد. بدیهی است در این روش گیرنده فقط کافی است دو بسته را قبل از بازپخش فایل صوتی دریافت نماید تا افزایش اندکی در زمان تأخیر بازپخش ایجاد نماید. علاوه برهمه‌ی موارد اشاره شده چنانچه نرخ رمزگذاری به مراتب کم‌تر از سرعت رمزگذاری نامی باشد آن‌گاه افزایش نسبی سرعت انتقال اندک خواهد شد.

به منظور مقابله با رخداد اتلاف پی‌درپی بسته‌ها می‌توان از یک رویکرد تغییری ساده بهره گرفت. به جای پیوست گروه بسته‌ی کند سرعت شماره‌ی (n-1) به گروه نامی پرسرعت شماره‌ی n فرستنده می‌تواند دو گروه بسته‌ی کند سرعت (n-1) و (n-2) را و یا آنکه دو گروه کم سرعت (n-1) و (n-3) و غیره را به یکدیگر پیوست نماید. با افزودن گروه‌های کند سرعت بیش‌تر به هر گروه، بسته‌ی نامی دارای سرعت بالا و کیفیت صدای دریافتی درگیرنده با انواع گوناگونی از محیط‌های دشوار که از رویکرد برمبنای بهترین تلاش استفاده می‌کنند به مراتب بهتر از شرایط دیگر می‌باشد. از طرف دیگر افزودن گروه‌های بیش‌تر، سرعت انتقال و زمان تأخیر بازپخش صدا را نیز افزایش می‌دهد. یکی از کاربردهای

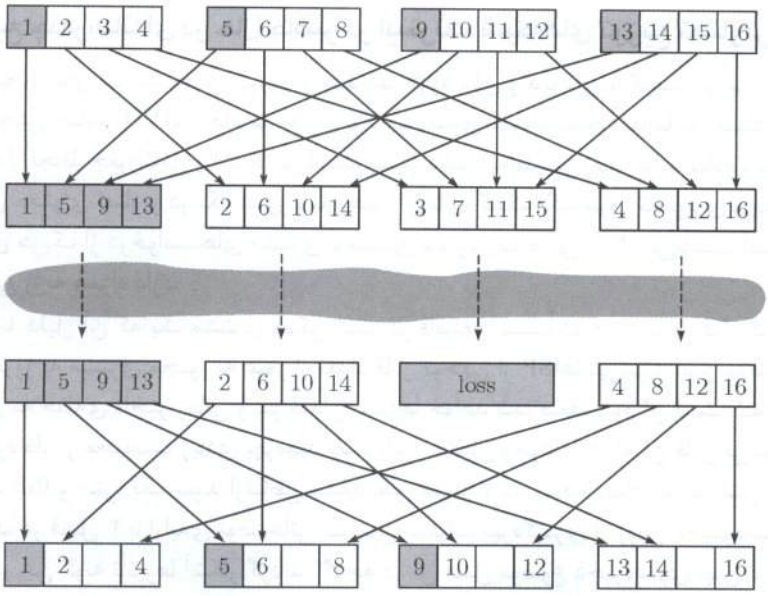
معروف تلفن اینترنتی که از روش FEC بهره گرفته است پروتکل RAT می‌باشد. این پروتکل قادر به انتقال جریان‌های صوتی با کیفیت پائین به همراه جریان صوتی نامی براساس موارد اشاره شده در بالا می‌باشد.



شکل ۷-۶ ♦ پیوست اطلاعات افزونه‌ی کیفیت پائین به اطلاعات اصلی

• روش جایگذاری

روش دوم جایگزین انتقال اطلاعات افزونه‌ی یک کاربرد تلفن اینترنتی می‌تواند از جایگذاری بسته‌های صوتی به منظور انجام عملیات بازیابی بسته‌های مفقود شده استفاده نماید. در این رویکرد برمبنای آنچه در شکل ۷-۷ نشان داده شده است فرستنده قبل از انتقال عناصر داده‌ی صوتی آن‌ها را شماره‌گذاری مجدد می‌نماید. انجام این عملیات به این گونه می‌باشد که عناصر اصلی مجاور با بهره‌گیری از یک فاصله‌گذاری ویژه در هنگام انتقال جریان داده‌ی صوتی از یکدیگر مجزا می‌شوند. این جایگذاری فاصله در بین عناصر اصلی مجاور سبب کاهش تأثیر اتلاف بسته‌ها می‌شود. به عنوان مثال اگر عناصر داده طول زمانی معادل 5 ms اختیار نموده‌اند و هر گروه نیز شامل 4 بسته و جمعاً به طول 20 ms باشند، آن‌گاه نخستین گروه می تواند حامل بسته‌های 1، 5، 9، 13 و گروه دوم حامل بسته‌های 2، 6، 10 و 14و غیره باشد. از شکل ۷-۷ ملاحظه می‌شود فقدان تنها یک بسته از یک جریان جایگذاری شده چندین فاصله‌ی کوتاه زمانی را در بازسازی مجدد جریان به همراه دارد. این حالت نقطه‌ی مقابل وضعیتی است که در آن یک فاصله‌ی زمانی طولانی در یک جریان جایگذاری نشده روی می‌دهد.



شکل ۷-۷ ♦ گسیل بسته‌های صوتی جایگذاری شده

عملکرد جایگذاری سبب بهبود چشمگیر کیفیت یک جریان بسته‌ی صوتی و نیز سرآیند اندک می‌گردد. نقطه ضعف بدیهی روش جایگذاری افزایش زمان تأخیر در خارج شدن نتیجه می‌باشد. این ضعف کاربرد آن را در رویکردهای تعاملی زنده مانند تلفن اینترنتی محدود می‌نماید اگرچه عملکرد مطلوبی را در حالت پخش صوتی ذخیره سازی شده به همراه دارد. نقطه‌ی قوت روش جایگذاری عدم افزایش نیاز پهنای باند (سرعت) در یک جریان بسته‌ی صوتی می‌باشد.

- **بازسازی جریان صوتی خسارت دیده در گیرنده**

روش‌های بازسازی برمبنای گیرنده کوشش در تولید جایگزینی برای یک بسته‌ی تلف شده دارد که آن بسته بسیار مشابه بسته‌ی اصلی می‌باشد. انجام این مهم از آن جهت امکان‌پذیر است که سیگنال‌های صوتی و به ویژه مکالمه تعدادانبوهی از شباهت‌های کوتاه مدت از لحاظ زمانی را شامل می‌شوند. این روش‌ها در شرایط نرخ اتلاف اندک یعنی کمتر از 15 درصد و نیز برای بسته‌های با فاصله‌ی زمانی کوتاه بین 4 تا 40 میلی‌ثانیه بسیار مطلوب می‌باشند. هنگامی طول زمان فقدان بسته به طول زمان رخداد یعنی 5 تا 100 میلی ثانیه نزدیک شود این روش‌ها بی‌ثمر خواهد شد زیرا تمامی رخدادها ممکن است از نقطه نظر شنونده مفهوم نباشد.

شاید ساده‌ترین روش بازیابی برمبنای گیرنده تکرار چندین مرتبه‌ای بسته‌ی موردنظر باشد. عملیات تکرار، بسته‌های مفقود شده را با نسخه‌هایی از بسته‌های وارد شده قبل از فقدان جایگزین می‌کند. این رویکرد پیچیدگی محاسباتی اندکی دارد و نسبتاً کارایی مطلوبی نیز به همراه دارد. روش دیگر بازیابی برمبنای گیرنده در قالب درون‌یابی می‌باشد. در این روش از گروه بسته‌ی صوتی قبل و بعد از رویداد مفقودی بسته به منظور درون‌یابی یک بسته‌ی مناسب برای جایگزین نمودن بسته‌ی تلف شده استفاده می‌شود. روش درون‌یابی در مقایسه با رویکرد تکرار بسته برتری به مراتب بیش‌تری را به‌همراه دارد اما عملیات محاسباتی در آن بسیار پیچیده‌تر می‌باشد.

۷-۶ توزیع چندرسانه‌ای در حال حاضر در اینترنت: شبکه‌های توزیع تصاویر (محتوا)

با توجه به محدوده‌ی سرعت پخش تصویر از چندصد kbps برای ویدیوهای با کیفیت پائین تا چندین Mbps در پخش تصاویر DVD، انجام توزیع ویدیو ذخیره‌سازی شده برمبنای تقاضا به تعداد انبوهی از کاربرانی که از لحاظ جغرافیایی پراکنده می‌باشند، بسیار دشوار به‌نظر می‌رسد. شاید ساده‌ترین رویکرد، ذخیره‌سازی محتوای تصاویر در یک سرور و بازپخش آن‌ها به کمک یک سرور تصویری و یا مزرعه‌ی سرورها برای هریک از درخواست‌های مشتری برمبنای موارد زیربخش ۷-۴ می‌باشد. این روش دو اشکال بدیهی را به همراه دارد.

نخست به دلیل آن که یک مشتری ممکن است در فاصله‌ی نسبتاً دوری از سرور قرار گرفته باشد، بسته‌های سرور به مشتری مجبور به عبور از تعداد قابل توجهی از ISPها می‌شوند این موضوع سبب افزایش قابل ملاحظه‌ی تأخیر زمانی و نیز فقدان بسته‌ها خواهد شد. ضعف دوم آن است که اگر فایل تصویری موردنظر از محبوبیت زیادی بهره‌مند باشد، این احتمال وجود دارد که آن فایل بی‌شمار مرتبه از درون یک ISP و حتی یک پیوند ارتباطی مشابه عبور نموده و در نتیجه پهنای باند قابل توجهی را مصرف نماید. در فصل ۲ با ارایه‌ی توضیحاتی پیرامون عملیات بهره‌گیری از روش ذخیره‌سازی نهان کاهش آن در این گونه تأثیرها آشکار گردید. اگرچه در آن فصل موضوع ذخیره‌سازی نهان در رابطه با محتوای وب سنتی تحلیل گردید اما بدیهی است این رویکرد می‌تواند برای کاربرد در فایل‌های چندرسانه‌ای محتوایی مانند صدا و تصویر نیز سودمند باشد. در این بخش موضوع **شبکه‌های توزیع محتوا**[1] **(CDN)** که رویکرد جایگزین توزیع محتوای چندرسانه‌ای ذخیره‌سازی شده می‌باشد، بررسی و تحلیل می‌شود.

فلسفه‌ی پیاده‌سازی شبکه‌های CDN براین مبنا قرار گرفته است که برنامه‌ی مشتری قادر به دستیابی محتوا نمی‌باشد زیرا مسیر بهترین تلاش از سرور به مشتری عملیات بازپخش تصاویر را پشتیبانی نمی‌کند. در این حالت اگرچه مشتری به سوی محتوا جاری نمی‌شود اما این محتوا است که بایستی به سوی مشتری گسیل گردد. به عبارت دیگر CDN از الگوهای متفاوتی در مقایسه با وب نهان بهره می‌گیرد. برای یک شبکه CDN مشتریان، ISPها نمی‌باشند بلکه تأمین کنندگان فایل‌های محتوایی اهمیت دارند. یک تأمین کننده‌ی محتوا مانند CNN با فایل‌های تصویری جهت توزیع، هزینه‌ی قابل توجهی را به یک شرکت CDN مانند Akamai می‌پردازد تا فایل‌های تصویری آن را با حداقل زمان تأخیر ممکن در دسترس کاربران قرار دهد. چگونگی تامین خدمات توزیع فایل‌های با محتوایی از سوی یک شرکت CDN را می‌توان در قالب ذیل خلاصه نمود:

۱- شرکت CDN اقدام به نصب و راه‌اندازی صدها **سرور CDN** در سراسر شبکه‌ی اینترنت می‌کند. معمولاً مکان قرار گرفتن سرورهای CDN در **مراکز داده**[2] می‌باشد که در اغلب موارد در ISPهای سطوح پائین و یا در مجاورت شبکه‌های دستیابی ISPها و سرانجام مشتریان شبکه می‌باشد.

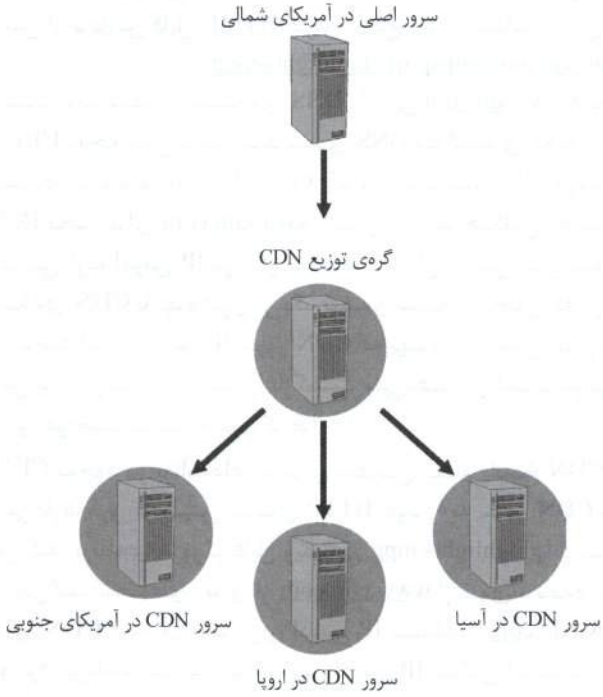
۲- معمولاً فایل‌های محتوای کاربران CDN به صورت نسخه‌های چندگانه در سرورهای CDN بارگذاری می‌شوند. هرگاه یک کاربر محتوا را به روزرسانی می‌کند، شبکه‌ی CDN فایل محتوایی جدید را بین سرورهای CDN مجدداً توزیع می‌کند.

1- Content distribution networks

2- Data centers

۳- از سوی شرکت CDN روش‌هایی فراهم می‌شود که از طریق آن‌ها هنگامی که یک برنامه‌ی مشتری تقاضای یک فایل محتوا را می‌کند، آن فایل از سوی یکی از سرورهای CDN که قادر به تأمین بهترین سرویس به آن مشتری می‌باشد، انجام می‌گیرد. این سرور ممکن است نزدیک‌ترین سرور در همان ISP به مشتری باشد و یا آن که سرور CDN از یک مسیر بدون تراکم به مشتری موردنظر برخوردار می‌باشد.

شکل ۷-۸ تعامل بین دو بخش تأمین کننده‌ی محتوا و شرکت CDN را نشان می‌دهد. برمبنای شکل ۷-۸ یک تأمین کننده مانند CNN ابتدا بایستی در مورد اشیاء موردنظر خویش مانند تصویر و یا غیره که مایل است CDN آن را توزیع نماید، تصمیم‌گیری کند. لازم به توضیح است مابقی عناصر از سوی تأمین کننده‌ی فایل‌های محتوا بدون دخالت CDN توزیع می‌گردد. تأمین کننده با پیوست برچسب، محتوا را به یک گره‌ی CDN تزریق و آن گره نیز محتوا را پس از نسخه برداری به درون سرورهای CDN گزینش شده تزریق می‌کند. شرکت CDN ممکن است مالک یک شبکه‌ی خصوصی به منظور تزریق محتوا از گره‌ی CDN به سرورهای CDN باشد. هرگاه تأمین کننده‌ی محتوا یک عنصر توزیع شده‌ی CDN را تغییر دهد و یا آن را به روزرسانی نماید، نسخه‌ی جدید محتوا جایگزین نسخه‌ی قبلی در گره‌ی CDN می‌شود. گره‌ی CDN نیز بی‌درنگ پس از نسخه برداری آن را بین سرورهای CDN توزیع می‌کند. یادآوری این نکته ضروری است هر سرور CDN حاصل عناصر انبوهی از تأمین کنندگان فایل‌های محتوایی گوناگون می‌باشد.



شکل ۷-۸ ♦ عنصر برچسب‌گذاری شده از سوی **CDN** به سرورهای مربوط به آن تزریق می‌شود.

اکنون این پرسش مهم مطرح می‌شود: چه هنگام یک مرورگر در میزبان کاربر فرمان بازیابی عنصر

ویژه‌ای را که از سوی یک URL مشخص شده است،دریافت می‌دارد؟ چگونه مرورگر به این تصمیم می‌رسد که آیا عنصر را بایستی از سرور اصلی بازیابی نماید و یا آن که از یکی از سرورهای CDN؟ معمولاً شبکه‌های CDN از یک DNS جهت هدایت مرورگرها به سرور مناسب و مطلوب بهره می‌گیرند.

به عنوان یک مثال فرض کنید نام میزبانی یک تأمین‌کننده‌ی فایل‌های محتوا www.foo.com باشد و نام شرکت CDN نیز cdn.com ، علاوه براین موارد فرض کنید تأمین کننده‌ی محتوا فقط مایل به توزیع فایل‌های تصویری mpeg خویش از سوی CDN می‌باشد و همه‌ی عناصر دیگر از جمله صفحه‌های برمبنای HTML مستقیماً از سوی تأمین کننده توزیع می‌شود. به منظور رسیدن به این هدف تأمین کننده‌ی فایل‌های محتوا تغییراتی را در همه‌ی عناصر HTML موجود در سرور اصلی اعمال می‌کند تا آن که آدرس URL فایل‌های تصویری با آدرس: http://www.cdn.com برچسب‌نگاری شوند ودر نتیجه اگر یک فایل HTML در تأمین کننده‌ی محتوا شامل آدرس مراجعه به: http://www/foo/com/sports/highlights.mpeg باشد آن‌گاه تأمین کننده این عنصر را از طریق جایگزینی ارجاع در فایل HTML با آدرس:

http://www/cdn.com/www.foo.com/sports/highlights.mpg برچسب‌نگاری می‌کند. هنگامی که یک مرورگر متقاضی یک صفحه‌ی وب شامل ویدیویی از highlights.mpg باشد آن‌گاه عملیات ذیل انجام خواهد شد:

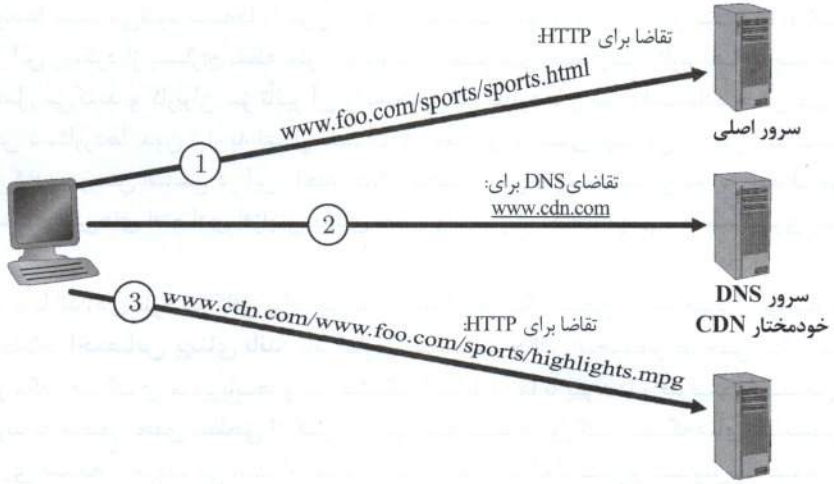
۱- مرورگر درخواست خود برای عنصر برمبنای HTML را به سوی سرور اصلی یعنی: www.foo.com گسیل می‌کند و آن نیز عنصر HTML مورد تقاضا را به مرورگر ارسال می‌دارد. مرورگر پس از پردازش فایل HTML آدرس ارجاع ذیل را شناسایی می‌کند:
http://www.cdn.com/www.foo.com/sports/highlights.mpg

۲- مرورگر سپس یک عملیات جستجوی DNS را برای www.cdn.com انجام می دهد که نام میزبانی URL مرجع می باشد. پیکربندی DNS به‌گونه‌ای انجام شده است که همه‌ی درخواست‌های مربوط به www.cdn.com که وارد یک سرور CDN ریشه می‌شوند را به یک سرور DNS مختار برای www.cdn.com گسیل می‌کند. هنگامی که سرور DNS مختار تقاضا را دریافت می‌دارد، آدرس IP مرورگر پرسشگر را از آن بازیابی می‌کند. در این هنگام سرور DNS شبکه‌ی CDN با بهره‌گیری از یک نقشه‌ی شبکه‌ی داخلی که آن را برای تمامی شبکه‌ی اینترنت ساخته است، آدرس IP سرور CDN که بهترین در میان سرورهای موجود به منظور پاسخگویی به مرورگر می‌باشد را بازگشت می‌دهد. در اغلب مواقع این سرور CDN در نزدیک‌ترین موقعیت نسبت به مرورگر قرار دارد.

۳- کاربرد CDN موجود در برنامه‌های مشتری متقاضی، پیام پاسخ CDN به همراه آدرس IP را دریافت می‌دارد. مرورگر سپس تقاضای HTTP خود را به سرور CDN به همراه آدرس IP آن گسیل می‌کند. برنامه‌ی مرورگر فایل ویدیویی highlights.mpg را از سرور CDN اشاره شده دستیابی می‌کند. تقاضاهای بعدی از www.cdn.com از سوی برنامه‌ی مشتری با بهره‌گیری از همان سرور CDN ادامه می‌یابند، زیرا آدرس IP سامانه‌ی www.cdn.com در حافظه‌ی نهان DNS موجود می‌باشد. بدیهی است این آدرس IP ممکن است در حافظه‌ی نهان میزبان مشتری و یا سرور نام DNS محلی قرار گرفته باشد.

شکل ۷-۹ نشان می‌دهد چگونه میزبان متقاضی ابتدا به وب سرور اصلی جهت دریافت عنصر HTML مراجعه می‌کند و پس از آن به سرور DNS خودمختار شبکه‌ی CDN به منظور دریافت آدرس

IP متعلق به بهترین سرور CDN مراجعه می‌کند و سرانجام نیز جهت دریافت فایل تصویری موردنظر به آن سرور CDN روانه می‌شود. لازم به توضیح است که نیاز به هیچ گونه تغییراتی در HTTP، DNS و یا مرورگر جهت پیاده‌سازی این روش توزیع فایل محتوایی نمی‌باشد.



شکل ۷-۹ ♦ شبکه‌ی **CDN** از سرور **DNS** به منظور هدایت تقاضا به نزدیک‌ترین سرور **CDN** بهره می‌گیرد.

آن‌چه در این مقطع نیاز به تأمل دارد، موضوع چگونگی تعیین **بهترین** سرور CDN برای میزبان متقاضی از سوی شبکه‌ی CDN است. اگرچه هریک از شرکت‌های مالک شبکه‌های CDN از روش‌های ویژه‌ی انحصاری خود جهت انجام این مهم بهره می گیرند اما در هرحال تشخیص چگونگی انجام این مهم چندان دشوار نیست. برای هریک از ISPهای موجود در شبکه‌ی اینترنت که شامل کاربران بالقوه جهت این موضوع می‌باشند معمولاً شرکت‌های CDN سابقه‌ای از بهترین سرورهای CDN را برای آن ISPها نگهداری می‌کنند. شرکت‌های CDN شناسایی بهترین سرور CDN را برمبنای آگاهی از جداول مسیریابی اینترنت و به ویژه جداول BGP موضوع فصل ۴ انجام می‌دهند. علاوه براین موارد از اطلاعات مربوط به محاسبه‌ی زمان رفت و برگشت سیگنال (RTT) و دیگر داده‌های اندازه‌گیری شده از سوی سرورهای آن شبکه CDN به شبکه‌های قابل دستیابی دیگر نیز استفاده می‌کنند. از این طریق شبکه‌ی CDN به محاسبه‌ی خدمات برمبنای بهترین تلاش از سوی یک سرور CDN به ISP موردنظر می‌پردازد. شبکه‌ی CDN این مهم را برای تعداد انبوهی از دستیابی ISPها در شبکه‌ی اینترنت انجام می‌دهد و با بهره‌گیری از این ا طلاعات است که پیکربندی سرور DNS مختار خود را انجام می‌دهد.

۷-۷ ابعادگذاری شبکه‌های بهترین تلاش به منظور تأمین کیفیت خدمات (QoS)

در بخش‌های قبل ملاحظه شد چگونه روش‌های مختلف مانند بازپخش بسته‌های چندرسانه، FEC، جایگذاری بسته در میزبان‌ها و نیز یک زیرساخت CDN می‌تواند کاربردهای چندرسانه‌ای را از لحاظ کیفی در شبکه‌ی اینترنت امروز که برمبنای بهترین تلاش می باشد، بهینه‌سازی نماید. دشواری بهینه‌سازی کیفیت در کاربردهای چندرسانه به علت چالش‌های موجود در کارایی آن‌ها یعنی تأخیر بسته در دو انتهای مسیر، تأخیر ناپایداری بسته و اتلاف بسته‌ها می‌باشد. بدیهی است همه‌ی این موارد

هنگامی روی می‌دهد که شبکه دچار تراکم گردد. سرانجام به منظور افزایش کیفیت کاربردهای چندرسانه‌ای رویکردی که اغلب جهت برطرف نمودن تقریباً هر چالشی که در آن منابع با محدودیت روبه‌رو می‌باشند اعمال می‌شود. افزایش ظرفیت پهنای باند پیوندها در سراسر شبکه است تا از تراکم شبکه و تأخیر و اتلاف بسته‌ها که نتیجه‌ی مستقیم آن می‌باشند تقریباً هرگز امکان‌پذیر نگردد. ظرفیت کافی پیوندها سبب می‌شود بسته‌ها را بتوان از درون اینترنت امروزی بدون تأخیر صف و یا اتلاف بسته عبور داد. این رویکرد از بسیاری نقطه نظرها مطلوب به نظر می‌رسد زیرا کاربردهای چندرسانه‌ای ایده‌آل عمل می‌کنند و کاربران نیز تأثیر آن را به خوبی در کاربردهای مورد استفاده حس می‌کنند. همه‌ی این دستاوردها بدون نیاز به اعمال تغییرات در معماری برمبنای بهترین تلاش اینترنت فعلی میّسر می‌گردد. پرسش اساسی در این رابطه مقدار ظرفیت کافی جهت دستیابی به این اهداف می‌باشد. بدیهی است چالش‌های اقتصادی افزایش پهنای باند پیوند نیز از نکات مهم تصمیم‌گیری به‌شمار می‌آید.

مقدار و یا اندازه‌ی ظرفیت کافی برای پیوندهای شبکه در یک سطح از عملکرد انتها به انتهای شبکه اصطلاحاً **اختصاص پهنای باند**[1] نام گذاری شده است. چالش پیچیده‌تر طراحی یک شبکه از نقطه نظر مکان قرارگیری مسیریاب‌ها و نیز چگونگی ارتباط آن‌ها با پیوندها و ظرفیت اختصاص یافته به هر پیوند به منظور تحقق سطحی از کارایی انتها به انتها در واژگان شبکه‌های کامپیوتری به **ابعادگذاری شبکه**[2] معروف می‌باشد. اختصاص پهنای باند و نیز ابعادگذاری عناوین پیچیده در یک شبکه به شمار می‌آیند که بررسی آن‌ها از چارچوب این اثر خارج می‌باشد. در ذیل موارد مهم پیش‌بینی کارایی مطلوب در سطح برنامه‌های کاربردی بین دو نقطه‌ی انتهایی شبکه و در نتیجه‌ی آن اختصاص ظرفیت کافی جهت نیاز یک کاربرد ویژه تشریح می‌گردد:

- **الگوهای تقاضای ترافیک بین نقاط انتهایی شبکه.** در دو سطح تقاضا یعنی ورود کاربران به شبکه به هدف بهره‌گیری از کاربردهای انتها به انتها و در سطح بسته یعنی تولید بسته‌ها از سوی کاربردهای در حال اجرا هر دو نیازمند اختصاص الگوهای مرتبط است. بدیهی است حجم بار ترافیک شبکه ممکن است در طول زمان تغییر نماید.
- **نیازهای یک کارایی تعریف شده‌ی بهینه.** یک کارایی به منظور پشتیبانی ترافیک حساس در برابر تأخیر مانند کاربردهای صوتی و تصویری بی‌درنگ ممکن است از نقطه نظر بهینه‌سازی نیازمند احتمال تأخیر انتها به انتها برای یک کاربرد به‌گونه‌ای باشد که مقدار آن بزرگ‌تر از حداکثر تأخیر قابل قبول و کوچک‌تر از یک مقدار معین e باشد.
- **الگوی پیش‌بینی کارایی انتها به انتها برای یک حجم بار.** الگوهای پیش‌بینی کارایی در این حالت و نیز روش‌های شناسایی حداقل هزینه‌ی اختصاص پهنای باند در شبکه که انتظارهای همه‌ی کاربران را تأمین نماید مربوط به روش‌های صف‌بندی می‌شود که یکی از موضوعات بررسی شده در فصل ۱ بود. در این رابطه پژوهشگران مشغول توسعه و تکمیل الگوهایی به منظور صف‌بندی می‌باشند که هدف آن تأمین عنوان اشاره شده در بالا می‌باشد.

با توجه به رویکرد بهترین تلاش اینترنت پشتیبانی از کارایی بهینه وابستگی نزدیکی به عوامل

1- Bandwidth provisioning
2- Network dimensioning

تجاری و سازماندهی فنی این کاربردها دارد. از نقطه نظر تجاری آمادگی کاربران جهت پرداخت هزینه‌ی پشتیبانی از پهنای باند بیش‌تر مهم است و از لحاظ سازماندهی توافق ISPها در ایجاد مسیرهای انتها به انتها که کاربردهای چندرسانه‌ای را پشتیبانی می‌کنند دارای اهمیت می‌باشند.

۸-۷ پروتکل‌های کاربردهای تعامل‌گرا و بی‌درنگ

کاربردهای بی‌درنگ تعامل‌گرا مانند تلفن اینترنتی و کنفرانس ویدیویی شاهد رشد فزاینده‌ای می‌باشند. بنابراین تعجبی ندارد که نهادهایی مانند IETF و ITU از مدت‌ها پیش و تاکنون در حال تدوین استانداردهای لازم در این زمینه می‌باشند. با تدوین استانداردهای مناسب به منظور کاربردهای بی‌درنگ تعامل‌گرا بخش خصوصی نیز قادر به طراحی و تولید تجهیزات و ابزارهای سخت افزاری و نرم افزاری لازم می‌باشد. در این بخش پروتکل‌های مربوط به کاربردهای بی‌درنگ تعاملی مانند RTP، SIP و H323 بررسی شده‌اند. این مجموعه از استانداردهای تولیدات سخت و نرم‌افزاری گسترده‌ای را در حال حاضر شامل می‌شوند.

۱- پروتکل RTP

در بخش قبل ملاحظه شد فرستنده‌ی یک کاربرد چندرسانه‌ای دامنه‌های سرپیام را به گروه (قطعه) بسته‌های صوتی و یا تصویری قبل از انتقال آن‌ها از درون لایه‌ی شبکه پیوست می‌کند. این دامنه‌های سرپیام از جمله شامل شماره‌های ردیف و برچسب‌های زمانی می‌باشند. از آن‌جائیکه اغلب کاربردهای چندرسانه‌ای شبکه از این دودامنه‌ی سرپیام استفاده می‌کنند بنابراین یک ساختار استاندارد شده‌ی بسته که شامل دامنه‌های داده‌ی صدا و یا تصویر، شماره ردیف و برچسب زمانی است به همراه دیگر دامنه‌های سودمند الزامی می‌باشد. پروتکل RTP در مستند [RFC 3550] تعریف شده است. از این پروتکل می‌توان جهت انتقال قالب‌بندی‌های متداولی مانند PCM، GSM و MP3 برای صدا MPEG و H.263 به منظور تصویر بهره گرفت. این پروتکل قادر به انتقال فایل‌های صدا و تصویر بخش خصوصی با پروتکل‌های ویژه نیز می‌باشد. از همه مهمتر این پروتکل به عنوان مکمل پروتکل‌های بی‌درنگ تعامل‌گرای دیگر مانند SIP و H.323 نیز می‌تواند به کار گرفته شود. در این بخش گفتاری پیرامون RTP و پروتکل همراه آن RTCP ارایه می‌شود.

• اصول پروتکل RTP

اجرای پروتکل RTP بر روی UDP می‌باشد در سمت فرستنده گروه بسته‌ی متعلق به یک رسانه در قالب یک بسته‌ی RTP بسته‌بندی و پس از آن تبدیل به یک قطعه‌ی UDP می‌شود و آن‌گاه به پروتکل IP تحویل می‌گردد. در سمت دریافت بسته RTP از UDP جداسازی و به دنبال آن نیز گروه بسته‌های رسانه از بسته‌ی RTP بازیابی و آن‌گاه تحویل برنامه‌ی بازپخش رسانه می‌گردد تا رمزگشایی و بازیابی شود.

به عنوان مثال حالتی را درنظر بگیرید که از RTP جهت انتقال صدا استفاده می‌شود. اگر فرض شود رمزگذاری منبع صدا به صورت کدهای PCM با سرعت 64 kbps است و اگر برنامه‌ی کاربرد داده‌های رمزگذاری شده را در قالب گروه‌های با طول زمانی 20ms جمع آوری نماید به عبارت دیگر تعداد 160 بایت در هر گروه قرار گیرد. فرستنده‌ی RTP هر گروه داده‌ی صوتی را به همراه یک **سرپیام RTP** شامل نوع رمزگذاری صدا، شماره‌ی ردیف و یک برچسب زمانی همراه می‌کند. معمولاً هر سرپیام RTP

شامل 12 بایت می‌باشد. گروه بسته‌ی صوتی به همراه سرپیام RTP تشکیل یک **بسته‌ی RTP** را می‌دهد. بسته‌ی RTP سپس وارد سوکت واسط UDP می‌شود. در طرف گیرنده، برنامه‌ی کاربردی بسته‌ی RTP را از واسط سوکت خویش دریافت می‌دارد و گروه بسته‌ی صوتی را از بسته‌ی RTP بازیابی نموده و دامنه‌های سرپیام بسته‌ی RTP را به منظور بازگشایی و بازپخش بسته‌ی صوتی مورد استفاده قرار می‌دهد.

در صورتی که یک کاربرد از RTP به جای یک روش خصوصی جهت نوع بازپخش، شماره ردیف و یا برچسب زمانی بهره گیرد آن‌گاه کاربرد با سهولت بیش‌تری قابلیت ترکیب با دیگر کاربردهای چندرسانه‌ای شبکه را خواهد داشت. به عنوان مثال اگر دو شرکت مختلف، نرم افزاری را به منظور کاربرد تلفن اینترنتی طراحی و تکمیل نمایند و هر دو از پروتکل RTP در محصول خود استفاده نمایند، کاربری که یکی از این دو محصول را مورد استفاده قرار می‌دهد قادر به ارتباط با کاربری است که از محصول دیگر تلفن اینترنتی استفاده می‌کند. در بخش ۷-۱۰ ملاحظه خواهد شد که RTP در اغلب موارد موازی با استانداردهای تلفن اینترنتی مورد استفاده قرار می‌گیرد.

تأکید این نکته نیز ضروری است که RTP هیچ گونه مکانیزمی را جهت اطمینان از تحویل به موقع داده و یا ضمانت QoS فراهم نمی‌آورد. حتی ضمانت تحویل بسته‌ها و یا جلوگیری از عدم تحویل خارج از نوبت بسته‌ها را نیز نمی‌کند. آن‌چه از تأثیر بسته‌بندی RTP قابل ملاحظه می‌باشد فقط در سیستم‌های انتهایی است. مسیریاب‌ها بین داده گرام‌های IP که بسته‌های RTP را حمل می‌کنند و داده گرام‌های IP که حامل آن‌ها نمی‌باشند تمایزی نمی‌گذارند.

پروتکل RTP اجازه می‌دهد هر منبع مانند یک دوربین و یا میکروفن بسته‌های سیال RTP مستقل خویش را اختصاص دهد. به عنوان مثال در یک کنفرانس ویدیویی بین دو کاربر تعداد 4 جریان RTP ممکن است تعریف شود، 2 جریان برای انتقال صدا هریک در یک جهت ارسال و 2 جریان برای انتقال تصویر بازهم هرکدام به منظور یک جهت. اغلب روش‌های رمزگذاری مانند MPEG1 و MPEG2 بسته‌های صدا و تصویر را در قالب یک جریان در هنگام فرآیند رمزگذاری تجمیع می‌کنند. هنگامی که تجمیع صدا و تصویر از سوی رمزگذار انجام می‌شود فقط یک جریان RTP در هر سوی انتقال تولید می‌شود.

بسته‌های RTP محدود به کاربردهای تک پخش نمی‌شوند، آن‌ها را می‌توان از طریق یک به تعداد انبوهی و انبوهی به انبوهی درخت چندپخش گسیل نمود. در حالت چندپخش انبوه به انبوه همه‌ی فرستنده‌ها و منابع از یک گروه چندپخش مشابه جهت ارسال جریان‌های RTP بهره می‌گیرند. جریان‌های RTP چندپخش متعلق به یکدیگر می‌باشند، مانند جریان‌های صدا و تصویر که از سوی چندین فرستنده در یک کاربرد کنفرانس تصویری متعلق به یک **تعامل RTP** سرچشمه می‌گیرند.

- **دامنه‌های سرپیام بسته‌ی RTP**

شکل ۷-۱۰ چهار دامنه‌ی مهم سرپیام بسته‌ی RTP که عبارتند از نوع محموله، شمارگان ردیف، برچسب زمان و شناسه منبع را نشان می‌دهد. دامنه‌ی نوع محموله در بسته‌ی RTP طولی معادل 7 بیت را شامل می‌شود. برای یک جریان صوتی از نوع محموله به منظور تعریف و مشخص نمودن رمزگذاری صدا مانند PCM، مدولاسیون خودسازگار دلتا، رمزگذاری تخمین خطی و غیره استفاده می‌شود. اگر یک فرستنده هنگامی تصمیم به تغییر رمزگذاری بگیرد که یک تعامل جاری در حال انجام است، فرستنده می‌تواند گیرنده را از تغییر انجام شده با بهره‌گیری از دامنه‌ی نوع محموله آگاه نماید.

یک فرستنده ممکن است تغییر رمزگذاری را به منظور افزایش کیفیت صدا و یا کاهش نرخ جریان بیت RTP طلب نماید. جدول ۷-۲ فهرستی از انواع محموله‌های صوتی که در حال حاضر از سوی RTP پشتیبانی می‌شود را نشان می‌دهد.

برای یک جریان تصویری، دامنه‌ی نوع محموله به مفهوم چگونگی رمزگذاری تصویر مانند JPEG، MPEG1، MPEG2 و یا H.261 و غیره می‌باشد. مانند جریان صدا در این مقطع نیز فرستنده می‌تواند نوع رمزگذاری ویدیویی را هنگامی که جریان تصویری در حال انتقال است تغییر دهد. جدول ۷-۳ فهرستی از برخی از انواع محموله‌های تصویری که در حال حاضر پروتکل RTP آن‌ها را پشتیبانی می‌کند نشان می‌دهد. از جمله دامنه‌های مهم دیگر می‌توان از موارد ذیل نام برد:

دامنه‌های متفرقه	شناسه‌ی همزمانی مبدا	برچسب زمان	شمارگان ردیف	نوع محموله

شکل ۷-۱۰ ♦ دامنه‌های سرپیام RTP

- **دامنه‌ی شمارگان ردیف.** این دامنه طولی معادل 16 بیت را اختیار می‌کند. ارزش درون این دامنه برای هر بسته‌ی RTP ارسال شده به مقدار 1 افزایش می‌یابد. از اطلاعات درون این دامنه گیرنده ممکن است به منظور تشخیص افت بسته و بازیابی نظم بسته‌ها بهره گیرد. به عنوان مثال اگر در سمت گیرنده برنامه‌ی کاربردی جریانی از بسته‌های RTP را دریافت نماید و در بین آن یک فاصله از لحاظ شمارگان ردیف مانند 86 و 89 وجود داشته باشد، آن‌گاه گیرنده می‌داند که دو بسته‌ی 87 و 88 مفقود شده‌اند. در این هنگام گیرنده می‌تواند جهت پنهان نمودن داده‌ی از دست رفته تلاش نماید.
- **دامنه‌ی برچسب زمان.** طول این دامنه برابر 32 بیت می‌باشد. اطلاعات درون این دامنه به مفهوم لحظه‌ی نمونه‌گیری نخستین بایت در بسته‌ی داده‌ی RTP است. در بخش قبل ملاحظه شد گیرنده از برچسب زمان به منظور حذف ناپایداری بسته که در شبکه ایجاد می‌گردد استفاده می‌کند، علاوه بر این موضوع گیرنده می‌تواند از این برچسب به منظور تأمین همزمانی بازپخش رسانه بهره گیرد. ارزش برچسب زمان از سرعت نمونه‌گیری ساعت در فرستنده مقداردهی می‌گردد. به عنوان مثال برای جریان‌های صوتی ساعت برچسب زمان برای هر فاصله‌ی زمانی نمونه‌برداری به مقدار یک افزایش می‌یابد. به عنوان نمونه در PCM معمولی سرعت ساعت نمونه گیری معادل 8KHz است به عبارت دیگر فاصله‌ی زمانی بین هر نمونه‌برداری 125μsec است. در نتیجه ارزش برچسب زمان هر 125μs یک بار به مقدار یک افزایش می‌یابد. اگر کاربرد صوتی گروه بسته‌های به تعداد 160 نمونه‌ی رمزگذاری شده را تولید نماید، آن‌گاه ارزش برچسب زمان به اندازه‌ی مقدار 160 برای هر بسته‌ی RTP هنگامی که منبع فعال است، افزایش می‌یابد. ساعت برچسب زمان این افزایش مقدار را با نرخ ثابت ادامه می‌دهد حتی اگر منبع غیرفعال باشد.
- **شناسه همزمانی منبع[1] (SSRC).** دامنه‌ی SSRC طول معادل 32 بیت را اختیار می‌کند. از این دامنه به منظور شناسایی منبع جریان RTP استفاده می‌شود. هر جریان در یک

1- Synchronizatain source identifier

تعامل RTP دارای یک SSRC مجزا می‌باشد. دامنه‌ی SSRC آدرس IP فرستنده نمی‌باشد اما ارزشی است که مبدأ به گونه‌ی تصادفی هنگامی اختصاص می‌دهد که جریان جدیدی آغاز شده است. احتمال آن که دو جریان به SSRC مشابهی اختصاص پیدا کند بسیار اندک می‌باشد. با وقوع چنین رویدادی هر دو منبع یک ارزش جدید SSRC را اختیار می‌نمایند.

جدول ۷-۲ ♦ انواع محموله‌های صوتی پشتیبانی شده از سوی **RTP**

شمارگان نوع محموله	قالب صدا	سرعت نمونه‌گیری	سرعت انتقال داده
0	PCM μ -law	8KHz	64 kbps
1	1016	8KHz	4.8 kbps
3	GSM	8KHz	13 kbps
7	LPC	8KHz	2.4 kbps
9	G.722	16KHz	48-64 kbps
14	MPEG Audio	90KHz	-
15	G.728	8KHz	16 kbps

- **طراحی نرم افزارهای کاربردی با RTP**

به منظور طراحی یک کاربرد شبکه برمبنای RTP از دو رویکرد استفاده می‌شود. نخستین رویکرد مربوط به طراح کاربرد می‌باشد که بایستی RTP را از طریق کدهای برنامه‌نویسی که قالب‌بندی RTP را در سمت فرستنده و سپس بازگشایی آن را در گیرنده انجام می‌دهد، درون برنامه‌ی کاربردی اعمال نماید. رویکرد دوم برای طراح بهره‌گیری از توابع کتابخانه‌ای در برنامه‌نویسی C و کلاس‌های جاوا در برنامه‌نویسی Java می‌باشد. این توابع و کلاس‌ها عملیات قالب‌بندی و بازگشایی را برای برنامه‌ی کاربردی انجام می‌دهند. از آن‌جائیکه ممکن است مایل به طراحی نخستین برنامه‌ی کاربردی چندرسانه‌ای با بهره‌گیری از RTP باشید. به تحلیل عمیق‌تر این دو رویکرد می‌پردازیم. این بررسی در چارچوب ارتباط تک پخش به جای چند پخش انجام می‌شود.

یادآوری می‌شود در فصل ۲ اشاره گردید که کاربرد UDPAPI از پردازش ارسال کننده می‌خواهد که برای هر قطعه‌ی UDP که ارسال می‌دارد، آدرس IP مقصد وشماره پورت مقصد را قبل از تزریق بسته به درون سوکت UDP تعریف و تنظیم نماید. پس از آن قطعه‌ی UDP می‌تواند در درون شبکه‌ی اینترنت حرکت نماید و اگر قطعه به دلایل گوناگون مانند سرریزی بافر مسیریاب مفقود گردد سرانجام وارد سوکت دریچه‌ی مربوطه در پردازش دریافت کننده برای آن کاربرد می‌شود. بدیهی است سوکت دریچه از سوی آدرس IP مقصد و شماره پورت مقصد به گونه‌ای کامل آدرس‌دهی می‌شود در حقیقت هر داده‌گرام IP که حامل چنین آدرس IP و شماره پورت مقصد می‌باشد به دریچه‌ی UDP فرآیند دریافت هدایت می‌شود. یادآوری این نکته از فصل ۲ نیز ضروری است که UDPAPI هم‌چنین این امکان را در اختیار طراح برنامه کاربردی می‌گذارد که شماره پورت مبدأ UDP را تعریف نماید اگرچه این ارزش هیچ گونه تأثیری برروی شناسایی فرآیندی که قطعه به آن گسیل می‌شود را به دنبال نخواهد داشت. توجه به این نکته ضروری است، پروتکل RTP شماره پورت ویژه و اختصاصی را نیاز ندارد و بنابراین هنگامی که طراح یک کاربرد RTP را خلق می‌کند این مهم برعهده‌ی وی می باشد که شمارگان پورت دو سوی کاربرد را مشخص و تعریف نماید.

جدول ۷-۳ ♦ برخی از انواع محموله‌های تصویری پشتیبانی شده از سوی **RTP**

شمارگان نوع محموله	قالب تصویر (ویدیو)
26	JPEG متحرک
31	H.261
32	تصویر MPEG 1
33	تصویر MPEG2

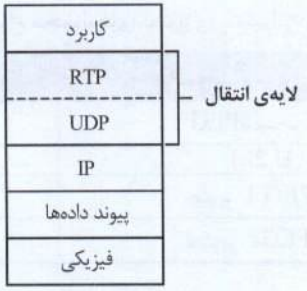
اکنون تلاش نمائید یک برنامه‌ی سرور RTP را طراحی نمائید که فریم‌های تصویری ذخیره‌سازی شده را در قالب بسته‌های RTP بسته‌بندی نماید. این برنامه را به‌گونه‌ای طراحی نمائید که کاربرد شما یک فریم تصویر را گزینش و سپس سرپیام‌های RTP را به آن اضافه نماید تا یک بسته‌ی RTP تولید شود، سپس قاب RTP را به درون سوکت UDP گسیل نماید. برای تحقق این مهم نیازمند تعریف مکان‌هایی به منظور نگهداری انواع سرپیام‌های RTP مانند دامنه‌های شمارگان ردیف و برچسب زمان می‌باشید. هر بسته‌ی RTP خلق شده بایستی قادر به تعریف ارزش شمارگان ردیف و برچسب زمان مناسب خود باشد و همه‌ی عملیات RTP را در سمت فرستنده‌ی کاربرد تعریف نماید. برمبنای شکل ۷-۱۱ برنامه‌ی API طراحی شده در شبکه برپایه‌ی استاندارد سوکت API در پروتکل UDP می‌باشد.



شکل ۷-۱۱ ♦ پروتکل **RTP** به عنوان بخشی از کاربرد و مکان آن در بالای سوکت **UDP** می‌باشد.

رویکرد جایگزین بهره گیری از کلاس RTP جاوا و یا کتابخانه RTP در زبان برنامه‌نویسی C می‌باشد که عملیات RTP را پیاده‌سازی می‌کند. این رویکرد در شکل ۷-۱۲ نشان داده شده است. آن‌چه طراح از این شکل استنباط می‌کند آن است که RTP بخشی از لایه‌ی انتقال است. در این رویکرد RTP/UDPAPI مابین دولایه‌ی کاربرد و انتقال قرار می‌گیرد. هنگامی که یک گروه بسته‌ی رسانه‌ای به درون API گسیل می‌گردد، کاربرد سوی ارسال نیازمند تأمین واسط با گروه بسته‌ی رسانه، شماره‌ی نوع محموله، یک SSRC و یک برچسب زمان است که بایستی به همراه شماره پورت مقصد و یک آدرس IP مقصد همراه گردد. لازم به توضیح است برنامه‌ی **چارچوب رسانه‌ی جاوا**[1] **(JMF)** پیاده‌سازی کامل RTP را در درون خود شامل می‌شود.

1- Java Media Framework



شکل ۷-۱۲ ♦ پروتکل RTP را می‌توان زیرلایه‌ای از لایه انتقال به شمار آورد.

۷-۹ پروتکل کنترل RTP

پروتکل کنترل RTP که اختصاراً RTCP خوانده می‌شود در مستند [RFC3550] توصیف شده است. این پروتکل در ارتباط با RTP می‌تواند از سوی یک کاربرد چند رسانه‌ای شبکه مورد استفاده قرار گیرد. در شکل ۷-۱۳ نشان داده شده است که بسته‌های RTCP از طریق هر کاربر در یک تعامل RTP به دیگر کاربران حاضر در آن تعامل با بهره‌گیری از پروتکل چند پخش IP گسیل می‌شوند. برای یک تعامل RTP در اغلب موارد فقط یک آدرس چند پخش وجود دارد و همه‌ی بسته‌های متعلق به تعامل از آن آدرس چند پخش بهره می‌گیرند. ممیزی بسته‌های RTP و RTCP از یکدیگر با بهره‌گیری از شماره درگاه‌های مجزا انجام می‌شود. شماره پورت RTCP از لحاظ ارزش معادل شماره درگاه RTP +1 می‌باشد.

بسته‌های RTCP گروهی بسته‌های صوتی و تصویری را بسته‌بندی نمی‌کنند بلکه بسته‌های RTCP به شکل متناوب ارسال می‌شوند و حامل گزارش‌های فرستنده و یا گیرنده از اطلاعات آماری مفید به کاربردها می‌باشند. این اطلاعات آماری شامل تعداد بسته‌های ارسالی، تعداد بسته‌های مفقود شده و ناپایداری بین ورود بسته‌ها می باشند. مشخصات RTP تأثیر این اطلاعات بازخورد را از سوی کاربردها مشخص نمی‌کند، زیرا این مهم برعهده‌ی طراح یک کاربرد گذارده شده است. فرستنده می‌تواند از اطلاعات بازخورد به منظور اعمال تغییر در سرعت انتقال و یا موارد دیگر استفاده نماید. اطلاعات بازخورد را می‌توان به منظور اهداف عیب‌یابی مانند تشخیص آنکه آیا عیب محلی، منطقه‌ای و جهانی است از سوی گیرنده‌ها به کار گرفت.

۱- انواع بسته‌ی RTCP

برای هر جریان بسته‌ی RTP که یک گیرنده به عنوان بخشی از یک تعامل دریافت می‌دارد، گیرنده یک گزارش دریافت تولید می کند. پس از آن گیرنده گزارش‌های دریافتی را درون یک بسته‌ی RTCP تجمیع می‌کند و بسته به درخت چند پخش که همه‌ی حاضران در تعامل را به یکدیگر ارتباط می‌دهد گسیل می‌شود. گزارش دریافتی شامل چندین دامنه است که مهم‌ترین آن‌ها در ذیل فهرست شده‌اند:

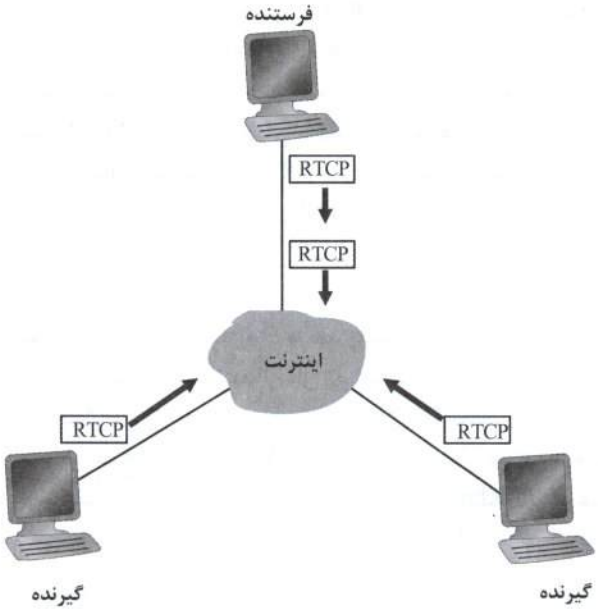
- دامنه‌ی SSRC از جریان RTP که برای آن گزارش دریافتی تولید می‌شود.
- درصد بسته‌های تلف شده در درون جریان RTP، هر گیرنده به عنوان بخشی از جریان مقدار نسبت تعداد بسته‌های RTP مفقود شده را به تعداد کل بسته‌های RTP گسیل شده محاسبه می‌نماید. حاصل این نسبت همان درصد بسته‌های تلف شده است. اگر فرستنده

گزارش‌های دریافتی را برمبنای این اطلاعات دریافت نماید که گیرنده‌ها فقط بخش کوچکی از بسته‌های ارسال شده از سوی فرستنده را دریافت می‌دارند، آن‌گاه فرستنده سرعت رمزگذاری خود را کاهش می‌دهد تا در نتیجه‌ی آن از تراکم شبکه کاسته شده و سرعت دریافت بهبود یابد.

- آخرین شماره ردیف از جریان بسته‌های RTP دریافت می‌شود.
- ناپایداری مابین دریافت که در واقع عبارت است از تخمین یکنواخت تغییرات در زمان بین ورود بسته‌ها در جریان RTP که بسته‌ها در آن حالت پی در پی دارند از جمله‌ی دامنه‌های گزارش دریافتی است.

برای هر جریان RTP که یک فرستنده گسیل می‌کند معمولاً فرستنده به خلق و انتشار بسته‌های گزارش RTCP فرستنده می‌پردازد. این بسته‌ها شامل اطلاعاتی در ارتباط با جریان RTP می‌باشند که عبارتند از:

- دامنه‌ی SSRC از جریان RTP
- برچسب زمان و زمان ساعت دیواری جدیدترین بسته‌ی RTP تولید شده در جریان.
- تعداد بسته‌های ارسالی در جریان
- تعداد بایت‌های ارسال شده در جریان



شکل ۷-۱۳ ♦ ارسال و دریافت پیام‌های RTCP از سوی فرستنده‌ها و گیرنده‌ها

از گزارش‌های فرستنده می‌توان به منظور هم زمانی جریان‌های رسانه‌ای مختلف در درون یک تعامل RTP بهره گرفت. به عنوان مثال یک کاربرد همایش تصویری (کنفرانس تصویری) را درنظر آورید که در آن هر فرستنده دو جریان مستقل RTP را تولید می‌کند، یکی از آن‌ها برای تصویر و دیگری برای صدا می‌باشد. برچسب‌های زمانی در بسته‌های RTP به سرعت نمونه‌برداری صدا و تصویر وابسته می‌باشند و اصطلاحاً وابستگی به **زمان ساعت دیواری** (زمان واقعی) ندارند. گزارش هر

فرستنده‌ی RTCP برای جدیدترین بسته‌ی تولید شده در یک جریان RTP می‌باشد، شامل برچسب زمان بسته‌ی RTP و زمان ساعت دیواری (واقعی) هنگامی که آن بسته خلق شده است. بنابراین بسته‌های گزارش RTCP فرستنده ساعت نمونه‌گیری از پیام را به ساعت واقعی وابسته می‌کند. گیرنده‌ها از این خاصیت وابستگی در گزارش‌های فرستنده‌ی RTCP به منظور هم‌زمانی بازپخش صدا و تصویر استفاده می‌کنند.

برای هر جریان RTP که یک فرستنده ارسال می‌کند بسته‌های توصیف مبدأ را نیز خلق و گسیل می‌نماید. این بسته‌ها شامل اطلاعاتی در رابطه با مبدأ مانند آدرس ایمیل و نام فرستنده و نیز کاربرد خالق آن جریان RTP می‌باشد. هم‌چنین اطلاعات SSRC وابسته به جریان RTP را نیز شامل می‌شود. این بسته‌ها نگاشتی را بین شناسه‌ی مبدأ یعنی SSRC و نام کاربر یا میزبان فراهم می‌کنند. بسته‌های RTCP پشته‌گرا می‌باشند، به بیان دیگر گزارش‌های گیرنده، فرستنده و توصیف‌گران مبدأ را می‌توان در قالب یک بسته تجمیع نمود. بسته‌های حاصل از این یکپارچگی در درون یک قطعه‌ی UDP بسته‌بندی و سپس وارد درخت چندپخش می‌گردد.

۲- مقیاس‌دهی پهنای باند RTCP

یکی از مهم‌ترین اشکال‌های RTCP عدم مقیاس‌پذیری آن می‌باشد. به منظور روشن شدن موضوع یک تعامل RTP را درنظر آورید که شامل یک فرستنده و گروه قابل توجهی از گیرنده‌ها می‌باشد. اگر هریک از گیرنده‌ها به صورت متناوب به تولید بسته‌های RTCP بپردازد، آن‌گاه جمع انباشتی نرخ انتقال بسته‌های RTCP می‌تواند فراتر از نرخ بسته‌های RTP ارسالی از سوی فرستنده باشد. بدیهی است مقدار ترافیک RTP ارسال شده به درون درخت چند پخش با افزایش تعداد گیرنده‌ها تغییر نمی‌کند، در حالیکه مقدار ترافیک RTCP به گونه‌ی خطی با تعداد گیرنده‌ها رشد می‌کند. به منظور برون رفت از چالش مقیاس‌پذیری نرخ ارسال بسته‌های RTCP از سوی فرستنده‌ها به درون درخت چندپخش به صورت تابعی از تعداد حاضران در تعامل از سوی RTCP تعریف می‌شود. از طرف دیگر چون هریک از حاضرین در تعامل بسته‌های کنترل را به دیگران گسیل می‌کند تخمین تعداد حاضران در تعامل از سوی هریک از حاضران میسّر می‌شود.

معمولاً RTCP مقدار ترافیک را محدود به 5 درصد پهنای باند یا همان سرعت تعامل می‌کند. به عنوان مثال اگر فرض شود فقط یک فرستنده ارسال تصاویر در سرعت 2 Mbps را انجام می‌دهد، آن‌گاه RTCP تلاش می‌کند ترافیک آن را محدود به 5 درصد 2 Mbps یعنی 100 Kbps نماید. به عبارت دیگر پروتکل 75 درصد این سرعت یعنی 75Kbps از آن را در اختیار گیرنده‌ها قرار می‌دهد. در نتیجه 25 درصد باقیمانده‌ی سرعت به فرستنده اختصاص می‌یابد. مقدار 75 Kbps سرعت به نسبت مساوی بین گیرنده‌ها تقسیم می‌شود. بنابراین اگر تعداد گیرنده‌ها R باشد آن‌گاه هرگیرنده ترافیک RTCP را با سرعت $\frac{75}{R}$ Kbps دریافت می‌کند و فرستنده نیز ترافیک RTCP را با نرخ 25 Kbps گسیل می‌کند. یک فرستنده و یا گیرنده زمان انتقال بسته‌ی RTCP را به گونه‌ی پویا از طریق محاسبه‌ی میانگین اندازه یا ظرفیت بسته در تمام طول تعامل و تقسیم این مقدار بر سرعت اختصاص یافته، انجام می‌دهد. به اختصار زمان یا دوره‌ی ارسال بسته‌های RTCP برای فرستنده از طریق رابطه‌ی (۷-۶) در ذیل محاسبه می‌شود:

$$T=\frac{\text{تعداد فرستنده‌ها}}{\text{پهنای باند تعامل} \times 0.25 \times 0.05} \times \text{(میانگین اندازه یا طول بسته‌ی RTCP)} \quad (۷-۶)$$

و زمان انتقال بسته‌های RTCP برای یک گیرنده نیز از طریق رابطه‌ی (۷-۷) محاسبه می‌شود:

$$T=\frac{\text{تعداد گیرنده‌ها}}{\text{سرعت تعامل} \times 0.05 \times 0.75} \times \text{(میانگین طول بسته‌ی RTCP)} \quad (۷-۷)$$

۷-۱۰ پروتکل SIP

حالت‌هایی را تصور نمائید که مشغول کار با PC خود می‌باشید و همزمان یک مکالمه‌ی تلفنی نیز از طریق اینترنت وارد PC شما می‌شود. هنگامی که از جای خود بلند می‌شوید و در اطراف اتاق قدم می‌زنید مکالمه‌های تلفنی جدید به صورت خودکار به سوی تبلت شما مسیردهی می‌شوند و هنگامی را نیز تصور نمائید که در حال رانندگی هستید و مکالمه‌های تلفنی خودکار به سوی یک وسیله‌ی اینترنتی در خودرو هدایت می‌شود. در این شرایط محیطی هنگام حضور در یک مکالمه‌ی تلفنی شما می‌توانید به دفترچه تلفنی به منظور انجام یک درخواست مکالمه و دعوت دیگران به حضور در مکالمه‌ی تلفنی جاری نیز دسترسی داشته باشید.

دیگر حاضران ممکن است با PC خود مشغول باشند و یا آنکه سیار و در حال رانندگی و یا قدم زدن باشند. هر کجا باشند از نقطه نظر کاربرد مهم نیست و دعوت شما به گونه‌ای آشکار به سوی آن‌ها هدایت می‌شود. در این شرایط محیطی هنگامی که مشغول بارگیری یک صفحه‌ی خانه از سامانه‌ی اینترنتی می‌باشید پیوندی وجود دارد با عنوان **CallMe** که با کلیک نمودن بر روی آن یک تعامل تلفن اینترنتی بین PC خود و مالک یک صفحه‌ی خانه وب را در هر نقطه برقرار می‌کنید.

در این حالت عملکردی شبکه‌ی راهیابی مدار تلفنی یعنی (PSTN) وجود ندارد بلکه همه‌ی مکالمه‌های از طریق اینترنت از یک نقطه به انتهای دیگری آمدوشد می‌کنند. در همین شرایط محیطی شرکت و نهادها نیز از مراکز تلفنی خصوصی (PBX) استفاده نمی‌کنند بلکه آن‌ها نیز ترافیک مکالمه‌های تلفنی خود را از طریق شبکه محلی پرسرعت آن شرکت یا نهاد انجام می‌دهند.

همه‌ی این سناریوها ممکن است به نظر شما یک داستان تخیلی باشد. اگرچه شبکه‌های تلفنی عمومی و خصوصی فعلی درآینده‌ی نزدیکی ناپدید نخواهند شد اما در هر حال پروتکل‌ها و محصولاتی وجود دارند که این دیدگاه را به واقعیت تبدیل خواهند نمود. در این راستا از جمله مهم‌ترین پروتکل‌هایی که در این مسیر حرکت نموده است **پروتکل آغازگر تعامل**[1] **(SIP)** می‌باشد که در مستندهای [RFC3261 , RFC5411] توصیف شده‌اند. پروتکل SIP به عنوان یک پروتکل سبک وزن عملیات ذیل را انجام می‌دهد:

- تأمین روش‌های برقراری ارتباط بین دو طرف مکالمه‌ی تلفنی از طریق یک شبکه‌ی IP. این پروتکل به متقاضی ارتباط امکان می‌دهد که دریافت کننده‌ی مکالمه را در جریان منظور خود برای شروع یک مکالمه قرار دهد. هم‌چنین اجازه می‌دهد حاضران در مکالمه برروی نوع رمزگذاری رسانه توافق نمایند و نسبت به قطع ارتباط نیز تصمیم‌گیری نمایند.
- تأمین روش‌های لازم برای متقاضی مکالمه جهت دستیابی آدرس IP فعلی متعلق به دریافت کننده‌ی مکالمه. کاربران از یک آدرس IP ثابت و تک برخوردار نیستند به دلیل آن‌که می‌توان آدرس‌ها را به آن‌ها از طریق DHCP و به صورت پویا اختصاص داد. علاوه بر

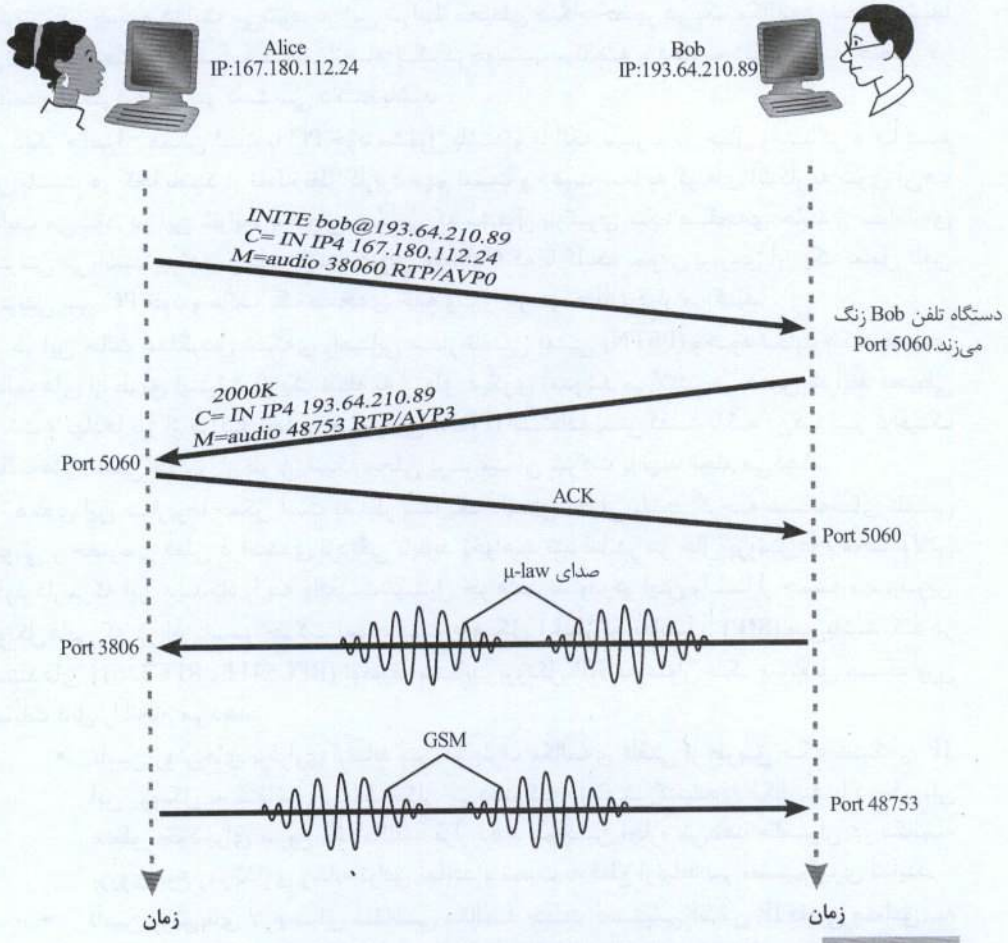
1- Session initiation protocol

آن آن‌ها ممکن است از تجهیزاتی استفاده نمایند که شامل چندین IP هریک با آدرس‌های متفاوت IP باشند.

- تأمین روش‌های لازم مدیریت مکالمه مانند افزودن جریان‌های رسانه‌ای جدید در هنگام انجام مکالمه، تغییر نوع رمزگذاری، دعوت از کاربران جدید به منظور پیوستن به مکالمه و انتظار مکالمه در هنگامی که یک مکالمه جاری است.

۱- برقراری یک ارتباط مکالمه‌ای با یک آدرس IP شناخته شده

به منظور درک شفاف پروتکل SIP شاید بهترین راهکار تحلیل یک مثال به صورت ذیل باشد. اگر فرض شود فرد A از طریق به کارگیری PC خود مایل به برقراری ارتباط مکالمه‌ای با B از طریق PC او باشد و هر کدام نیز از برنامه‌های کاربردی برمبنای SIP استفاده می‌کنند. آن‌گاه با فرض آنکه A از آدرس IP کامپیوتر B آگاه می‌باشد، شکل ۷-۱۴ فرآیند برقراری یک ارتباط SIP را نشان می‌دهد.



شکل ۷-۱۴ ♦ برقراری یک ارتباط **SIP** مشروط بر آگاهی **A** از آدرس **IP** مربوط به **B**

در شکل ۷-۱۴ ملاحظه می‌شود یک تعامل SIP هنگامی آغاز می‌شود که A یک پیام INVITE که مفهوم آن مانند پیام تقاضای HTTP است را به B گسیل می‌نماید. این پیام INVITE از طریق قطعه‌ی

UDP به سوی پورت شناخته شده 5060 برای SIP ارسال می‌شود. بدیهی است پیام‌های SIP را می‌توان از طریق قطعه‌های TCP نیز ارسال نمود. پیام ارسالی INVITE شامل شناسه‌ی B مانند bob@193.64.210.89، آدرس IP جاری A، اطلاعاتی مبنی بر تمامی A به دریافت فایل صدا در قالب AVPO یعنی قانون $\mu - law$ برای PCM رمزگذاری شده باشد و به صورت یک بسته‌ی RTP می‌باشد. علاوه بر آن نیز اطلاعاتی مبنی بر تمایل A به دریافت بسته‌های RTP برروی درگاه 38060 تعریف شده است. پس از آنکه B پیام INVITE را از سوی A دریافت نمود آن‌گاه B یک پیام پاسخ SIP را به B گسیل می‌کند که در واقع مشابه یک پیام پاسخ HTTP است. این پیام پاسخ SIP نیز به درگاه 5060 گسیل می‌شود. واکنش B شامل یک پیام 200 OK و آدرس IP او، قالب رمزگذاری و بسته‌بندی قابل قبول جهت دریافت از سوی B و شماره‌ی درگاه او که بسته‌های صوتی بایستی به آن پورت ارسال شوند، می‌باشد. لازم به توضیح است در این مثال A و B هرکدام از روش‌های رمزگذاری صدای غیرمشابه بهره گرفته‌اند. به عبارت دیگر A رمزگذاری مطلوب خود را در قالب GSM می‌خواهد در صورتی که B مایل به دریافت صدای رمزگذاری شده از طریق قانون $PCM\mu - law$ می‌باشد. پس از آن که A پاسخ B را دریافت نماید آن‌گاه A پیام تأئید SIP را به B ارسال می دارد. با انجام تراکنش SIP هر دو نفر A و B قادر به انجام مکالمه می‌شوند. بدیهی است اگرچه در شکل ۷-۱۴ این موضوع آشکار نمی‌باشد اما هر دو نفر A و B از لحاظ انجام مکالمه همزمان عمل می‌نمایند. شخص B رمزگذاری و بسته‌بندی صدا برمبنای تقاضا را از وی انجام و بسته‌های صوتی را به شماره درگاه 38060 به آدرس IP:167.180.112.24 گسیل می‌نماید. از طرف دیگر A رمزگذاری و بسته‌بندی صدا را برمبنای درخواست B انجام می‌دهد و بسته‌ها را به پورت 48753 در آدرس 193.64.210.89 ارسال می‌دارد.

آن‌چه از این مثال می‌توان آموخت چندین ویژگی کلیدی پروتکل SIP می‌باشد که عبارتند از: نخست SIP یک پروتکل خارج از محدوده (باند) است و پیام‌های از نوع SIP از سوکت‌هایی ارسال و یا دریافت می‌شوند که کاملاً متناوب با سوکت‌های بهره‌گیری شده از سوی داده‌ی رسانه به‌همین منظور می‌باشند. دوم پیام‌های SIP دارای قالب مشابه ASCII و از لحاظ مفهوم مانند HTTP می‌باشند و سوم پروتکل SIP نیازمند دریافت تأئید (ACK) برای همه‌ی پیام‌ها به منظور گسیل آن‌ها از طریق TCP و یا UDP می‌باشد.

در مثال شکل ۷-۱۴ اگر فرض بر آن باشد که B رمزگذاری PCM را برمبنای قانون $\mu - law$ برای صدا درخواست ننماید آنگاه به جای پاسخ با پیام 200 OK آن‌چه B به عنوان پاسخ واکنش نشان می‌دهد عبارتی مانند 600 Not Acceptable و فهرستی از انواع روش‌های رمزگذاری قابل قبول وی می‌باشد. در این شرایط A از میان عناوین رمزگذاری در فهرست یکی از آن‌ها راگزینش و یک پیام INVITE دیگری را گسیل می‌کند که در این پیام نوع روش رمزگذاری گزینش شده نیز قید شده است. بدیهی است B می‌تواند درخواست یک مکالمه را از طریق ارسال یک پیام عدم پذیرش مانند: busy، gone، پرداخت لازم است و یا مکالمه ممنوع است را به متقاضی مکالمه مانند A اعلام نماید.

۲- آدرس SIP

در مثال قبل آدرس SIP وابسته به B به صورت: sip:bob@193:64.210.89 می‌شد. معمولاً قالب بسیاری از آدرس‌های SIP همانند آدرس‌های پست الکترونیکی (ایمیل) می‌باشد. به عنوان مثال آدرس

B ممکن است در قالب sip:bob@domain.com باشد. هنگامی که وسیله‌ی SIP متعلق به A پیام INVITE را ارسال می‌دارد، این پیام می‌تواند شامل یک آدرس مانند ایمیل باشد در این حالت زیرساخت SIP پیام را به سوی وسیله‌ی IP که B در حال حاضر مشغول استفاده‌ی آن می‌باشد گسیل می‌کند. حالت‌های دیگر آدرس SIP می‌تواند در قالب شماره تلفن B و یا نام و نام خانوادگی فرد B باشد.

یکی از نکات جالب در مورد آدرس‌های SIP افزودن آن در صفحه‌های وب می‌باشد. مشابه قرار دادن آدرس‌های ایمیل در صفحه‌های وب آدرس‌های SIP را نیز می‌توان در این گونه صفحه‌ها فهرست نمود. به‌عنوان مثال اگر فرض شود B دارای یک صفحه‌ی شخصی می‌باشد و مایل به تأمین امکانات برای مراجعه کنندگان به آن صفحه می‌باشد او می‌تواند به‌سادگی آدرس URL مانند: sip:bob@domain.com را به صفحه‌ی خود پیوست نماید. هنگامی که یک بازدید کننده برروی این URL کلیک نماید کاربر SIP از سوی وسیله بازدید کننده فعال و یک پیام INVITE را به سوی B ارسال می‌دارد.

۳- پیام‌های SIP

بدیهی است در این گفتار کوتاه نمی‌توان انواع پیام‌های SIP را به همراه سرپیام‌های آن‌ها تحلیل نمود. به‌همین منظور نگاهی خلاصه به پیام SIP/INVITE می‌اندازیم. فرض کنید A مایل به‌دعوت یک مکالمه‌ی تلفنی IP از B می‌باشد و این مرتبه A فقط از آدرس SIP متعلق به B یعنی: bob@domain.com آگاه است به عبارت دیگر از آدرس IP وسیله‌ای که B در حال حاضر از آن استفاده می‌کند، آگاهی ندارد. در این حالت پیام A مشابه قطعه‌ی ذیل می‌تواند به‌شمار آید:

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

در این قطعه به ترتیب: سطر مربوط به INVITE بیان کننده‌ی نسخه‌ی SIP مانند آن‌چه در یک پیام تقاضای HTTP است، می‌باشد. هرگاه یک پیام SIP از یک وسیله‌ی از نوع SIP عبور نماید یک عبارت سرپیام Via را به آن پیوست می کند که بیان کننده‌ی آدرس IP آن وسیله است. بدیهی است این عمل حتی اگر وسیله خود منبع تولید پیام باشد نیز انجام می‌گیرد. بدیهی است یک پیام INVITE می‌تواند از درون تعداد قابل توجهی از تجهیزات SIP قبل از ورود به کاربرد SIP عبور نماید. همانند یک پیام ایمیل یک پیام SIP شامل یک سرانداز From و To می‌شود. علاوه بر همه‌ی این موارد پیام شامل یک عبارت Call-ID که مکالمه را شناسایی می‌کند نیز می‌باشد. این موضوع مشابه پیام ID در ایمیل است. از جمله موارد دیگر خط سرپیام نوع محتوا Content Type است که قالب استفاده شده جهت توصیف پیام SIP را تعریف می‌کند. خط سرپیام طول محتوا Content Length تعداد بایت‌های موجود در محتوای پیام را مشخص می‌سازد و نیز کلید بازگشت (لـ) و تغذیه‌ی خط به دنبال پیام افزوده شده

و سرانجام در ادامه‌ی همه‌ی این موارد محتوای اصلی داده‌ی موردنظر وارد می‌شود. در این حالت آن‌چه محتوا فراهم می‌آورد عبارت است از آدرس IP متعلق به A و نیز چگونگی قالب دریافت صدا از سوی A.

۴- ترجمه‌ی نام و مکان کاربر

در مثال شکل ۷-۱۴ فرض بر آن بود که وسیله‌ی SIP متعلق به A از آدرس IP تجهیزات B که بایستی به آن مرتبط می‌گردید، آگاهی دارد. این فرض در عمل دور از واقعیت می‌باشد نه فقط به دلیل آنکه آدرس‌های IP به‌گونه‌ی پویا از سوی DHCP اختصاص می‌یابند بلکه به دلیل آنکه B ممکن است چندین وسیله‌ی IP را در مکان‌های گوناگون مانند محل کار، مسکن و یا خودرو در اختیار داشته باشد. اکنون فرض کنید A فقط از آدرس ایمیل B مانند: bob@domain.com آگاه می‌باشد و از همین آدرس نیز به منظور ارتباط‌های برمبنای SIP بهره می‌گیرد. در این حالت A نیازمند اخذ آدرس IP تجهیزاتی است که کاربر bob@domain.com در حال حاضر استفاده می‌کند. به همین منظور A یک پیام INVITE تعریف می‌کند که قالب آن به صورت:

INVITEbob@domain.comSIP/2.5 می‌باشد و این پیام را به یک سرور نهان یعنی SIP proxy گسیل می‌کند. سرور نهان یک پیام پاسخ SIP را ارسال می‌کند که ممکن است شامل آدرس IP وسیله‌ی جاری باشد که B از طریق bob@domain.com استفاده می‌کند. از طرف دیگر سرور نهان ممکن است پیام پاسخ SIP را به‌گونه‌ای آماده‌سازی نماید که شامل آدرس IP از صندوق پست صوتی (Voicemail) متعلق به B و یا آنکه URL صفحه‌ی وب متعلق به B باشد. هم‌چنین نتیجه‌ی حاصل از سرور نهان (proxy) ممکن است به نوعی وابستگی به متقاضی مکالمه داشته باشد به بیان دیگر ممکن است فردی که متقاضی ارتباط با B می‌باشد همسر وی باشد و در این‌صورت B ممکن است مکالمه را پذیرش نموده و آدرس IP خود را بازگشت دهد. از طرف دیگر منبع مکالمه ممکن است از سوی مادر همسر B باشد و در این صورت B پاسخ مکالمه را ممکن است با آدرس URL صفحه‌ی وب خود پاسخگویی نماید و آن صفحه نیز ممکن است به عنوان مثال حامل پیامی باشد که اعلام می‌دارد "من یک صفحه وب در حال خواب می‌باشم!"

اکنون ممکن است این پرسش برای شما مطرح شود چگونه سرور نهان آدرس IP جاری bob@domain.com را دستیابی می‌کند؟ برای پاسخ به این پرسش نیازمند معرفی وسیله‌ی SIP دیگری می‌باشیم که اصطلاحاً به آن **ثبات SIP**[1] گفته می‌شود. هر کاربر SIP به نوعی وابسته به یک ثبات SIP می باشد. هرگاه یک کاربر یک برنامه‌ی کاربردی SIP را فعال می‌کند آن کاربرد یک پیام ثبت SIP را به ثبات ارسال می‌دارد و از این طریق ثبات را از آدرس IP جاری خود آگاه می‌سازد. به‌عنوان مثال هنگامی که B کاربرد SIP خود را برروی تبلت فعال می‌سازد آن کاربرد ممکن است پیامی را در قالب سطرهای ذیل ارسال نماید:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

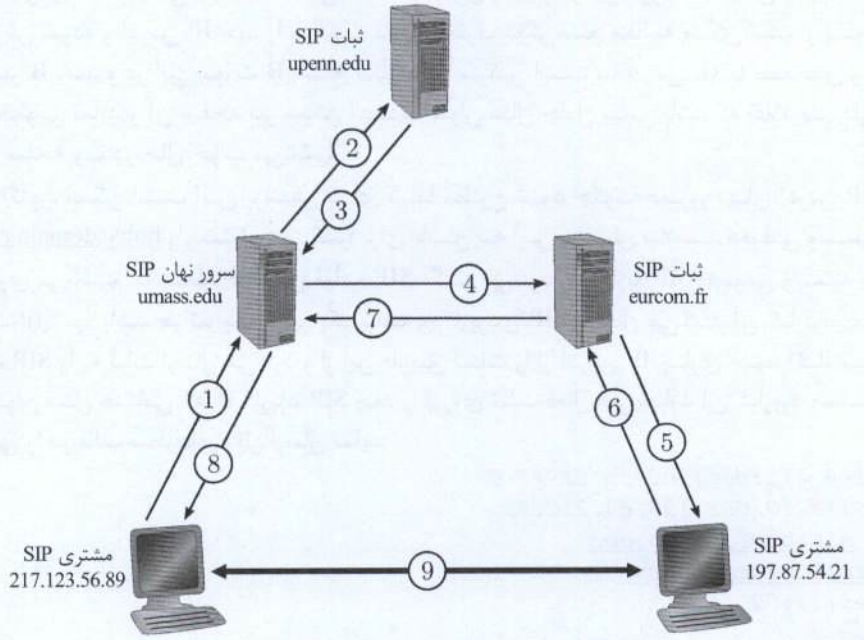
ثبات متعلق به B نسخه‌ای از سابقه‌ی آدرس IP جاری B را نگهداری می‌کند. هرگاه B یک

1- SIP registrar

وسیله‌ی SIP جدیدی را فعال‌سازی می‌کند آن وسیله پیام ثبت جدیدی را به ثبات گسیل می‌کند و از طریق این پیام آدرس IP جدید را در ثبات ذخیره می‌کند. بدیهی است اگر B برای مدت طولانی از یک وسیله‌ی SIP استفاده نماید آن وسیله پیام تمدید آدرس IP را به ثبت گسیل می‌کند، به عبارت دیگر اعلام می‌دارد آخرین آدرس IP ارسال شده هنوز معتبر می‌باشد. در مثال بالا پیام تمدید اعتبار آدرس IP بایستی هر 3600 ثانیه یک بار به منظور تمدید اعتبار به سرور ثبات ارسال گردد. لازم به توضیح است که ثبات آدرس بسیار مشابه سرور نام خودمختار DNS می‌باشد که در فصل‌های پیشین از آن یاد شده است. یک سرور DNS نام‌های ثابت میزبان را به آدرس‌های IP ثابت ترجمه می‌نمود. ثبات SIP نیز شناسه‌های فردی ثابت انسانی مانند bob@domain.com را به آدرس‌های IP پویا ترجمه می‌کند. معمولاً ثبات‌های SIP و سرورهای نهان SIP برروی یک میزبان مشابه اجرا می‌شوند.

اکنون بازگردیم به چگونگی دستیابی آدرس IP جاری B از سوی سرور نهان SIP متعلق به A. از مطالب گذشته می‌توان استنباط نمود که سرور نهان نیازمند گسیل پیام INVITE متعلق به A به سوی ثبات B می‌باشد. ثبات و سرور نهان نیز می‌تواند پیام را به سوی وسیله‌ی SIP که B در حال حاضر از آن استفاده می‌نماید، گسیل دارد. سرانجام B پس از دریافت پیام INVITE از سوی A قادر به ارسال یک پاسخ SIP به سمت A می‌شود. برای آشکار شدن بیش‌تر موضوع مثال شکل ۷-۱۵ را درنظر بگیرید. در این مثال:

Jim@umass.edu در حال حاضر از طریق آدرس 217.123.56.89 مشغول فعالیت می‌باشد و مایل به انجام یک تعامل VOIP با keith@Upenn.edu که آدرس جاری او 197.87.54.21 می‌باشد. مراحل ذیل بایستی به منظور انجام این مهم صورت پذیرد:



شکل ۷-۱۵ ♦ آغاز یک تعامل با به‌کارگیری ثبات و سرورهای نهان **SIP**

۱- jim یک پیام INVITE به سرور SIP متعلق به Umass ارسال می‌دارد.
۲- سرور SIP جستجوی DNS در ثبات SIP متعلق به upenn.edu که در شکل ۷-۱۵ نشان داده نشده است را انجام می دهد و سپس پیام را به ثبات سرور ارسال می‌کند.
۳- از آن‌جائیکه keith@upenn.edu در ثبات upenn حضور ندارد در نتیجه ثبات upenn یک پاسخ غیرمستقیم را به منظور جستجوی keith@eurecom.fr ارسال می‌دارد.
۴- سرور نهان umass یک پیام INVITE به ثبات SIP در eurecom گسیل می‌کند.
۵- ثبات eurecom از آدرس IP مربوط به keith@eurecom.fr آگاه است و بنابراین پیام INVITE را به میزبان 197.87.54.21 که در حال حاضر برنامه‌ی مشتری SIP برای keith را اجرا می‌کند، گسیل می‌دارد.

گام‌های ۶ تا ۸- یک پیام پاسخ SIP به سوی سرورهای نهان و ثبات‌ها و از آنجا به برنامه مشتری SIP در آدرس 217.123.56.89 ارسال می‌دارد.

۹- رسانه مستقیماً بین دو برنامه‌ی مشتری مبادله می‌شود و یک پیام تأیید SIP نیز همراه آن می‌باشد.

ملاحظه گردید بحث پیرامون SIP در اطراف چگونگی فعال‌سازی مکالمه‌های صوتی متمرکز شده بود. پروتکل SIP به‌طور کلی یک عملیات سیگنالینگ برای ایجاد و قطع ارتباط‌ها را کنترل می‌نماید. از این پروتکل می توان در کاربردهای صوتی، تصویری و مکالمه‌های به‌صورت جمعی (همایش) و نیز تعامل‌های متنی بهره گرفت. در واقع پروتکل SIP یکی از عناصر بنیادی کاربردهای انتقال پیام بی‌درنگ می‌باشد.

۷-۱۱ پروتکل H.323

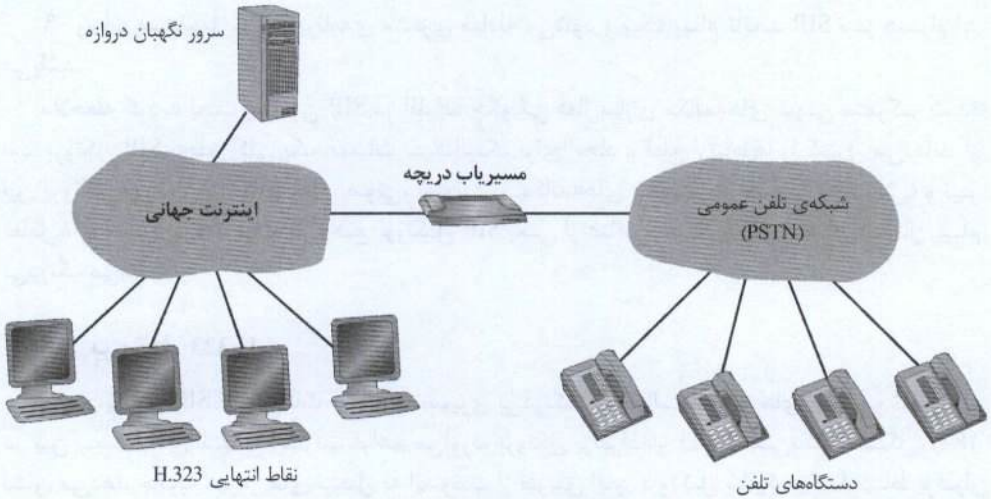
پروتکل جایگزین SIP که امکانات صوتی و تصویری بی‌درنگ را در قالب همایش‌های صوتی و تصویری در بین سیستم‌های انتهایی اینترنت فراهم می‌آورد، پروتکل پرطرفدار H.323 می‌باشد. شکل ۷-۱۶ نشان می‌دهد چگونه میزبان‌های متصل به اینترنت از طریق این پروتکل با یکدیگر ارتباط برقرار می‌کنند. از شکل می‌توان مشاهده نمود همین میزبان‌های اینترنتی قادر به تعامل با تلفن‌های معمولی متصل به شبکه‌های متداول تلفنی (PSTN) نیز می‌باشند. بدیهی است پروتکل SIP نیز قادر به تأمین این امکانات می‌باشد. مشابه ثبات SIP، پروتکل H.323 از امکانات نرم افزاری با عنوان **نگهبان دروازه‌ی H.323**[1] برای انجام امور ثبت آدرس‌های IP استفاده می‌کند. استاندارد H.323 شامل مجموعه‌ای از مشخصات ذیل است:

- مشخصات مربوط به توافق برروی روش‌های رمزگذاری صدا و تصویر بین میزبان‌ها. از آنجائیکه پروتکل H.323 انواع متفاوتی از استانداردهای رمزگذاری را برای صدا و تصویر پشتیبانی می‌کند بنابراین حضور یک پروتکل ویژه به منظور ارتباط بین سیستم‌های انتهایی و کنترل توافق آن‌ها برای یک روش رمزگذاری مشترک ضروری می‌باشد.
- مشخصات به منظور بسته‌بندی گروهی بسته‌های صوتی و تصویری و ارسال آن‌ها از طریق شبکه به‌ویژه آنکه پروتکل H.323 از پروتکل RTP به عنوان یک الزام بهره می‌گیرد.

1- H.323 gate keeper

- مشخصات به منظور چگونگی برقرار ارتباط بین سیستم‌های انتهایی و نگهبانان دروازه‌ی H.323.
- مشخصات لازم جهت برقراری ارتباط تلفن اینترنتی از طریق مسیریاب دریچه با تلفن‌های معمولی در PSTN.

هر نقطه‌ی انتهایی H.323 حداقل لازم است استاندارد فشرده سازی مکالمه‌ی G.711 را پشتیبانی نماید. پروتکل G.711 با بهره‌گیری از روش PCM مکالمه‌های دیجیتالی را در قالب سرعت‌های 56kbps و یا 64kbps رمزگذاری می‌نماید. اگرچه پروتکل H.323 هر نقطه‌ی انتهایی را مجبور به تأمین امکانات G.711 به منظور رمزگذاری صدا می‌نماید اما این مهم برای تصویر حالت اختیاری دارد. از آن‌جائیکه رمزگذاری تصاویر در H.323 حالت اختیاری دارند در نتیجه تولیدکنندگان تجهیزات بازپخش صوتی و تصویری قادر خواهند شد که وسایل ساده‌تر صوتی را در کنار تجهیزات پیچیده‌ی صوتی و تصویری در معرض فروش قرار دهند.



شکل ۷-۱۶ ♦ سیستم‌های انتهایی **H.323** و چگونگی ارتباط آن‌ها از طریق اینترنت با تلفن‌های معمولی متصل به شبکه‌های تلفنی راهیابی مدار **(PSTN)**.

توان‌مندی‌های تصویری H.323 کاملاً اختیاری می‌باشد اما در هر حال اگر یک میزبان انتهایی امکانات ویدیویی را پشتیبانی نماید آن‌گاه لازم است حداقل استاندارد تصویری مربوط به پروتکل QCIFH.261 را که عبارت است از شفافیت تصویر 144×176 پیکسل را نیز پشتیبانی نماید.

به‌طور کلی پروتکل H.323 را می‌توان در برگیرنده‌ی مجموعه‌ای از استانداردهای صوتی و تصویری مانند آن‌چه در بالا اشاره گردید به همراه انواع دیگری نظیر پروتکل‌های کنترل H.245، سیگنالینگ Q.931 و یک پروتکل RAS به منظور ثبت نام با امکانات نگهبان دروازه به‌شمار آورد. تفاوت‌های عمده‌ی مابین دو پروتکل SIP و H.323 عبارتند از:

- پروتکل H.323 شامل مجموعه‌ای از پروتکل‌های عمودی برای پشتیبانی چندرسانه‌ای شامل: سیگنالینگ، ثبت‌نام، کنترل پذیرش، انتقال و رمزگذاری می‌باشد.
- از طرف دیگر پروتکل SIP فقط قادر به تأمین تعامل و مدیریت می‌باشد و فقط یک عنصری

در برابر چندتایی H.323 می‌باشد. پروتکل SIP به همراه RTP عمل می نماید اما الزامی در به‌کارگیری آن ندارد. این پروتکل با سیستم‌های رمزگذاری مکالمه‌ی G.711 و تصویری QCIFH.261 نیز عمل می‌نماید اما الزامی را درکاربرد آن‌ها تحمیل نمی‌کند. این پروتکل را می‌توان با خدمات و پروتکل‌های دیگر ترکیب نمود.

- منبع پروتکل H.323 از ITU و در نتیجه تلفنی است در حالیکه سرچشمه‌ی پروتکل SIP از نهاد IETF می‌باشد و بسیاری از اصول خود را از وب، DNS و ایمیل اینترنت گرفته است.
- پروتکل H.323 اصطلاحاً حالت چتر پوششی دارد و بنابراین گسترده و پیچیده است در حالیکه پروتکل SIP از اصول **KISS**[1] به مفهوم بهره‌گیری از طراحی ساده و بی‌خردانه بهره می‌گیرد.

۷-۱۲ تأمین کلاس‌های چندگانه‌ی خدمات

در بخش‌های قبل کاربرد شمارگان ردیف، برچسب‌های زمانی FEC، RTP و H.323 در برنامه‌های کاربردی چندرسانه‌ای اینترنت بررسی گردیدند. شبکه‌های توزیع محتوا (CDN) در ارتباط با توزیع چندرسانه‌ای اینترنت نیز تشریح گردید. اکنون این پرسش مطرح می‌شود که آیا همه‌ی این روش‌ها به تنهایی جهت پشتیبانی قابل اطمینان و قدرتمند از کاربردهای چندرسانه‌ای مانند خدمات تلفنی IP که معادل شبکه‌های تلفنی فعلی در سراسر گیتی است، کافی می‌باشد؟ قبل از پاسخ این پرسش یادآوری این نکته ضروری است که اینترنت امروزی خدمات با بهترین تلاش را در اختیار همه‌ی کاربردهای خویش قرار می‌دهد. به بیان دیگر ضمانت QoS دریافتی ویژه‌ای را برای کاربردها فراهم نمی‌آورد. یک کاربرد، سطحی از کارآمدی را دریافت می‌نماید که شبکه قادر به تأمین آن در یک لحظه‌ی ویژه می‌باشد. بدیهی است اینترنت امروزی هیچ نوع امتیاز ویژه‌ای را در اختیار کاربردهای چندرسانه‌ای حساس در برابر تأخیر قرار نمی‌دهد. از آن‌جائیکه هر بسته حتی بسته‌های صوتی و تصویری حساس در برابر تأخیر به‌گونه‌ی برابر در مسیریاب‌ها خدمات‌رسانی می‌شوند آن‌چه باعث تخریب این برابری در یک مکالمه‌ی تلفن IP در جریان می‌شود تراکم ترافیک شبکه است که سبب افزایش تأخیر و تضعیف مکالمه‌ی تلفنی IP می‌شود.

اما اگر هدف تأمین یک الگوی خدمات بیش از توان بهترین تلاش شبکه در اینترنت فعلی باشد آن‌گاه چه نوع خدماتی بایستی فراهم گردد؟ یک الگوی توسعه یافته‌ی خدمات، تقسیم ترافیک به درجه‌ها یا کلاس‌های مختلف و تأمین سطوح گوناگون خدماتی به این ترافیک‌ها می‌باشد. به عنوان مثال یک ISP ممکن است مایل به تأمین کلاس خدماتی به مراتب بالاتری برای کاربرد تلفن IP و یا کنفرانس تلفنی حساس در برابر تأخیر در مقایسه با یک ترافیک انعطاف‌پذیر مانند FTP و یا HTTP باشد. بدیهی است همه‌ی ما به نوعی با مزایای خدمات برمبنای کلاس مانند بلیط هواپیمای درجه یک و غیره آشنا می‌باشیم، به عنوان مثال یک خدمات هوایی از نوع VIP دارای این ویژگی است که مسافر VIP بی‌درنگ پس از ورود به سالن پرواز فرودگاه با هیچ گونه تأخیری از لحاظ خدماتی مانند تحویل بار و گرفتن کارت پرواز و نظایر آن روبه‌رو نمی‌شود در حالیکه مسافران درجه‌های دیگر بنا بر نوع آن‌ها بایستی در انتظار دریافت این خدمات قرار گیرند.

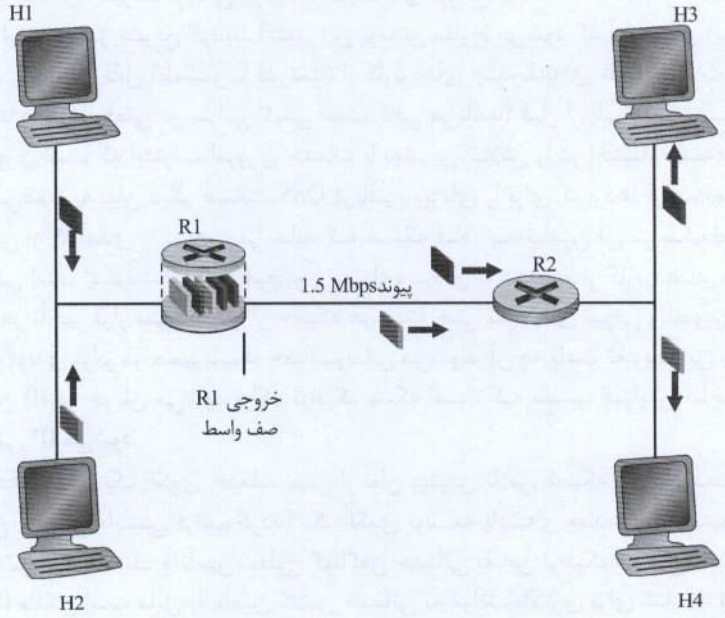
آن‌چه اهمیت دارد توجه به این نکته است که چنین خدمات متفاوتی در بین ترافیک انباشتی در

1- Keep it simple, stupid

بین کلاس‌های ترافیکی و نه در بین ارتباط‌های منفرد چشمگیر می‌باشد. به عنوان مثال همه‌ی مسافران درجه یک از خدمات مشابه برخوردار می‌باشند و تفاوتی بین آن‌ها وجود ندارد این نوع برخورد به یک ارتباط انتها به انتها که آن‌ها متعلق به آن می‌باشند، وابستگی ندارد. در ادامه ملاحظه می‌شود برخورد با گروه کوچکی از ترافیک انباشتی در مقایسه با تعداد بی‌شماری ارتباط منفرد نیازمند روش‌های جدید جهت تأمین خدمات ساده‌ی از نوع بهترین تلاش در شبکه می‌شود.

بدیهی است طراحان اینترنت اولیه این موضوع را درنظر داشته‌اند زیرا دامنه‌ی نوع خدمات (TOS) را در سرپیام بسته‌های پروتکل IPV4 در شکل ۴-۱۳ (فصل ۴) تعریف نموده‌اند. حتی سه دهه‌ی پیش نیاز به تأمین سطوح خدماتی متفاوت به درجه‌های گوناگون ترافیکی آشکار بوده است! اگرچه زمان نسبتاً طولانی گذشته است تا این حقیقت خود را آشکار سازد.

در بخش ۷-۱۳ چندین سناریوی مختلف که نیاز برای پشتیبانی از کلاس‌های گوناگون خدماتی را آشکار می‌سازد تحلیل خواهند شد. پس از آن دو عنوان بسیار مهم که عبارتند از **زمان‌بندی سطح پیوند** و **درجه‌بندی بسته** بررسی می‌شوند و در ادامه موضوع پروتکل جاری اینترنت یعنی Diffserv تحلیل خواهد شد.



شکل ۷-۱۷ ♦ یک شبکه‌ی ساده با دو کاربرد

۷-۱۳ موارد گوناگون خدمات اینترنت

شکل ۷-۱۷ حالتی از یک شبکه‌ی ساده را نشان می‌دهد. فرض کنید دو جریان کاربردی بسته‌ی داده از طریق میزبان‌های H1 و H2 در یک شبکه‌ی محلی تعریف می‌شوند و هدف آن‌ها میزبان‌های H3 و H4 در یک شبکه‌ی محلی دیگر است. مسیریاب‌های موجود در دو شبکه‌ی محلی از طریق یک پیوند 1.5Mbps به یکدیگر متصل شده‌اند. فرض کنید سرعت شبکه‌ی محلی به مراتب بالاتر از 1.5Mbps باشد و تمرکز برروی صف خروجی مسیریاب R1 باشد. در این مکان است که تأخیر و فقدان بسته در

صورت افزایش سرعت انباشتی H1 و H2 نسبت به سرعت 1.5 Mbps روی می‌دهد. اکنون با توجه به این مفروضات چندین حالت مختلف تحلیل می‌شوند که هریک از آن‌ها اطلاعات مهم را از نقطه‌نظر نیاز برای روش‌های ویژه‌ی پشتیبانی خدمات کلاس‌های چندگانه در اختیار ما قرار می‌دهند.

سناریوی ۱: یک کاربرد صوتی 1Mbps و یک انتقال FTP

شکل ۷-۱۸ حالتی را نشان می‌دهد که در آن یک کاربرد صوتی 1Mbps مانند پخش با کیفیت CD از یک پیوند مشترک 1.5Mbps را بین R1 و R2 با یک کاربرد FTP که انتقال یک فایل از H2 به H4 را انجام می‌دهد، بهره می‌گیرد. در اینترنت بهره‌گیر از بهترین تلاش، بسته‌های FTP و صوتی در صف خروجی R1 ترکیب می‌شوند و معمولاً به روش FIFO از صف خارج می‌شوند. در این حالت زنجیره‌ای از بسته‌ها از منبع FTP قادر به انباشت صف خواهند شد و در نتیجه بسته‌های صوتی IP با تأخیرهای طولانی و یا اتلاف آن‌ها به واسطه‌های سرریزی بافر در R1 روبه‌رو می‌شوند. چگونه این چالش عمده را می‌توان برطرف نمود؟

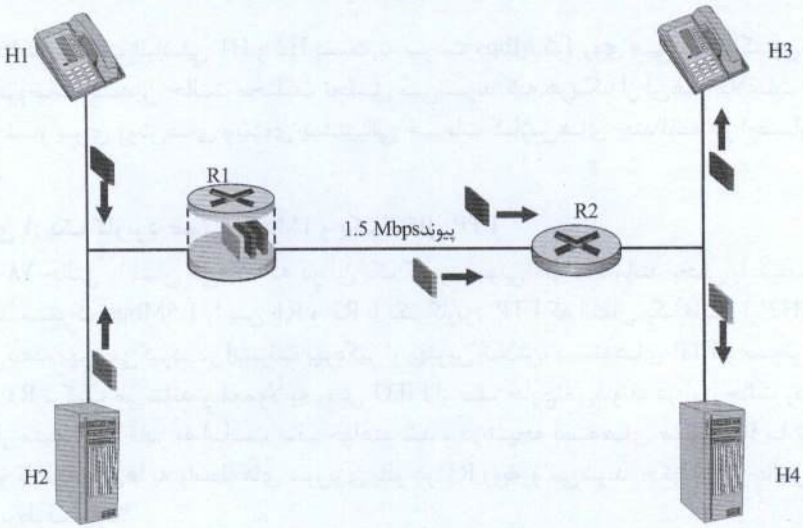
یک پاسخ می‌تواند این باشد که با توجه به این مهم که کاربرد FTP محدودیت زمانی ندارد این استنباط را می‌توان در نظر گرفت که اولویت ویژه‌ای به بسته‌های صوتی در R1 اختصاص پیدا کند. در شرایط اختصاص اولویت زمان‌بندی شده‌ی یک بسته‌ی صوتی در بافر خروجی R1 همواره قبل ازهر بسته‌ی FTP از همان بافر خارج می‌شود. پیوند R1 به R2 مانند یک پیوند انحصاری 1.5Mbps برای ترافیک صوتی عمل می‌کند و انتقال FTP با بهره‌گیری از پیوند R1 به R2 فقط هنگامی که هیچ گونه ترافیک صوتی در صف وجود ندارد استفاده می‌شود.

به منظور تمایز بین بسته‌های صوتی و FTP در صف R1، هر بسته بایستی از نظر وابستگی به یکی از دو نوع کلاس ترافیکی علامت‌گذاری شوند. رعایت این موضوع در حقیقت منظور اصلی نوع خدمات (TOS) در دامنه‌ی سرپیام IPV4 می‌باشد. اگرچه این مهم بدیهی به‌نظر می‌رسد اما در هرحال این نخستین راهکاری است که در ذهن متبادر می‌شود که بتوان کلاس‌های گوناگونی از ترافیک را تعریف نمود. نتایج مختلف در این رابطه را می‌توان به‌صورت ذیل فهرست نمود:

نتیجه‌گیری نخست: علامت‌گذاری بسته‌ها اجازه می‌دهد یک مسیریاب بین بسته‌های متعلق به کلاس‌های گوناگون ترافیکی تمایزی را شناسایی نماید.

سناریوی ۲: یک کاربرد صوتی 1Mbps با یک انتقال FTP الویت بالا

حالت دوم اندکی تفاوت نسبت به وضعیت اشاره شده در بالا دارد. فرض کنید کاربر FTP از امکانات دستیابی اینترنت سرعت بالای یک ISP برخوردار است و در نقطه‌ی مقابل کاربر صوتی از همان ISP پیوندی را اجاره نموده است که ارزان قیمت و در نتیجه سرعت نسبتاً پائین دستیابی را در اختیار دارد. آیا در این شرایط کاربر صوتی بایستی از الویت بالایی نسبت به کاربری که بسته‌های FTP جهت مبادله دارد، برخوردار باشد؟ بدیهی است پاسخ خیر می‌باشد. در این حالت آن‌چه منطقی به نظر می‌رسد استفاده از رویکرد تمایز بسته‌ها برمبنای آدرس IP فرستنده می‌باشد. در حالت کلی‌تر لازم است که مسیریاب بسته‌ها را برمبنای اعمال شرایط و قواعدی درجه‌بندی نماید. نتیجه آنکه برداشت نخست در موضوع قبلی اندکی بایستی تغییر نماید و در قالب جدید آن به‌گونه‌ی ذیل تعریف شود:



شکل ۷-۱۸ ♦ رقابت کاربردهای صوتی و **FTP** با یکدیگر

- **برداشت تغییر یافته‌ی نخست:**

طبقه‌بندی بسته‌ها اجازه می‌دهد یک روتر تمایز بین بسته‌ها را برمبنای تعلق آن‌ها به کلاس‌های مختلفی از ترافیک انجام دهد.

علامت‌گذاری انحصاری بسته یکی از روش‌هایی است که در آن بسته‌ها ممکن است از یکدیگر متمایز شوند. علامت‌گذاری انجام شده از سوی یک بسته به مفهوم الزام آن‌که یک بسته دریافت کننده‌ی کیفیت خدماتی ویژه‌ای است، نخواهد بود، بلکه علامت‌گذاری بسته‌ها تنها یک روش به‌منظور تمایز بسته‌ها از یکدیگر می‌باشد. چگونگی تمایز بین بسته‌ها از سوی مسیریاب وابسته به یک سیاست تصمیم‌گیری دارد.

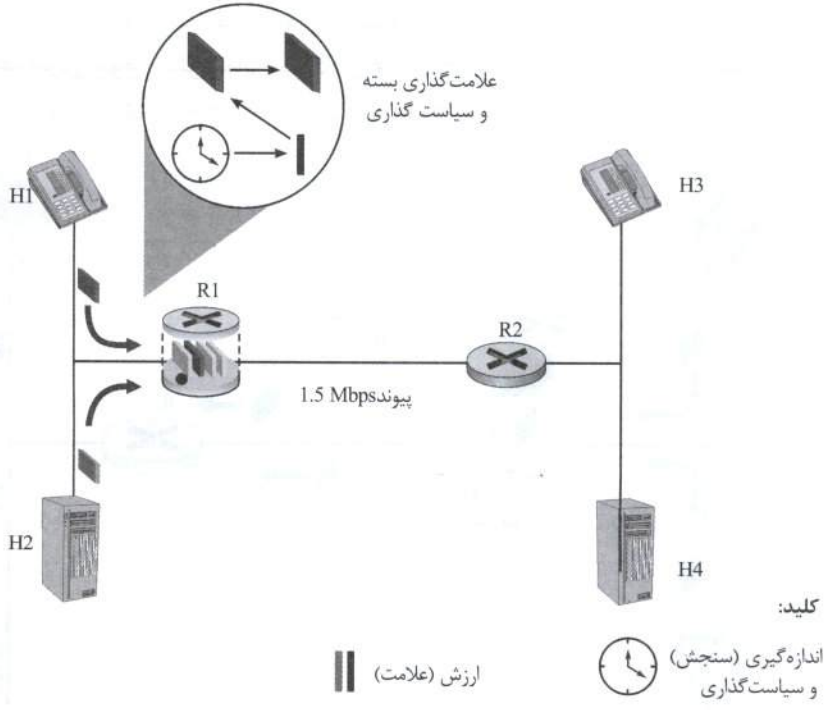
سناریوی ۳: عملکرد نامتعارف کاربرد صوتی و یک انتقال FTP

اکنون فرض کنید به نوعی مسیریاب از اختصاص الویت به بسته‌های صوتی از یک کاربرد 1Mbps آگاه است. از آن‌جائیکه سرعت پیوند خروجی 1.5Mbps است، حتی اگر بسته‌های FTP دارای الویت کمتری باشند آن‌ها هنوز هم به‌طور میانگین 0.5Mbps از خدمات انتقال را دریافت می‌دارند. اما اگر کاربرد صوتی شروع به ارسال بسته‌ها با سرعت 1.5Mbps و یا بیش‌تر عمداً و یا در اثر وقوع خطا در کاربرد نماید، آن‌گاه چه اتفاقی روی می‌دهد؟ در این حالت بسته‌های FTP دچار گرسنگی پهنای باند می‌شوند. به بیان دیگر آن‌ها هیچ‌گونه خدماتی را برروی پیوند R1 به R2 دریافت نمی‌دارند. مشابه این وضعیت هنگامی روی می‌دهد که کاربردهای چندگانه مانند مکالمه‌های صوتی چندگانه که همگی از لحاظ الویت یکسان می‌باشند از یک پهنای باند مشترک در یک پیوند بهره می‌گیرند. یک جریان غیرمتعارف می‌تواند سبب تضعیف و تخریب کارایی جریان‌های دیگر شود. ایده‌آل آن است که نوعی درجه‌ی تمایز بین درجه‌های ترافیک و احتمالاً بین جریان‌های ترافیکی موجود در یک کلاس لازم باشد. منظور از ایمن‌سازی یک جریان از دیگری که اصطلاحاً غیرعادی و نامتعارف است در تناقض با گفتار پیشین قرار می‌گیرد که مبنای آن برابری رفتار برای بسته‌های متعلق به همه‌ی جریان‌ها در درون یک کلاس می‌شد. در عمل بسته‌های درون یک کلاس یقیناً با نگاهی برابر در مسیریاب‌های درون هسته‌ی شبکه

انگاشته می‌شوند. در نبش شبکه بسته‌های متعلق به یک جریان ممکن است مورد سنجش قرار گیرند تا اطمینان حاصل شود که سرعت انباشتی هریک از جریان‌ها از مقدار معین و تعریف شده‌ای فراتر نمی‌رود. این موضوع سبب تعریف نتیجه‌گیری دوم به‌گونه‌ی ذیل می‌شود:

- **نتیجه‌گیری دوم:** اعمال درجه‌ای از جداسازی در بین کلاس‌های ترافیک و جریان به نظر مطلوب است زیرا سبب می‌شود یک کلاس یا جریان به‌صورت جدی در معرض تأثیر یک کلاس و یا جریان نامتعارف قرار نگیرد.

در بخش‌های بعدی چندین راهکار ویژه به‌منظور تأمین رویکردهای جداسازی مابین درجه‌های ترافیکی و جریان‌ها ارایه می‌شود. به‌طور کلی دو روش فراگیر امکان‌پذیر است. نخست امکان کنترل ترافیک برمبنای آن‌چه در شکل ۷-۱۹ نشان داده شده است. اگر یک ترافیک ویژه و یا جریان بایستی شرایط معینی مانند ممانعت از افزایش سرعت فراتر از 1Mbps را برای یک جریان صوتی رعایت نماید آن‌گاه اعمال فرآیندهای نظارتی به‌منظور دستیابی این هدف لازم می‌باشد. اگر کاربردی که از سوی فرآیند نظارتی کنترل می‌شود از خود سوء رفتاری را نشان دهد آن‌گاه فرآیند ناظر می‌تواند مبادرت به اقداماتی مانند حذف و یا تأخیر بسته‌هایی نماید که عامل این وضعیت می‌باشند تا در نتیجه‌ی این اقدام ترافیک وارد شده به درون شبکه کاملاً رفتاری معقول را به‌دنبال داشته باشد. از جمله روش‌های نظارتی پرطرفدار که در آینده بررسی خواهد شد اصطلاحاً **روش دَلو سوراخ**[1] و یا نشت سطل می‌باشد.



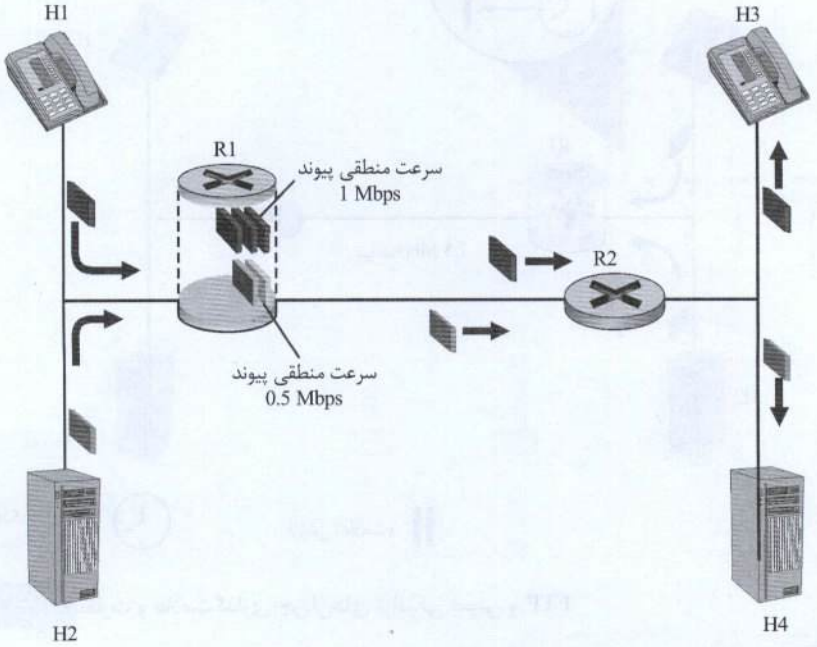
شکل ۷-۱۹ نظارت و علامت‌گذاری جریان‌های ترافیکی صوتی و **FTP**

1- Leaky bucket algorithm

شکل ۷-۱۹ طبقه‌بندی بسته و علامت‌گذاری برمبنای نظارت 1 و کنترل براساس نظارت 2 در یک مکان مشترک در لبه‌ی شبکه در میزبان‌ها و یا مسیریاب‌های نبش شبکه را نشان می‌دهد. رویکرد جایگزین به‌منظور جداسازی ترافیک مابین کلاس‌ها و یا جریان‌ها روش زمان‌بندی بسته‌ها در سطح پیوند می‌باشد. به‌گونه‌ای کاملاً انحصاری پهنای باند پیوندها را براساس مقدار ثابت و از پیش تعیین شده به هر کلاس یا جریان اختصاص می‌دهند. به عنوان مثال جریان صوتی می‌تواند برابر 1Mbps در R1 و جریان FTP به‌مقدار 0.5Mbps در همان مکان اختصاص یابد. در این حالت هر دو جریان صدا و FTP یک پیوند منطقی با ظرفیت یک و 0.5 مگابیت بر ثانیه را به‌ترتیب برمبنای شکل ۷-۲۰ ملاحظه می‌کنند.

با اعمال قدرتمند اختصاص پهنای باند در سطح پیوند یک کلاس یا جریان فقط می‌تواند پهنای باند اختصاص یافته به هریک را مورد بهره‌برداری قرار دهد و بدیهی است از امکانات پهنای باند موجود و استفاده نشده از سوی دیگر کاربران نمی‌تواند استفاده نماید. به عنوان مثال اگر جریان صوتی در اثر سکوت گوینده بسته‌های صدا تولید نکند یک جریان FTP قادر به ارسال بسته‌های خود با سرعت بالاتر از 0.5Mbps از طریق پیوند R1 به R2 نخواهد بود حتی اگر جریان صوتی 1Mbps در آن لحظه مورد بهره‌برداری واقع نشده باشد. بنابراین استفاده‌ی کارآمد از پهنای باند بسیار مطلوب می‌باشد. تأمین این مهم از طریق مجاز دانستن یک کلاس یا جریان در بهره‌گیری از پهنای باند دیگر کاربرانی که از امکانات خود در یک لحظه‌ی زمانی استفاده نمی‌کنند، میّسر می‌شود. این موضوع سبب تعریف نتیجه‌گیری سوم از امکانات به‌صورت توصیف ذیل می‌شود:

- **نتیجه‌گیری سوم**: تأمین تمایز بین کلاس‌ها و یا جریان‌ها هنگامی مطلوب خواهد بود که حداکثر بهره‌برداری از منابع مانند پهنای باند پیوندها و بافرها به عمل آید.



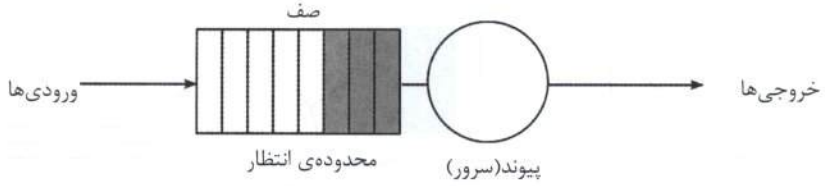
شکل ۷-۲۰ ♦ جداسازی منطقی جریان‌های کاربردی صوتی و FTP

۷-۱۴ الگوریتم‌های زمان‌بندی و نظارت

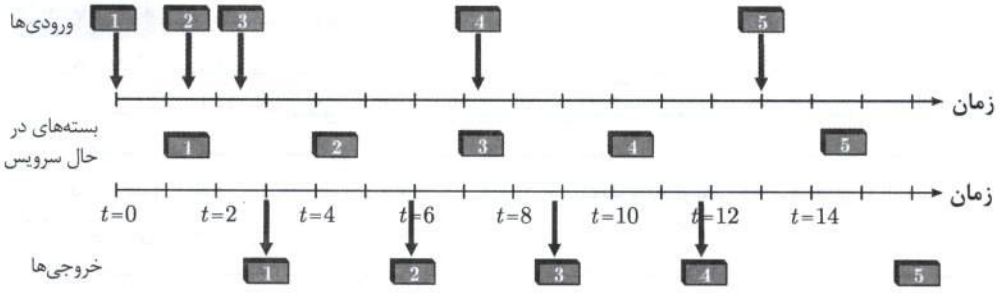
اکنون که نیاز تأمین خدمات با کلاس‌های مختلف بررسی شد به جزئیات دو روش مهم زمان‌بندی و نظارت می‌پردازیم. الگوریتم‌های زمان‌بندی از آن جهت دارای اهمیت می‌باشند که بسته‌های متعلق به جریان‌های گوناگون شبکه تسهیم و صف‌بندی می‌شوند و سپس به منظور انتقال در اختیار بافرهای خروجی وابسته به یک پیوند قرار می‌گیرند. روش صف‌بندی بسته‌ها به‌منظور انتقال از طریق یک پیوند اصطلاحاً **نظام زمان‌بندی پیوند** نامیده شده است. در ادامه چندین نظام زمان‌بندی پیوند مهم تحلیل می‌شوند.

۱- الگوریتم زمان‌بندی FIFO

شکل ۷-۲۱ مراحل صف‌بندی براساس نظام زمان‌بندی پیوند FIFO را نشان می‌دهد. بسته‌های وارد شده در صف خروجی پیوند در صورتی در انتظار انتقال قرار می‌گیرند که پیوند مشغول انتقال بسته‌ی دیگری باشد. در صورت عدم وجود فضای ذخیره‌سازی جهت انباشت بسته‌های ورودی صف از امکانات **سیاست حذف بسته**[1] به منظور اینکه آیا بسته بایستی حذف (تلف) شود و یا آنکه بسته‌های دیگری از صف خارج می‌شوند تا فضای کافی به‌منظور پذیرش بسته جاری تأمین شود، بهره می‌گیرد. در بحث ذیل ما از سیاست حذف بسته صرف‌نظر می‌کنیم. هنگامی که یک بسته از طریق پیوند خروجی به‌طور کامل انتقال می‌یابد آن بسته از صف حذف می‌شود.



شکل ۷-۲۱ ♦ نظام صف‌بندی FIFO



شکل ۷-۲۲ ♦ عملیات صف FIFO

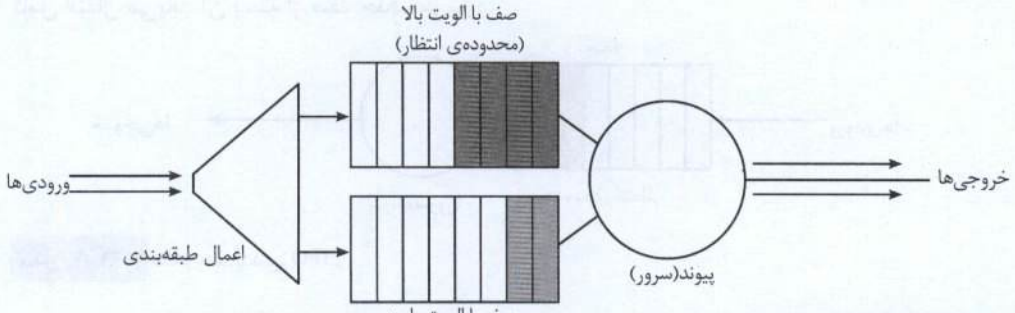
در نظام زمان‌بندی FIFO گزینش بسته‌ها برای انتقال از طریق پیوند به همان ترتیبی صورت می‌پذیرد که آن‌ها وارد پیوند خروجی صف شده‌اند. شکل ۷-۲۲ عملیات صف FIFO را نشان می‌دهد. ورود بسته‌ها از طریق پیکان شماره‌گذار خط زمانی مشخص می‌شود. عدد نشان داده شده در خط

1- Packet discarding policy

زمانی ترتیب ورود بسته‌ها به صف را نشان می‌دهد. خروج انفرادی هر بسته در ذیل خط زمان نشان داده شده است. زمان لازم جهت انتقال هر بسته از طریق یک چهارگوش سایه‌دار در بین دو خط زمانی نشان داده شده است. از آن‌جائیکه در نظام FIFO خروج بسته‌ها به همان ترتیب ورود آن‌ها می‌باشد، پس از خروج بسته‌ی شماره‌ی 4 پیوند در حالت آزاد قرار می‌گیرد زیرا بسته‌های 1 تا 4 انتقال یافته و در نتیجه از صف حذف می‌شوند. حالت آزاد بودن پیوند تا ورود بسته‌ی 5 ادامه می‌یابد.

۲- الگوریتم صف‌بندی اولویت‌دار[1]

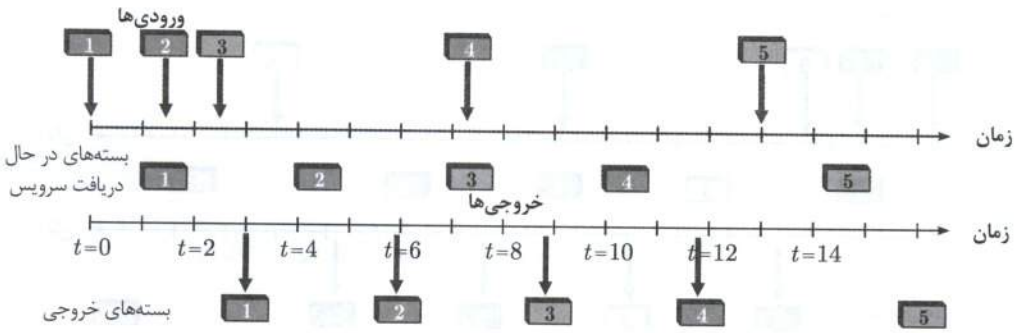
در نظام صف‌بندی الویت‌دار بسته‌های ورودی در پیوند خروجی در قالب مراتب اولویت آن‌ها در صف قرار می‌گیرند. شکل ۷-۲۳ نشان می‌دهد چگونه بسته‌ها در دو گروه صف با مرتبه‌ی بالا و پائین در صف‌های مربوط قرار گرفته‌اند. قبلاً اشاره گردید مرتبه‌ی الویت بسته انحصاراً وابستگی به علامت گذاری انجام شده در سرپیام بسته مانند ارزش بیت‌های TOS در یک بسته‌ی IPV4 دارد. علاوه براین موارد مرتبه‌ی الویت به منبع یا آدرس IP مقصد، شماره پورت مقصد و یا عوامل ساختاری بسته نیز وابسته می‌باشد و هر مرتبه‌ی الویتی نیز صف ویژه‌ی خود را دارد. هنگامی که یک بسته جهت انتقال گزینش می‌شود الگوریتم صف‌بندی الویت‌دار یک بسته را از بالاترین مرتبه‌ی صفی که بسته‌هایی را جهت انتقال در انتظار دارد، انتقال می‌دهد. گزینش بسته‌ها در درون یک صف با الویت مشابه برمبنای FIFO انجام می‌شود.



شکل ۷-۲۳ ♦ الگوی صف‌بندی الویت‌دار

شکل ۷-۲۴ عملیات مربوط به یک صف الویت‌دار را نشان می‌دهد که از دو مرتبه‌ی الویتی بهره می‌گیرد. بسته‌های 1، 3 و 4 متعلق به گروه با الویت بالا می‌باشند و بسته‌های 2 و 5 متعلق به مرتبه‌ی پائین‌تر هستند. هنگامی که بسته‌ی 1 وارد صف می‌شود پیوند آزاد است و در نتیجه انتقال آغاز می‌شود. در طول انتقال بسته‌ی 1، بسته‌های 2 و 3 وارد می‌شوند و به ترتیب در صف‌های با الویت پائین و بالاتر قرار می‌گیرند. پس از گسیل بسته‌ی 1، بسته‌ی شماره‌ی 3 که دارای بالاترین الویت است گزینش شده و آن‌گاه بسته‌ی 2 اگرچه این بسته قبل از بسته‌ی شماره‌ی 3 وارد صف با الویت پائین شده است. در انتهای انتقال بسته‌ی 3، گسیل بسته‌ی 2 انجام می شود. بسته‌ی شماره 4 با الویت بالا هنگام انتقال بسته‌ی 2 وارد شده است. براساس الگوریتم صف‌بندی الویت‌دار هنگامی که انتقال بسته‌ی شماره 1 آغاز شود، دیگر توقف ناپذیر است در این شرایط بسته‌ی 4 در صف انتقال قرار می‌گیرد و گسیل آن هنگامی آغاز می‌شود که انتقال بسته‌ی 2 کامل شده است.

1- Priority queuing



شکل ۷-۲۴ ♦عملیات صف الویت‌دار

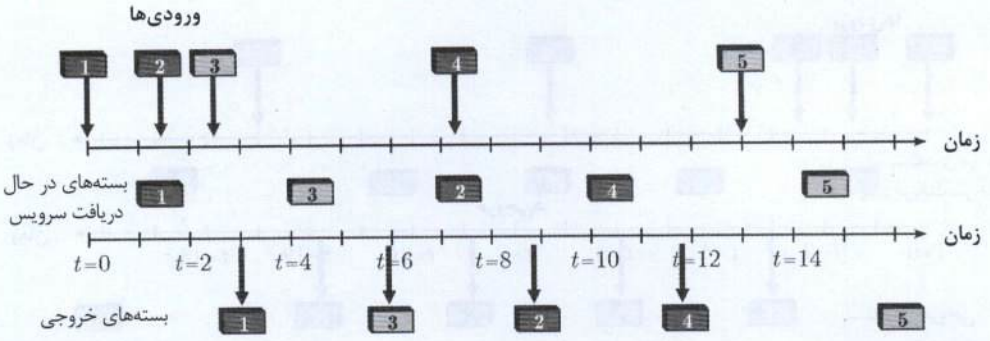
۳- الگوریتم صف‌بندی توزین عادلانه (WFQ)

در نظام صف‌بندی توزین عادلانه که اصطلاحاً **صف‌بندی گردش نوبتی**[1] نیز خوانده می‌شود، بسته‌ها برمبنای مرتبه مانند صف الویت‌دار طبقه‌بندی می‌شوند. تنها تفاوت این الگوریتم با صف الویت‌دار آن است که به جای رعایت الویت بدون انعطاف در بین کلاس‌ها در WFQ از روش گردش نوبتی در بین مرتبه‌های بسته‌ها استفاده می‌شود. به بیان ساده‌تر در نظام گردش نوبتی، یک بسته با مرتبه‌ی 1 گسیل می‌شود و در پی آن بسته‌ی کلاس 2 و به دنبال آن یک بسته‌ی با الویت 1 و سپس مرتبه‌ی 2 و غیره گسیل می‌شوند. در این روش یک پیوند هرگز در صورت وجود بسته با هر مرتبه‌ای در صف جهت انتقال بیکار نخواهد ماند. این الگو اصطلاحاً **الگوریتم گردش نوبتی صرفه‌جو در کار** نیز نامیده شده است. علت این نام‌گذاری به دلیل آن است که این روش به کاوش یک بسته با الویت داده شده‌ای می‌پردازد، هنگامی که چنین بسته‌ای را پیدا نکند بی‌درنگ کلاس بعدی را در قالب گردش نوبتی جستجو می‌کند.

شکل ۷-۲۵ عملیات یک صف گردش نوبتی دو الویتی را نشان می‌دهد. در این مثال بسته‌های 1، 2 و 4 متعلق به کلاس یک و بسته‌های 3 و 5 مربوط به کلاس 2 می‌باشند. انتقال بسته‌ی 1 بی‌درنگ پس از ورود آن در صف خروجی انجام می‌شود. بسته‌های 2 و 3 در هنگام گسیل بسته‌ی 1 وارد شده و در نوبت انتقال قرار می‌گیرند. پس از انتقال بسته‌ی 1 الگوریتم زمان‌بندی به جستجوی بسته‌ی با مرتبه‌ی 2 پرداخته و بسته‌ی 3 را انتقال می‌دهد. پس از گسیل بسته‌ی شماره 3 به جستجوی بسته‌ی با الویت 1 می‌پردازد و بنابراین بسته‌ی شماره 2 را گسیل می‌نماید. پس از ارسال بسته‌ی شماره 2 تنها بسته‌ی باقی مانده در صف شماره‌ی 4 می‌باشد و این بسته بی‌درنگ پس از 2 گسیل می‌شود.

حالت کلی‌تر این الگوریتم که در معماری‌های QoS به گونه‌ی گسترده مورد بهره‌برداری قرار گرفته است **صف‌بندی توزیع عادلانه (WFQ)**[2] نامیده شده است. شکل ۷-۲۶ الگوریتم WFQ را نشان می‌دهد. بسته‌های ورودی برمبنای محدوده‌ی انتظار مناسب خود طبقه‌بندی و در صف قرار می‌گیرند.

1- Round robin queuing
2- Weighted fair queuing



شکل ۷-۲۵ ♦ عملیات صف گردش نوبتی با دو الویت

مانند زمان‌بندی گردش نوبتی، الگوریتم WFQ کلاس‌ها را به روش گردشی خدمات رسانی می‌کند. به بیان دیگر ابتدا بسته‌های مرتبه‌ی یک، سپس کلاس 2 و پس از آن درجه‌ی 3 و اگر فرض شود تنها سه مرتبه وجود دارد الگوی گردش مجدداً تکرار می‌شود. الگوریتم WFQ هم‌چنین یک نظام صف‌بندی صرفه‌جو در کار می‌باشد و بنابراین بی‌درنگ به کلاس بعدی پس از اطمینان از تهی بودن یک کلاس مراجعه می‌کند.

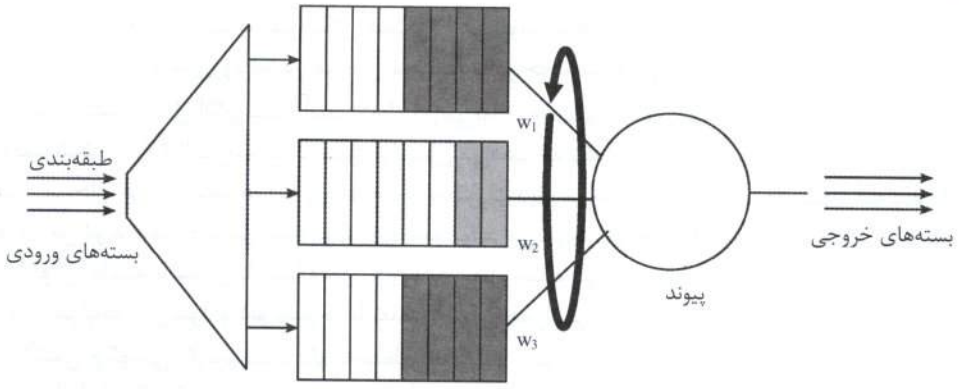
تفاوت WFQ با نظام گردش نوبتی در آن است که هر کلاس ممکن است دیفرانسیلی از مقدار خدمات را در هر فاصله‌ی زمانی دریافت نماید. اگر به هر کلاس i وزنی معادل w_i اختصاص پیدا کند، در الگوریتم WFQ در هر فاصله زمانی که بسته‌های کلاس i جهت ارسال وجود دارند، مرتبه‌ی i این اطمینان را ایجاد می‌کند که بخشی از خدمات به آن کلاس اختصاص می‌یابد. این مقدار برابر است با:

$$\frac{w_i}{(\sum w_j)} \qquad (۸-۷)$$

در این رابطه حاصل جمع در مخرج کسر شامل همه‌ی کلاس‌هایی می‌شود که بسته در صف جهت گسیل دارند. در حادترین شرایط حتی اگر همه‌ی کلاس‌ها بسته در صف داشته باشند، بسته‌های مرتبه‌ی i هنوز هم به دریافت مقدار اشاره شده در بالا از پهنای باند کلی پیوند ادامه خواهند داد. بنابراین برای یک پیوند با سرعت انتقال R یک بسته‌ی کلاس i همواره حداقل نهفتگی برابر ذیل را اختیار می‌کند.

$$\text{حداقل نهفتگی بسته‌ی کلاس i} = R\frac{w_i}{(\sum w_j)} \qquad (۹-۷)$$

بدیهی است توصیف انجام شده برای WFQ بسیار ایده‌آل می‌باشد زیرا این واقعیت درنظر گرفته نشده است که بسته‌ها عناصر گسسته‌ی داده‌ها هستند و گسیل بسته با آغاز انتقال بسته‌ی دیگر دچار توقف نمی‌شود. در ادامه‌ی مطالب فصل ملاحظه خواهد شد که الگوریتم WFQ نقش بسیار مهمی را در معماری QoS بازی می‌کند و اغلب مسیریاب‌های فعلی از این امکانات بهره‌برداری می‌کنند.



شکل ۷-۲۶ ♦صف‌بندی توزین عادلانه (WFQ)

- **کنترل و نظارت: دَلو نشت کننده**

یکی از نتیجه‌گیری‌های انجام شده در بخش قبل تنظیم سرعتی است که یک کلاس یا جریان از طریق آن اجازه می‌یابد بسته‌ها را به درون شبکه تزریق نماید. اعمال این سیاست یکی از مهم‌ترین روش‌های QoS می‌باشد. پرسش آن است که چه اصولی از سرعت جریان بسته بایستی نظارت شوند؟ سه مورد مهم را در این ارتباط می‌توان شناسایی نمود هریک از این سه بنابر مقیاس زمانی که بسته جاری می‌شود نسبت به یکدیگر متفاوت می‌باشند:

۱- **میانگین سرعت.** شبکه ممکن است مایل به محدود نمودن میانگین سرعتی باشد که بسته‌ها در درازمدت جاری می‌شوند. این میانگین عبارت است از فاصله زمانی بین ورود هر بسته و سپس ارسال آن به درون شبکه. یک موضوع مهم در این رابطه فاصله‌ی زمانی است که در آن میانگین سرعت نظارت می‌شود. جریانی که میانگین سرعت آن محدود به تعداد 100 بسته در هر ثانیه شده است با فشار بیش‌تری در مقایسه با منبعی که محدود به 6000 بسته در هر دقیقه است روبه‌رو می‌باشد اگرچه هر دو از میانگین سرعت مشابه در فاصله‌ی زمانی در حد کافی طولانی برخوردار می‌باشند. به عنوان مثال محدودیت آخر اجازه می‌دهد یک جریان، تعداد 1000 بسته در یک فاصله‌ی زمانی که نسبتاً طولانی است گسیل نماید در حالیکه محدودیت پیشین آن مانع چنین عملکردی می‌شود.

۲- **حداکثر سرعت.** در حالیکه میانگین سرعت مقدار ترافیک ارسالی به درون شبکه را در یک فاصله‌ی زمانی نسبتاً طولانی محدود می‌کند. رویکرد دوم حداکثر مقدار بسته‌ی قابل ارسال در یک فاصله‌ی زمانی کوتاه را با چالش روبه‌رو می‌کند. با استفاده از مثال بالا شبکه ممکن است کنترل جریان را در میانگین سرعت 6000 بسته در دقیقه نظارت نماید در حالیکه حداکثر سرعت، جریان را محدود به 1500 بسته در ثانیه می‌نماید.

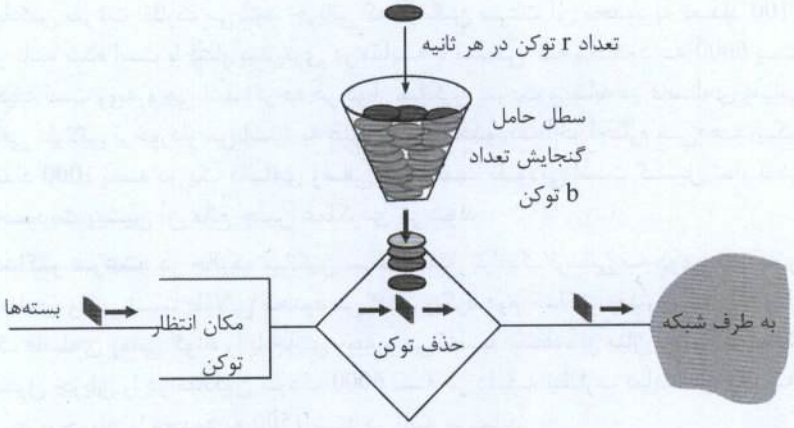
۳- **حجم بسته.** شبکه ممکن است مایل به محدود نمودن حداکثر تعداد بسته‌ها در هر جریان باشد که بایستی به درون شبکه در یک فاصله‌ی زمانی فوق‌العاده کوتاهی ارسال گردد. هرچه حد طول فاصله‌ی زمانی به صفر نزدیک‌تر شود تعداد بسته‌هایی را که می‌توان به صورت لحظه‌ای به درون شبکه گسیل نمود محدود می‌کند. اگرچه در عمل ارسال لحظه‌ای چندین

بسته به درون شبکه غیرممکن است زیرا هر پیوند شبکه دارای سرعت انتقال فیزیکی است که از آن فراتر نمی‌رود اما در هر حال محدودیت در حجم بسته می‌تواند مفید باشد.

روش نشت سطل الگوریتمی است که از آن می‌توان به منظور اعمال این سیاست‌گذاری‌ها بهره گرفت. شکل ۷-۲۷ الگوریتم نشت سطل را نشان می‌دهد که شامل یک دَلو با حداکثر تعداد b توکن است. مراحل افزودن نشانه‌ها (توکن‌ها) به دَلو به‌صورت ذیل می‌باشد. نشانه‌های جدیدی که به دلو اضافه می‌شوند همواره با سرعت r توکن در ثانیه تولید می‌شوند. اگر تعداد نشانه‌های حاضر در دَلو کمتر از b توکن داشته باشد آن‌گاه نشانه‌های جدید به سطل اضافه می‌شوند در غیراین صورت از توکن‌های جدید صرف‌نظر می‌شود و دَلو نشانه‌ها با تعداد b توکن همراه پر باقی می‌ماند.

اکنون چگونگی کاربرد نشت دَلو به منظور نظارت و کنترل جریان بسته‌ها بررسی می‌شود. اگر فرض شود قبل از انتقال یک بسته به درون شبکه آن بسته الزام به خارج نمودن یک نشانه از ظرف توکن‌ها باشد و اگر دَلو نشانه‌ها خالی باشد، بسته بایستی در انتظار یک توکن قرار گیرد و یا آن که بسته حذف و از آن صرف‌نظر شود. رفتار عملیات کنترل و نظارت جریان بسته‌ها را می‌توان چنین تحلیل نمود: از آن‌جائیکه حداکثر b نشانه می‌تواند در ظرف باشد حداکثر اندازه‌ی حجم بسته‌ها برای یک ناظر جریان دَلو سوراخ برابر b بسته می‌باشد. علاوه بر آن از آنجایی‌که تولید نشانه‌ها با نرخ r برثانیه است حداکثر تعداد بسته‌هایی که می‌تواند وارد شبکه در هر فاصله‌ی زمانی t ثانیه شود برابر rt+b خواهد بود. بنابراین سرعت تولید نشانه‌ها یعنی r در خدمت محدود نمودن درازمدت میانگین سرعتی است که بسته‌ها می‌توانند به شبکه وارد شوند. این امکان نیز وجود دارد که از سیاست نشت سطل به‌گونه‌ی دو دَلو نشت سری به‌منظور نظارت بر کنترل حداکثر سرعت جریان بسته‌ها علاوه بر میانگین سرعت درازمدت بهره گرفت. به بیان دیگر می‌توان به نتیجه‌گیری ذیل دست یافت:

حداکثر تأخیر قابل اثبات در صف = نشت سطل + صف‌بندی توزین عادلانه (WFQ)



شکل ۷-۲۷ ♦ ناظر دَلو نشت‌کننده

به منظور تأمین کیفیت خدمات (QoS) مطلوب در اینترنت در ادامه‌ی مطالب فصل با دو رویکرد اصطلاحاً Intserv و Diffserv آشنا خواهیم شد و نقش حیاتی دو الگوریتم WFQ و سطل نشت‌دار را در این رویکردها مشاهد خواهیم نمود. پایان بخش مطالب این زیربخش بررسی پیوند خروجی یک روتر است که تعداد n جریان را تسهیم می‌کند و هریک از جریان‌ها از سوی یک سطل نشت کننده کنترل و

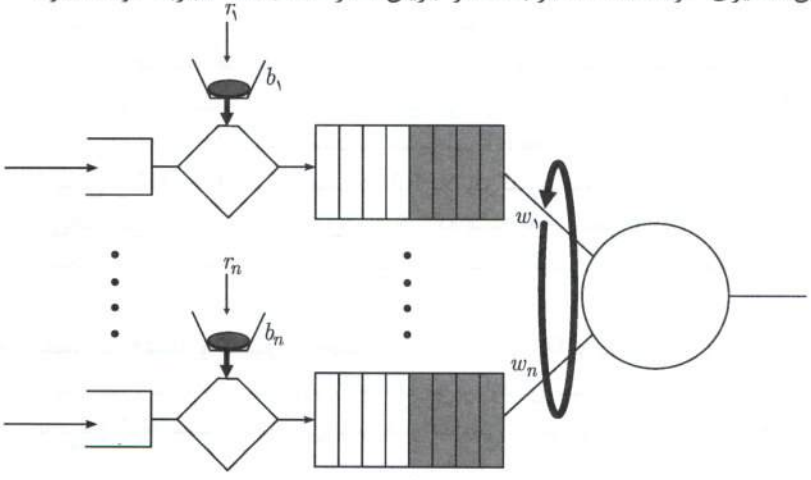
نظارت می‌شوند که از عوامل b_i و r_i بهره می‌گیرد. ارزش‌های i از طریق الگوریتم WFQ در قالب i=1 , ... , n تعریف می‌شود. در این بررسی از عبارت **جریان** به منظور اشاره به مجموعه‌ای از بسته‌ها که از یکدیگر تمایز ناپذیر در قبال بهره‌گیری از روش زمان‌بندی می‌باشند، استفاده می‌شود. در عمل یک جریان می‌تواند شامل ترافیکی از یک ساختار انتها به انتها و یا مجموعه‌ای از تعداد زیادی ارتباط باشد. شکل ۷-۲۸ این موضوع را نشان می‌دهد.

با یادآوری بحث الگوریتم WFQ که هر جریان مانند i اطمینان از دریافت سهمی از پهنای باند پیوند را دارد که حداقل برابر است با: $R\frac{w_i}{\sum w_j}$ و R نیز سرعت ارسال پیوند بر مبنای تعداد بسته در ثانیه می‌باشد، آن‌گاه حداکثر تأخیری که یک بسته هنگام انتظار برای دریافت خدمات در WFQ و به عبارت دیگر پس از عبور کنترل الگوریتم سطل نشت کننده تجربه می‌نماید چقدر است؟ با تمرکز بر روی جریان 1 فرض کنید سطل نشت کننده‌ی این جریان در آغاز پر می‌باشد.

زنجیره‌ای از بسته‌ها به ناظر سطل نشت کننده‌ی جریان 1 وارد می‌شوند. این بسته‌ها همه‌ی نشانه‌ها را از دلو خارج می‌کنند و سپس به منطقه‌ی انتظار الگوریتم WFQ برای جریان 1 وارد می‌شوند. از آن‌جا که این تعداد bi بسته با حداقل سرعت معادل $R\frac{w_i}{w_j}$ بسته در ثانیه خدمات‌رسانی می‌شوند، آن‌گاه آخرین بسته از این گروه دارای حداکثر تأخیر d_{max} خواهد شد که رابطه (۷-۹) آن را تعریف می‌کند تا انتقال کامل گردد.

$$d_{max} = \frac{b_1}{(Rw_1 / \sum w_j)} \qquad (۷-۱۰)$$

استدلال پشت پرده‌ی رابطه (۷-۱۰) آن است که اگر تعداد b_1 بسته در صف وجود دارد و بسته‌ها در حال خارج شدن از صف با سرعت حداقل $\frac{Rw_i}{\sum w_j}$ بسته بر ثانیه می‌باشند آن‌گاه طول زمان برای آن که آخرین بیت از آخرین بسته انتقال یابد نمی‌تواند بیش‌تر از آن‌چه رابطه‌ی (۷-۱۰) نشان می‌دهد، در نظر گرفته شود. بدیهی است هنگامی که مقدار r کمتر از رابطه‌ی (۷-۹) باشد آن‌گاه d_{max} نیازمند حداکثر زمان تأخیری خواهد شد که هر بسته در جریان 1 در صف WFQ تجربه خواهد نمود.



شکل ۷-۲۸ ♦تعداد **n** جریان مالتی‌پلکس سطل نشت کننده با الگوریتم WFQ

۷-۱۵ رویکرد Diffserv

معماری **تمایز خدمات اینترنت**[1] **(Diffserv)** توصیف شده در مستند [RFC2475] با هدف توان‌مندی به‌کارگیری گروه‌ها یا کلاس‌های متفاوت ترافیکی در درون شبکه‌ی اینترنت به‌منظور اعمال انعطاف‌پذیری و مقیاس‌پذیری شبکه تعریف شده است. مقیاس پذیری از آن جهت موردنیاز شبکه می‌باشد که صدها هزار منبع ترافیکی مبدأ و مقصد ممکن است در مسیریاب‌های زیرساخت شبکه‌ی اینترنت حاضر باشند. در ادامه‌ی موضوعات این بخش ملاحظه خواهد شد که تحقق این نیاز به‌سادگی از طریق قرار دادن یک کارکرد ساده در درون هسته‌ی شبکه امکان‌پذیر می‌شود در حالیکه عملیات کنترل پیچیده‌تر این نیاز در لبه یا نبش شبکه و به عبارت دیگر سیستم‌های انتهایی پیاده‌سازی می‌شوند. انعطاف‌پذیری در شبکه‌ی اینترنت نیز از آن جهت مورد نیاز است که ظهور گروه‌های خدماتی نوین و منسوخ شدن گروه‌های خدماتی پیشین همواره امکان‌پذیر و قطعی می‌باشد. معماری Diffserv از آن جهت انعطاف‌پذیر به شمار می‌آید که به تعریف خدمات ویژه و یا گروه‌های خدماتی نمی‌پردازد بلکه عناصر عملیاتی متعلق به یک نیاز را تأمین می‌نماید. به بیان دیگر بخش‌هایی از معماری شبکه را تأمین می‌کند که با بهره‌گیری از آن‌ها ساخت و تعریف چنین خدماتی امکان‌پذیر می‌شود. در ادامه این عناصر به شکل دقیق‌تر آن‌ها ارزیابی می‌شوند:

۱- خدمات متمایز برای یک سناریوی ساده

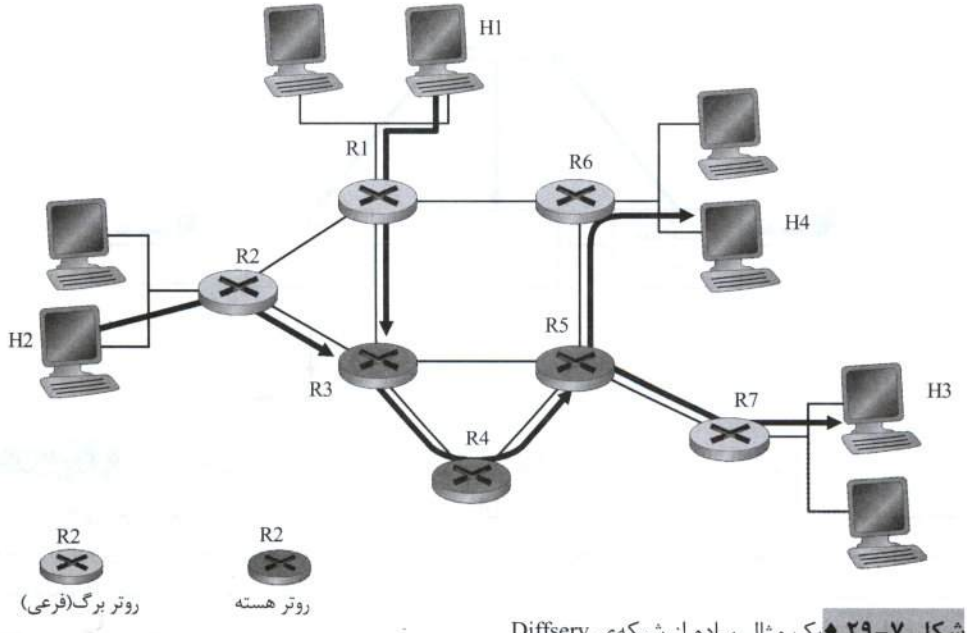
به‌منظور تعیین چارچوبی جهت تعریف الگوی عناصر معماری خدمات متمایز (Diffserv) از یک شبکه‌ی ساده‌ی نشان داده شده در شکل ۷-۲۹ بهره‌گیری می‌شود. در این بخش یکی از موارد کاربرد عناصر Diffserv توصیف می‌شود. بدیهی است بنابر مطالب مستند [RFC2475] بسیاری دیگر از این عوامل نیز امکان‌پذیر می‌باشند. هدف اصلی این بخش معرفی عوامل کلیدی Diffserv می‌باشد و قصد ندارد به توصیف عمیق الگوی معماری Diffserv بپردازد. معماری Diffserv شامل دو مجموعه عناصر عملیاتی است:

- **عملیات لبه شامل گروه‌بندی بسته و کنترل ترافیک.** در لبه‌ی ورودی شبکه که میزبان‌های توان‌مند در ارایه‌ی خدمات Diffserv ترافیک را تولید می‌کنند و یا در روتر نخست نبش شبکه که با بهره‌گیری از توان‌مندی Diffserv ترافیک جاری را از درون خود عبور می‌دهد بسته‌های ورودی علامت‌گذاری می‌شوند. دامنه‌ی تمایز خدمات (DS) در سر پیام بسته با تعریف یک ارزش مقداردهی می‌شود. به عنوان مثال در شکل ۷-۲۹ بسته‌های ارسالی از H1 به H3 ممکن است در مسیریاب R1 علامت‌گذاری شوند در حالیکه بسته‌های ارسالی از H2 به H4 ممکن است در R2 علامت گذاری شده باشند. علامتی که یک بسته دریافت می‌دارد گروه و یا کلاس ترافیکی که بسته به آن تعلق دارد را شناسایی می‌کند و در نتیجه‌ی آن گروه‌های متفاوت ترافیکی خدمات گوناگونی را در درون هسته‌ی شبکه دریافت می‌دارند.
- **عملیات هسته و گسیل به جلو.** هنگامی که یک DS به علامت‌گذاری بسته‌ی ورودی در یک مسیریاب قادر به عملیات Diffserv می‌پردازد آن بسته به نقطه‌ی کنترل ترافیک بعدی بنابرنوع گروه بسته گسیل می‌شود. رفتار عملیاتی روتر دریافت کننده از لحاظ چگونگی

1- Service differentiation

ذخیره‌سازی و پهنای باند پیوند مشترک مابین گروه ترافیک‌های رقیب متأثر از این رفتار عملیاتی و در نتیجه ارزش دامنه‌ی DS می‌باشد. یکی از ویژگی‌های برجسته‌ی معماری Diffserv آن است که رفتار مسیریاب فقط برمبنای علامت گذاری انجام شده بر روی بسته می‌باشد به عبارت دیگر تأثیرپذیر از گروه ترافیکی است که بسته به آن تعلق دارد. بنابراین اگر بسته‌های ارسالی از H1 به H3 در شکل ۷-۲۹ همان علامت‌گذاری را دریافت نمایند که بسته‌های گسیل شده از H2 به H4 آن‌گاه برخورد مسیریاب‌های شبکه با این بسته‌ها به صورت جمعی می‌باشد و هیچ‌گونه تمایزی از اینکه آیا سرچشمه‌ی بسته H1 و یا H2 است وجود ندارد. به عنوان مثال در شکل ۷-۲۹ مسیریاب R3 تمایزی را بین بسته‌های گسیل شده از H1 و H2 هنگام گسیل آن‌ها به R4 نمی‌گذارد.

بنابراین معماری Diffserv نیاز برای نگهداری وضعیت جفت روترهای مبدأ و مقصد را برطرف می‌سازد که یکی از مهم‌ترین نیازهای ایجاد مقیاس‌پذیری در شبکه می‌باشد و در ابتدای مطالب این بخش به آن اشاره رفت. جهت شفافیت بیش‌تر موضوع به این نکته اشاره می‌شود که معمولاً در رخدادهای اجتماعی بزرگ و با اهمیت از نوعی پروانه‌ی عبور استفاده می‌شود به عنوان مثال در یک کنسرت از کارت‌های ویژه جهت مشخص نمودن اهمیت حاضران بهره‌گیری می شود و یا در هنگام ورود افراد به نشست‌های مهم از کارت‌های VIP استفاده می‌شود. این نوع مجوزهای عبور معمولاً در هنگام ورود به محل برگزاری یک رخداد یعنی در لبه یا نبش آن رخداد صادر می‌شود. در این نقطه است که معمولاً عملیات محاسباتی حیاتی مانند پرداخت جهت ورود، بازرسی نوع مناسب دعوت و نگاشت یک دعوت‌نامه در برابر یک اطلاعات شناسنامه‌ای صورت می‌پذیرد.



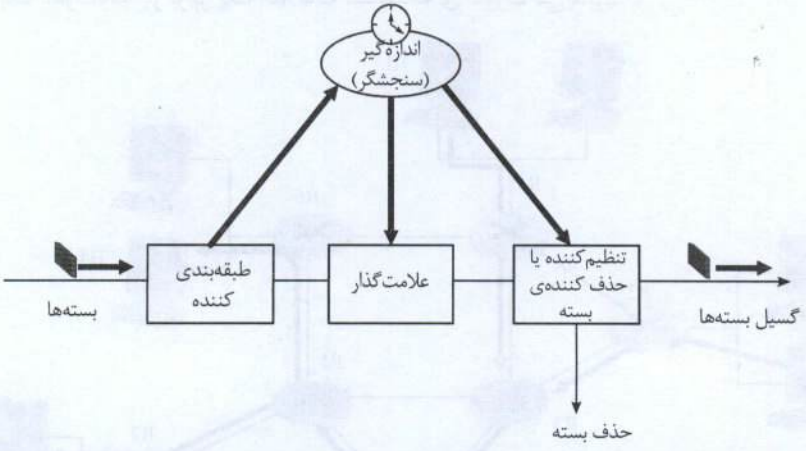
شکل ۷-۲۹ یک مثال ساده از شبکه‌ی Diffserv

علاوه بر این موارد ممکن است محدودیت‌هایی از نقطه نظر تعداد افراد با ویژگی‌های معین در یک رخداد نیز اعمال گردد. در صورت وجود چنین محدودیت‌هایی افراد ممکن است مجبور به انتظار قبل از

ورود به یک رخداد باشند. هنگامی که افراد وارد محل وقوع رخداد می‌شوند مجوز عبور فردی این اجازه را برای فرد فراهم می‌آورد تا نسبت به دریافت خدمات متمایز در بسیاری از نقاط در اطراف آن رخداد بهره‌گیری نماید. به عنوان مثال یک فرد با مشخصه‌ی VIP از بسیاری از خدمات مجانی در محل رخداد می‌تواند استفاده نماید در حالیکه یک فرد با شناسه‌ی معمولی اجازه ورود به برخی از مکان‌های محل رخداد و یا بهره‌گیری از خدمات ویژه را ندارد و تنها خدمات قابل استفاده به وسیله‌ی وی امکانات پایه‌ای می‌باشد. در هر دو مورد خدمات دریافت شده وابستگی انحصاری به نوع مجوز فرد دارد. بدیهی است همه‌ی افراد در درون یک گروه از لحاظ برخورد یکسان می‌باشند.

۲- طبقه‌بندی ترافیک Diffserv و اعمال کنترل آن

شکل ۷-۳۰ یک نماد منطقی از گروه‌بندی و علامت‌گذاری درون نبش مسیریاب را نشان می‌دهد. بسته‌های وارد شده به روتر لبه ابتدا گروه‌بندی می‌شوند. چگونگی گروه‌بندی بسته‌ها برمبنای ارزش درون یک و یا چند دامنه در سر پیام بسته دارد. به عنوان مثال دامنه‌های آدرس‌های مبدأ و مقصد، درگاه‌های مبدأ و مقصد و یا شناسه‌ی (ID) پروتکلی بسته. این ارزش‌ها سبب هدایت بسته به عملیات مناسب علامت گذاری می‌شوند. علامت یک بسته معمولاً در درون دامنه‌ی DS آن بر مبنای مستند [RFC3260] و در سرپیام بسته‌ی IPV4 و یا IPV6 حمل می‌شوند. تعریف دامنه‌ی DS فراتر از تعاریف درج شده در دامنه‌های نوع خدمات IPV4 و یا IPV6 حمل می‌شوند. بدیهی است تعریف دامنه‌ی DS فراتر از تعاریف درج شده در دامنه‌های نوع خدمات IPV4 و یا دامنه‌ی گروه ترافیک IPV6 می‌باشد که در فصل ۴ به آن‌ها اشاره رفت.



شکل ۷-۳۰ ♦ نگاه منطقی گروه‌بندی بسته و اعمال کنترل ترافیک در مسیریاب لبه یا انتهایی

در برخی موارد یک کاربر انتهایی ممکن است توافق خود را در ارسال بسته‌ها محدود به سرعت معینی از سوی یک ترافیک تعریف شده نماید. این ترافیک تعریف شده ممکن است شامل یک محدودیت در حداکثر سرعت انتقال و نیز نرخ تناوب بسته بر مبنای آن‌چه در مورد الگوریتم دلو نشت‌دار قبلاً توضیح داده شد، گردد. تا زمانی که کاربر، بسته‌های ارسالی به درون شبکه را با روشی انجام دهد که در توافق با مشخصه‌ی ترافیکی تعریف شده‌ای می‌باشد آن‌گاه بسته‌ها علامت‌گذاری اولویت خود را برمبنای توافق انجام شده و دریافت و گسیل آن‌ها نیز در امتداد مسیر تعیین شده به

مقصد انجام می‌شود. چنان‌چه مشخصه‌ی ترافیک تعریف شده از سوی بسته‌ها رعایت نگردد، این‌گونه بسته‌های خارج از چارچوب دارای علامت گذاری متفاوتی خواهند شد و امکان اعمال تأخیر به منظور رعایت حداکثر سرعت در آن‌ها نیز وجود دارد. این امکان وجود دارد که این گونه بسته‌ها در نبش شبکه دچار حذف گردند. نقش **عملیات اندازه‌گیری**[1] در شکل ۷-۳۰ به منظور مقایسه‌ی نرخ جریان ورود بسته‌ها با اندازه‌ی مشخصه ترافیکی و تعیین این موضوع که آیا این نرخ در چارچوب توافق انجام گرفته قرار دارد یا نه می‌باشد. تصمیم‌گیری قطعی در مورد اینکه یک بسته در معرض تغییر علامت گذاری، گسیل، تأخیر و یا حذف قرار گیرد موضوع سیاست‌گذاری شبکه است که از سوی مدیر شبکه تعیین خواهد شد و ارتباطی با معماری Diffserv ندارد.

۳- الگوی رفتار انفرادی مسیریاب

تاکنون تمرکز موضوع بر عملیات شبکه در معماری Diffserv قرار گرفته بود. عنصر کلیدی دوم معماری Diffserv شامل موضوعی می‌باشد که از سوی مسیریاب‌های مجهز به امکانات Diffserv انجام می‌شود. این موضوع اصطلاحاً **رفتار انفرادی مسیریاب**[2] **(PHB)** نام دارد. یک روتر PHB را این گونه می‌توان تعریف نمود: توصیفی از مشاهده‌ی خارجی رفتار گسیل بسته در یک گره‌ی Diffserv بر مبنای مستند [RFC2475].

با مطالعه‌ی عمیق این تعریف چند ویژگی ذاتی مهم را می‌توان به شرح ذیل شناسایی نمود:

- یک PHB می‌تواند سبب دریافت کارایی‌های گوناگون از سوی گروه‌های مختلف ترافیکی گردد. به بیان دیگر رفتار گسیل مختلف قابل مشاهده خارج از مسیریاب.
- اگرچه یک PHB تفاوت‌های کارایی یا به عبارت دیگر رفتاری را در بین گروه‌های ترافیکی تعریف می‌کند اما به اعمال روش ویژه‌ی اجباری برای دستیابی به این رفتارها نمی‌پردازد. تا زمانی که کارایی قابل مشاهده از خارج امکان‌پذیر باشد هر الگوی پیاده‌سازی و یا سیاست اختصاص پهنای باند و بافر می‌تواند مورد استفاده قرار گیرد. به عنوان مثال یک PHB نیازمند یک نظام ویژه‌ی صف‌بندی بسته مانند یک صف اولویت‌دار در برابر WFQ و یا FCFS جهت دستیابی به رفتاری ویژه نمی‌باشد. یک PHB نقطه‌ی انتهایی اختصاص منابع و الگوهای پیاده‌سازی به‌شمار می‌آید.
- تفاوت در کارایی بایستی قابل مشاهده و اندازه‌گیری باشد.

در حال حاضر دو نوع الگوی PHB تعریف شده است. این دو عبارتند از الگوی **گسیل تسریع شده**[3] **(EF)** و الگوی **گسیل ضمانت شده**[4] **(AF)** که به ترتیب در دو مستند [RFC3246] و [RFC2597] تعریف شده‌اند:

الف) گسیل تسریع شده (EF). یک PHB از نوع EF به تعریف سرعت ارسال یک گروه ترافیکی از یک روتر می‌پردازد که آن سرعت بایستی برابر و یا فراتر از نرخ پیکربندی شده باشد. به عبارت دیگر در طول هر تناوب زمانی گروه ترافیکی مورد نظر دارای این ضمانت می‌شود که پهنای باند کافی به منظور آنکه سرعت ترافیک خروجی برابر و یا فراتر از حداقل نرخ پیکربندی شده است را در اختیار

1- Metering function
2- Per hop behavior
3- Expedited forwards
4- Assured forwards

دارد. بدیهی است الگوی EF نوعی جداسازی بین گروه‌های ترافیکی تعریف می‌کند زیرا ضمانت تأمین شده مستقل از شدت ترافیک هریک از گروه‌ها است که وارد مسیریاب می‌شوند. بنابراین حتی اگر دیگر گروه‌های ترافیکی سبب سرریزی مسیریاب و پیوندهای آن گردند هنوز هم منابع کافی بایستی در دسترس آن گروه ترافیک ویژه قرار گیرد که امکان دریافت حداقل نرخ را ضمانت نماید. الگوی EF پیوندی را تعریف می‌کند که ضمانت حداقل پهنای باند لازم را برعهده می‌گیرد.

ب) گسیل ضمانت شده (AF). امکانات PHB از نوع AF به مراتب پیچیده‌تر می‌باشد. در روش AF ترافیک جاری به چهار گروه تقسیم می شود و هر گروه AF با یک حداقل مقدار فضای ذخیره‌سازی و پهنای باند تأمین می‌شود. در هر گروه ترافیکی بسته‌ها به یکی از سه گروه حذف ترجیحی تقسیم می‌شوند. هنگامی که تراکم ترافیک در یک گروه AF روی می‌دهد یک مسیریاب می‌تواند بسته‌ها را برمبنای ارزش ترجیحی آن‌ها حذف نماید. از طریق تغییر مقدار منابع اختصاص یافته به هر کلاس ترافیکی یک ISP می‌تواند سطوح مختلفی از کارایی را به گروه‌های گوناگون ترافیکی اختصاص دهد.

۴- اهمیت Diffserv

در 20 سال گذشته تلاش‌های بی‌شماری به منظور معرفی QoS یا همان کیفیت خدمات در شبکه‌های راهیابی بسته صورت پذیرفته است. بیش‌تر این تلاش‌ها بنابر دلایل اقتصادی و بازماندگی ناموفق بوده‌اند تا آنکه علل فنی باعث آن شده باشد. این تلاش‌ها عبارتند از شبکه‌ی انتها به انتهای ATM و TCPP/IP. تعدادی از این موضوعات در چارچوب Diffserv به شرح ذیل تحلیل می‌شوند. تاکنون فرض بر آن بود که از امکانات Diffserv فقط در درون یک محدوده یا دامنه‌ی مدیریتی استفاده می‌شود در حالیکه موارد کلی‌تر این قضیه عبارت است از یک خدمات انتها به انتها که بایستی از بین چندین ISP حاضر در بین سیستم‌های انتهایی ارتباطی سرچشمه گیرد. به منظور تأمین خدمات انتها به انتهای Diffserv همه‌ی ISPهای بین سیستم‌های انتهایی نه تنها بایستی این خدمات را تأمین نمایند بلکه اغلب آن‌ها در همراهی با یکدیگر بسترسازی این امکانات را نیز انجام می‌دهند. بدون این همراهی، یک ISP تأمین کننده‌ی خدمات Diffserv به مشتریان به‌گونه‌ی تکراری با این موضوع روبه‌رو می‌شود که "بله ما مبلغ اضافه‌ای را می‌پردازیم اما یک توافق خدماتی با یکی از ISPهای سطوح بالاتر نداریم. ما متأسفیم که پرش‌های بسیاری در مکالمه‌ی VOIP شما وجود دارد!"

حتی در یک محدوده‌ی مدیریتی Diffserv، به تنهایی قادر به تأمین ضمانت کیفیت خدمات برای یک گروه خدمات ویژه نمی‌باشد. امکانات Diffserv اجازه می‌دهد گروه‌های گوناگون ترافیک از سطوح مختلف کارایی بهره‌مند شوند. اگر یک شبکه طراحی بهینه نداشته باشد حتی کلاس‌های ترافیکی با بالاترین اولویت از کارایی مناسبی بهره‌مند نخواهد شد. برای آنکه Diffserv تأثیر مطلوب خود بر کارایی شبکه را داشته باشد امکانات آن بایستی در کنار طراحی بهینه‌ی یک شبکه مورد استفاده قرار گیرد. از طرف دیگر Diffserv می‌تواند ظرفیت خدمات‌رسانی یک ISP را از طریق در دسترس قرار دادن منابع به گروه‌های ترافیک با اولویت بالا افزایش دهد. از این طریق یک ISP قادر می‌شود سطح کارایی بالاتر و مطلوب را در اختیار این گونه گروه‌های ترافیکی قرار دهد. بدیهی است هنگامی که این منابع مورد نیاز گروه‌های با اولویت بالا نمی‌باشند می‌توانند در اختیار سایر گروه‌های ترافیکی قرار گیرند. اهمیت دیگر این خدمات پیشرفته کنترل و سازماندهی ترافیک می‌باشد که در اغلب موارد پیچیده و پرهزینه است. هزینه گذاری خدمات مختلف براساس حجم به جای مبالغ ثابت اشتراک ماهیانه که اغلب ISPها از آن در حال حاضر بهره می‌گیرند می‌تواند برای یک ISP هزینه‌ی اضافی را به

دنبال داشته باشد. سرانجام اگر Diffserv در یک شبکه به‌کار گرفته شده باشد و شبکه نیز از ترافیک نسبتاً قابل قبولی بهره‌مند باشد در اغلب موارد تفاوت محسوسی را نمی‌توان بین آن‌چه Diffserv ارایه می‌کند در مقایسه با روش خدمات بهترین تلاش ملاحظه نمود. در حال حاضر تأخیر انتها به انتها تحت تأثیر سرعت‌های دستیابی و مسیریاب‌های ترانزیت است تا آن‌که تأخیرهای صف‌بندی در روترها مانع از کارایی بهینه باشد. حالتی را تصور نمائید که یک کاربر Diffserv با پرداخت هزینه از همان امکانات کارایی بهره‌مند باشد که کاربران خدمات بهترین تلاش از آن برخوردار می‌باشند.

۷-۱۶ تأمین ضمانت‌های QoS

در بخش قبل ملاحظه گردید علامت گذاری بسته و کنترل آن، انزواسازی ترافیکی و زمان‌بندی سطح پیوند عوامل تأمین یک گروه خدمات به شمار می‌آیند که نتیجه‌ی آن‌ها کارایی بهینه در مقایسه با سایر گروه‌های خدماتی است. بدیهی است برخی از نظام‌های زمان‌بندی ویژه مانند زمان‌بندی الویت، گروه‌های خدماتی با مرتبه‌ی پائین‌تر در برابر گروه‌های خدماتی با مراتب بالاتر از ترافیک غیرمحسوس و نهان می‌باشند. از طریق ابعادگذاری مناسب شبکه، گروه‌های خدماتی با اهمیت از کم‌ترین مقدار فقدان بسته و تأخیر برخوردار خواهند شد. اما آیا شبکه قادر به ضمانت تداوم جریان جاری در یک گروه ترافیکی با مرتبه‌ی بالا در طول مدتی که جریان ادامه دارد فقط با بهره‌گیری از مکانیزم‌های توصیف شده در بالا می‌باشد؟ قطعاً خیر، در ادامه‌ی مطالب این فصل ملاحظه خواهد شد روش‌ها و پروتکل‌های اضافه‌ی دیگری به منظور تأمین ضمانت کیفیت خدمات شبکه (QOS) نیاز می‌باشد. این روش‌ها و پروتکل‌ها را می‌توان به اختصار در قالب موارد ذیل خلاصه نمود:

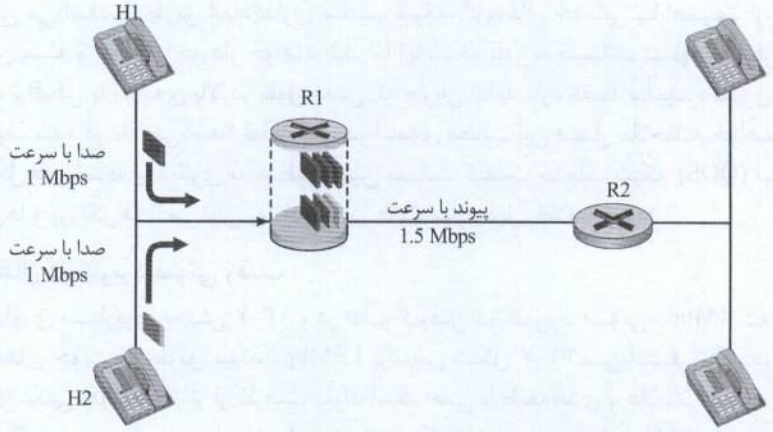
۱- مثال دو کاربرد صوتی رقیب

با یادآوری سناریوی بخش ۷-۱۳ و در نظر گرفتن دو کاربرد صوتی 1Mbps که در حال انتقال بسته‌های خویش از طریق پیوند 1.5Mbps براساس شکل ۷-۳۱ می‌باشند. ترکیب سرعت‌های دو جریان یعنی 2Mbps فراتر از ظرفیت پیوند است. حتی با طبقه‌بندی و علامت گذاری، انزواسازی جریان و بهره‌گیری مشترک از پهنای باند استفاده نشده که البته وجود ندارد، آشکار است که این نظریه مطلوب و مؤثر نخواهد بود. علت آن نیز عدم کفایت پهنای باند لازم جهت تأمین نیازهای هر دو کاربرد به صورت همزمان می‌باشد. اگر هر دو کاربرد به تساوی از پهنای باند مشترک بهره‌گیری نمایند، هر یک از آن‌ها فقط قادر به دریافت 0.75Mbps از آن خواهد بود. به بیان دیگر هر کاربرد 25 درصد از بسته‌های ارسالی خود را از دست می‌دهد. بدیهی است چنین مقدار اندک QoS برای هر دو کاربرد صوتی غیرقابل قبول و بی‌ثمر می‌باشد.

با این تصور که دو کاربرد شکل ۷-۳۱ قادر به تأمین امکانات همزمان نمی‌باشند بنابراین واکنش شبکه چگونه بایستی تعریف شود؟ اگر هر دو کاربرد از QoS بی‌ثمر استفاده نمایند منابع شبکه از طریق یک کاربرد بیهوده مورد بهره‌برداری قرار می‌گیرد که هیچ گونه بهره‌ای را برای کاربر انتهایی به‌همراه ندارد. شاید پاسخ آشکار این چالش آن باشد که یکی از کاربردها بایستی مسدود گردد و دسترسی آن به شبکه محدود شود در حالیکه کاربرد دیگر امکان بهره‌گیری از تمامی ظرفیت 1Mbps موردنیاز کاربرد را در اختیار داشته باشد. شبکه‌ی تلفنی نمونه‌ای از یک شبکه است که چنین محدودیت‌هایی را بر مکالمه‌های تلفنی اعمال می‌کند. اگر منابع لازم جهت ایجاد یک مدار فیزیکی انتها به انتها در شبکه میّسر نباشد آن درخواست مکالمه مسدود می‌شود و از دسترسی آن به شبکه تلفنی از

طریق ارسال یک سیگنال اشغال جلوگیری به عمل می‌آید. در مثال عنوان شده در موضوع این زیربخش، وارد نمودن یک جریان در شبکه در صورتی‌که از یک QoS کافی بهره‌مند نباشد، ثمره‌ای را در پی نخواهد داشت. بدیهی است در چنین شرایطی هزینه‌ای جهت ورود جریانی که QoS موردنیاز خود را دریافت نمی‌دارد به شبکه تحمیل می‌شود زیرا منابع شبکه در اختیار کاربردی قرار گرفته است که هیچ‌گونه بهره‌وری را برای کاربر به‌همراه ندارد.

از سوی دیگر از طریق ورود انحصاری یا مسدود نمودن جریان برمبنای نیازمندی‌های منابع و نیز تأمین نیازهای جریان‌های در حال انجام، شبکه قادر به تأمین ضمانت کافی برای کاربرد جهت دستیابی به QoS مورد تقاضا خواهد شد. تأمین QoS انحصاری موردنیاز یک جریان از طریق آگهی نمودن نیازهای QoS از سوی آن جریان امکان‌پذیر خواهد شد. فرآیند آگهی یک جریان برای QoS مورد نیاز و سپس امکان پذیرش جریان با QoS تقاضا شده و یا مسدود نمودن آن در واژگان شبکه‌ی اینترنت به فرآیند **پذیرش ارتباط**[1] معروف می‌باشد. این مورد اخیر چهارمین روش به همراه سه روش اشاره شده در بخش ۷-۱۳ جهت تأمین QoS موردنیاز می‌باشد.



شکل ۷-۳۱ ♦ سرریزی پیوند **R1** به **R2** از طریق دو کاربرد صوتی رقیب

نتیجه‌گیری چهار: اگر منابع کافی همیشه در دسترس نباشد و QoS بایستی الزاماً ضمانت شود یک فرآیند پذیرش ارتباط لازم است، تا جریان‌ها نیازهای QoS خود را از طریق آن آگهی نموده و سپس و یا در شبکه با QoS مورد تقاضا پذیرش شده و یا آنکه از دسترسی به شبکه در صورت عدم تأمین QoS مورد نظر محروم شوند.

۲- ذخیره و اختصاص منابع، پذیرش ارتباط، برقراری ارتباط

در عنوان قبلی نیاز به چندین پروتکل و روش جدید شبکه به منظور تأمین QoS مورد نظر یک کاربرد هنگامی که آغاز به فعالیت می‌کند معرفی شدند. این موارد عبارتند از:

- **ذخیره و اختصاص منابع.** تنها حالتی که از طریق آن می‌توان تأمین منابع مانند بافرهای موردنیاز و یا پهنای باند پیوند را برای یک ارتباط ضمانت نمود تا دسترسی به QoS مورد نظر و قابل قبول امکان‌پذیر شود، اختصاص انحصاری این منابع به یک ارتباط می‌باشد.

1- Call admission

فرآیندی که در اصطلاح شبکه **ذخیره‌ی منابع**[1] نام گرفته است. هنگامی که منابع اختصاص پیدا می‌کند ارتباط دارای دستیابی بر مبنای تقاضا به این منابع در طول زمان حیات خود بدون توجه به تقاضاهای دیگر ارتباط‌های در جریان دارد. اگر یک ارتباط نسبت به ذخیره و سپس دریافت یک پهنای باند ضمانت شده‌ی xMbps اقدام نماید و هرگز با سرعت فراتر از x عملیات انتقال را انجام ندهد آن ارتباط از کارایی بدون تأخیر و فقدان بسته بهره‌مند خواهد شد.

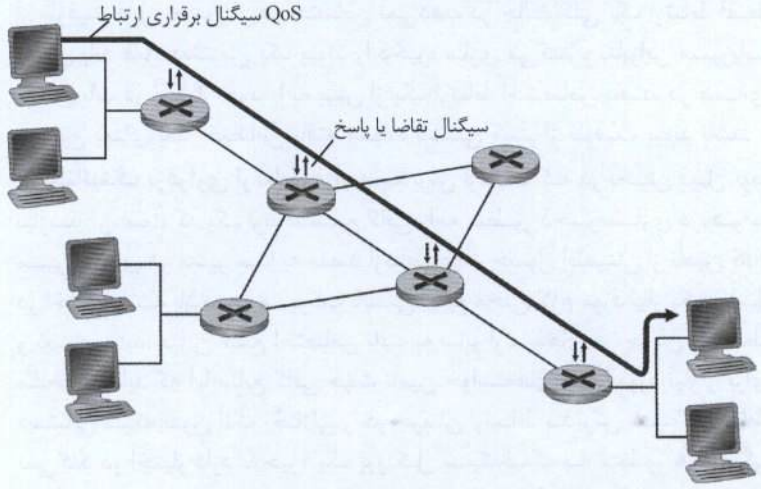
- **پذیرش ارتباط.** اگر ذخیره نمودن منابع الزامی باشد آن‌گاه شبکه بایستی روشی را برای تقاضا و ذخیره‌سازی منابع از سوی ارتباط‌ها در اختیار داشته باشد. این فرآیند معروف به پذیرش ارتباط است. از آن‌جا که منابع نامحدود نمی‌باشند یک ارتباط که تقاضای پذیرش را می‌نماید در صورتی عدم پذیرش و یا مسدود می‌شود که منابع تقاضا شده در اختیار نباشند. چنین مکانیزم پذیرش ارتباط از سوی شبکه‌ی تلفنی انجام می‌شود. هنگامی که اقدام به شماره‌گیری یک خط تلفن می‌نمائیم متقاضی اختصاص منابع شبکه می‌باشیم، اگر مدارهای موردنیاز ارتباط که همان شیارهای TDMA می‌باشند در دسترس قرار گیرند آن‌گاه این مدارها اختصاص می‌یابند و ارتباط کامل می‌شود. اگر مدارها در دسترس نباشند نتیجه‌ی آن مسدود شدن ارتباط و دریافت یک سیگنال اشغال از سوی شبکه می‌باشد. یک ارتباط مسدود شده مجدداً تلاش می‌کند تا در شبکه پذیرش شود اما اجازه‌ی ارسال ترافیک را تا زمانی‌که فرآیند پذیرش ارتباط کامل گردد، نخواهد داشت. بدیهی است این فرآیند مانند عملکرد مدیر یک رستوران است که از ذخیره‌ی مکان برای تعداد میزهای فراتر از ظرفیت رستوران خود خودداری می‌کند. یک روتر نیز اختصاص پهنای باند پیوندهای خود را بیش‌تر از ظرفیت قابل تحمل آن‌ها اختصاص نمی‌دهد. در حالت کلی یک ارتباط فقط درصدی از پهنای باند قابل دسترس یک پیوند را ذخیره سازی می‌کند و بنابراین مسیریاب ممکن است پهنای باند در اختیار خود را به بیش از یک ارتباط اختصاص دهد. در همه‌ی این موارد مجموع پهنای باند اختصاص یافته همواره بایستی کمتر از ظرفیت پیوند باشد.
- **سیگنالینگ برقراری ارتباط.** فرآیند پذیرش ارتباط که در بخش قبل توصیف گردید نیازمند آن است که یک ارتباط منابع کافی را به منظور ذخیره‌سازی در هر یک و تمامی مسیریاب‌های در مسیر مبدأ به مقصد ارتباط جهت حصول اطمینان از تأمین QoS مورد نیاز در اختیار داشته باشد. هر مسیریاب بایستی منابع محلی لازم مورد نیاز یک تعامل را محاسبه و تعیین نماید. میزان منابع اختصاص یافته به سایر ارتباط‌های در جریان را مدنظر قرار دهد و مشخص نماید که آیا منابع کافی جهت تأمین خواسته‌های QoS مورد نیاز را برای هر نقطه‌ی دستیابی شبکه بدون آنکه اختلالی را در جریان ارتباط پذیرش شده از لحاظ QoS ایجاد نمی‌کند در اختیار دارد یا خیر؟ یک پروتکل سیگنالینگ به منظور هماهنگی این موارد گوناگون یعنی اختصاص منابع به هر مسیریاب در مسیر ارتباط و کنترل انتها به انتهای مسیر ارتباط از نقطه نظر اینکه آیا ارتباط از اختصاص منابع کافی برخوردار می‌باشد یا نه، مورد نیاز است. این مهم از طریق **پروتکل برقراری ارتباط**[2] میسر می‌شود.

1- Resource reservation
2- Call setup protocol

شکل ۷-۳۲ فرآیند برقراری ارتباط را نشان می‌دهد. جزئیات مربوط به گام‌های مورد نیاز به منظور پذیرش یک ارتباط عبارتند از:

۱- **تعریف مشخصات ترافیک و QoS مطلوب آن**. برای آنکه یک مسیریاب قادر به تشخیص این مهم باشد که آیا منابع لازم جهت تأمین QoS موردنیاز یک ارتباط را در اختیار دارد آن ارتباط ابتدا بایستی QoS مورد نیاز خود را آگهی نماید و مشخصات ترافیکی مورد نیاز به منظور ارسال به درون شبکه را نیز تعریف نماید. در معماری Intserv اینترنت آن‌چه اصطلاحاً Rspec نامیده شده است و در مستند [RFC2215] تعریف شده است به توصیف QoS مورد درخواست یک ارتباط می‌پردازد. بدیهی است R در این عبارت به مفهوم ذخیره سازی (رزرو) می‌باشد. عبارت Tspec در همین ارتباط که در مستند [RFC2210] تعریف شده است ویژگی‌های ترافیکی ارسال کننده‌ی اطلاعات به درون شبکه و یا نقطه‌ی مقابل آن یعنی گیرنده‌ی اطلاعات را توصیف می‌کند در این عبارت T به مفهوم ترافیک می‌باشد. بدیهی است الگوی ویژه‌ی Rspec و Tspec متغیر و وابستگی به خدمات تقاضا شده دارد. در شبکه‌ای مانند ATM توصیف ترافیک کاربر و اطلاعات عناصر QoS حامل اطلاعات برای یک هدف مشابه برای هر دو مورد Tspec و Rspec می‌باشند.

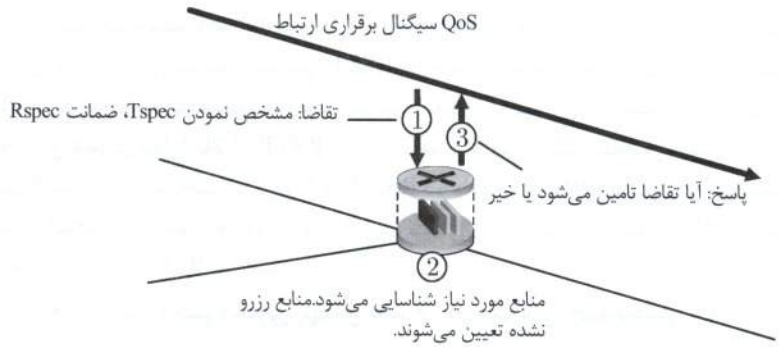
۲- **سیگنالینگ مورد نیاز به منظور برقراری ارتباط**. یک توصیف‌گر ترافیک ارتباط و تقاضای QoS مورد نیاز بایستی به سوی مسیریاب‌هایی که منابع آن‌ها برای یک ارتباط ذخیره‌سازی شده‌اند، حمل گردد. در شبکه‌ی اینترنت پروتکل Rsvp توصیف شده در مستند [REC2210] به منظور تأمین این هدف در درون معماری Intserv به‌کار گرفته می‌شود. در شبکه‌های ATM پروتکل Q2931b حامل این اطلاعات در بین راهیاب‌ها و نقطه‌ی انتهایی شبکه‌ی ATM می‌باشند.



شکل ۷-۳۲ ♦فرآیند برقراری ارتباط

۳- **عنصر پذیرش ارتباط**. هنگامی که یک مسیریاب ویژگی ترافیکی و QoS آن را دریافت می‌دارد بایستی مشخص نماید آیا ارتباط مورد پذیرش قرار می‌گیرد یا خیر. این تصمیم‌گیری پذیرش ارتباط وابسته به ویژگی ترافیک، نوع خدمات تقاضا شده و تأمین منابع موجود جاری اختصاص یافته از سوی روتر به ارتباط‌های در جریان دارد. یادآور می شود در بخش ۷-۱۵

ملاحظه شد چگونه ترکیب منبع کنترل کننده دلو نشت کننده و WFQ می‌تواند حداکثر تأخیر صف را برای یک منبع تعریف نماید. شکل ۷-۳۳ عنصر پذیرش ارتباط را نشان می‌دهد.



شکل ۷-۳۳ ♦ عنصر پذیرش ارتباط

۳- ضمانت QoS تأمین شده در اینترنت: دو پروتکل RSVP , Intserv

معماری خدمات یکپارچه[1] (Intserv) چارچوب تکامل یافته در IETF است که منظور آن ضمانت‌های منفرد QoS به هریک از کاربردها در شبکه‌ی اینترنت می‌باشد. مشخصات خدمات Intserv در مستند [RFC2212] تعریف شده است و رابطه‌های قوی و از نقطه‌نظر ریاضی قابل اثباتی را برای تأخیرهای صف تعریف می‌کند که یک بسته در یک روتر تجربه می‌کند. اگرچه جزئیات پشت پرده‌ی خدمات ضمانت شده بسیار پیچیده می‌باشد اما نکته‌ی بنیادی آن بسیار ساده می‌باشد. در نخستین تقریب یک ویژگی ترافیکی از منبع، از طریق روش نشت کننده‌ی بخش ۷-۱۴ تأمین می‌شود که عوامل (r,b) و خدمات درخواست شده از سوی سرعت انتقال R که بسته‌ها گسیل می‌شوند را تعریف می‌کند. یک ارتباط متقاضی خدمات ضمانت شده نیازمند آن است که بیت‌های موجود در بسته‌اش با سرعت ضمانت شده‌ی Rbps ارسال شوند. با فرض آنکه ترافیک تعریف شده از الگوریتم سطل نشت کننده بهره می‌گیرد و سرعت ضمانت شده‌ی Rbps را نیز اختیار می‌کند این امکان وجود دارد که حداکثر تأخیر صف در مسیریاب را محدود به مقدار تعریف شده‌ای کرد. با یادآوری این نکته که ویژگی ترافیک سطل نشت کننده مقدار ترافیک تولید شده در هر فاصله‌ی زمانی t ثانیه محدود به rt+b می‌باشد. هم چنین با یادآوری از بخش ۷-۱۴ که در آن هنگام که یک منبع سطل نشت کننده درون یک صف را تغذیه می‌کند و ترافیک به صف درآمده با حداقل سرعت Rbps خدمات رسانی می‌شود آن‌گاه حداکثر تأخیر صف تجربه شده از سوی هر بسته محدود به $\frac{b}{R}$ تا هنگامی می‌شود که $R > r$ است. نوع دوم ضمانت خدمات Intserv نیز تعریف شده است که اصطلاحاً به آن خدمات کنترل بار گفته می‌شود و مشخص کننده‌ای این نکته است که یک ارتباط به دریافت QoS مورد نیازی می‌پردازد که همان جریان از یک عنصر غیر متراکم شبکه دریافت می‌دارد.

پروتکل **اختصاص منابع**[2] **(RSVP)** تعریف شده در مستند [RFC 2205] یک پروتکل سیگنالینگ اینترنت است که از آن می‌توان به‌منظور عملیات سیگنالینگ مورد نیاز برقراری ارتباط از سوی Intserv

1- Integrated service architecture
2- Resource Re SerVation Protocol

بهره گرفت. پروتکل RSVP همچنین در ارتباط نزدیک با Diffserv به منظور همانگ سازی عملیات Diffserv در بین چندین شبکه و نیز به منظور سیگنالینگ MPLS بنابر مستند [RFC3209] با نام RSVP-TE که در فصل ۵ از آن سخن به میان آمد، مورد بهره‌برداری قرار می‌گیرد.

در یک چارچوب Intserv، پروتکل RSVP به کاربردها اجازه می‌دهد پهنای باند مورد نیاز جریان داده‌ی خود را رزرو نمایند. پروتکل RSVP هم‌چنین از سوی یک میزبان به نمایندگی از سوی یک کاربرد جریان داده به منظور تقاضای ظرفیت معینی از پهنای باند شبکه مورد استفاده قرار می‌گیرد. علاوه بر همه‌ی موارد بالا از RSVP مسیریاب‌ها به منظور ارسال تقاضای پهنای باند مورد نیاز خود بهره می‌گیرند. در پیاده‌سازی پروتکل RSVP نرم‌افزار RSVP بایستی در گیرنده‌ها، فرستنده‌ها و مسیریاب‌های در امتداد مسیر انتها به انتها نشان داده شده در شکل ۷-۳۲ حاضر باشند. دو اصل بنیادی پروتکل RSVP عبارتند از:

- تأمین **ذخیره‌سازی پهنای باند در درخت‌های چندپخشی** که در آن فرآیند تک پخش سرچشمه گرفته از حالت چندپخش می‌باشد. این موضوع به ویژه برای کاربردهای چندرسانه‌ای مانند جریان پخش زنده‌ی تصاویر TV از طریق IP مفید می‌باشد که در آن بسیاری از گیرنده‌ها نیازمند دریافت ترافیک چندرسانه‌ای مشابه ارسالی از یک منبع می‌باشند.
- این پروتکل **دریافت‌گرا** می‌باشد به عبارت دیگر گیرنده‌ی یک توده داده ذخیره‌سازی منابع مورد استفاده‌ی یک جریان را تعریف و نگهداری می‌کند. این ویژگی خلاق مرکزگرایی گیرنده در RSVP سبب می‌شود تا گیرنده‌ها تسلط کامل بر کنترل ترافیک دریافت شده داشته باشند. به عنوان مثال گیرنده‌های گوناگون توانایی دریافت و مشاهده‌ی فرآیند چندپخش چندرسانه‌ای با مراتب شفاف‌سازی متفاوت را داشته باشند. این عملکرد با آن‌چه پروتکل Q2931b در اختیار گیرنده‌ها قرار می‌دهد تفاوت چشمگیر دارد.

استاندارد RSVP تعریف شده در مستند [RFC2205] مشخص نمی‌کند چگونه شبکه پهنای باند ذخیره‌سازی شده‌ی جریان‌های داده را تأمین می‌کند، بلکه RSVP تنها به عنوان یک پروتکل اجازه‌ی ذخیره‌ی پهنای باند لازم پیوندها را به کاربردها می‌دهد. هنگامی که عملیات مربوط به ذخیره‌سازی کامل می‌شود برعهده‌ی مسیریاب‌های اینترنت است که پهنای باند رزرو شده را برای جریان‌های داده تأمین نمایند. فرآیند اختصاری و کنترل پهنای باند از این طریق با بهره‌گیری از روش‌هایی مانند زمان‌بندی الویت، صف‌بندی عادلانه‌ی توزینی و یا الگوریتم دلو نشت کننده انجام می‌شود.

۷-۱۷ اصول وضعیت نرم

از پروتکل RSVP به منظور نصب یک حالت مانند ذخیره سازی پهنای باند در مسیریاب‌ها استفاده می‌شود که معمولاً معروف به **یک پروتکل وضعیت نرم**[1] می‌باشد. معمولاً عبارت وضعیت نرم با رویکردهای سیگنالینگ که در آن، زمان خروج وضعیت نصب شده در صورتیکه وضعیت متناوباً نوسازی نگردد، وابسته می‌شود. بدیهی است نوسازی زمان خروج از طریق یک پیام سیگنالینگ صادر شده از سوی یک موجودیت که نخست وضعیت را نصب نموده است، انجام می‌گیرد و از این طریق تقاضای ادامه‌ی نصب حالت نرم را می‌کند. از آن‌جائیکه حالت نوسازی نشده سرانجام به پایان می‌رسد

1- Soft state protocol

سیگنالینگ وضعیت نرم نیازمند هیچ یک از موارد حذف حالت انحصاری و یا یک مرحله به منظور حذف حالت در صورت فروپاشی نصاب وضعیت نمی‌باشد. از طرف دیگر از آن جا که نصب حالت و پیام‌های بازسازی به دنبال پیام‌های نوسازی متناوب یکدیگر انجام می‌شود بهره‌گیری از روش‌های سیگنالینگ مطمئن الزامی نمی‌باشد. عبارت حالت نرم نخستین مرتبه از سوی کلارک (clark) مطرح شد که از آن برای توصیف فرآیند پیام‌های بازسازی متناوب حالت‌ها که از سوی میزبان‌ها بایستی ارسال می‌گردید مورد بهره‌برداری قرار می‌گرفت. پیشنهاد نهفته در این کاربرد این موضوع است که با چنین پیام‌های بازسازی یک وضعیت ممکن است در اثر یک فروپاشی تلف شود و سپس به‌گونه‌ی خودکار از سوی پیام بازسازی مجدداً بازیابی گردد با این نتیجه که همه‌ی موارد از سوی میزبان قابل رؤیت بوده و نیازی به تحریک و فراخوانی مراحل بازیابی نمی‌باشد.

در حالت کلی رویکرد حالت نرم بهره‌گیری از بهترین تلاش نصب و بازیابی متناوب حالت از طریق یک نصاب وضعیت و حذف آن در نگهدارنده‌ی حالت در زمان خروج آن می‌باشد. پروتکل‌های متعددی از این رویکرد بهره‌گیری نموده‌اند از جمله پروتکل‌های RSVP، PIM، SIP و IGMP که در فصل ۴ تشریح شده‌اند و نیز جداول گسیل جلو مربوط به فصل ۵ که در زیربخش ۵-۱۰ تشریح گردیدند.

رویکرد دیگر سیگنالینگ **وضعیت سخت** است که عملیاتی برعکس حالت نرم را شامل می‌شود و هنگامی که وضعیتی در این رویکرد نصب گردد تنها در صورتی حذف می‌شود که پیام قطع حالت را از نصاب وضعیت دریافت نماید. از آن‌جا که حالت نصب شده به قوت خود باقی می‌ماند تا آن‌که انحصاراً حذف شود سیگنالینگ وضعیت سخت نیازمند روشی به منظور حذف در شرایطی است که نصاب حالت، دچار فروپاشی و یا ترک عملیات بدون حذف وضعیت شده است. از طرف دیگر از آن‌جا که نصب و یا حذف حالت فقط یک مرتبه انجام می‌شود و بازیابی یا زمان خروج برای حالت وجود ندارد بنابراین برای نصاب حالت بسیار مهم است که از زمان نصب و یا حذف حالت آگاه باشد. بنابراین پروتکل‌های سیگنالینگ قابل اطمینان در مقابل بهترین تلاش وابسته به پروتکل‌های حالت سخت می‌باشند. به بیان دیگر رویکرد حالت سخت نصب و حذف وضعیت اطلاعات به‌گونه‌ی انحصاری و قابل اطمینان است. بهره‌گیری از رویکرد وضعیت سخت در پروتکل‌هایی مانند ST-II [RFC1190] و [ITU-T Q2931] می‌باشد.

از رویکرد سیگنالینگ مطمئن برمبنای ACK به عنوان بخشی از گسترش RSVP در مستند [RFC2961] استفاده شده است و بنابراین RSVP برخی از عناصر وضعیت سخت را نیز در درون خود به‌کار گرفته است.

۷-۱۸ برنامه‌های پیشرفته‌ی کاربردی در شبکه‌های چند رسانه‌ای

در بخش ۷-۱ همین فصل کاربرد شبکه‌ی چند رسانه‌ای به‌عنوان یک برنامه‌ی شبکه معرفی شد که از صدا و تصویر استفاده می‌کند تا محتوایی را انتقال دهد. این بخش جدید اضافه شده به ویراست ششم واژگانی جدید از کاربردهای چند رسانه برپایه‌ی فناوری‌های نوین را معرفی می‌کند. هر گروه از کاربردها در واژگان خود مجموعه‌ای از نیازهای خدماتی انحصاری و موضوعات مربوط به طراحی آنها را شامل می‌شود. قبل از گام نهادن به بحث پیرامون تغییرات نوین در شبکه‌های چند رسانه‌ای، تحلیل ویژگی‌های ذاتی رسانه‌های صوتی و تصویری نیز از اهمیت خاص بهره‌مند است. این صفات ذاتی به‌شرح ذیل تشریح می‌شوند:

۱- ویژگی‌های محتوای ویدیویی

شاید یکی از مهم‌ترین صفات محتوای تصویری یا ویدیویی **سرعت بالای انتقال** مورد نیاز آن است. تصاویر توزیع شده‌ی اینترنتی معمولاً از محدوده‌ی 100kbps برای کنفرانس‌های تصویری کیفیت پائین آغاز می‌شوند و به سرعت بیش از 3Mbps برای پخش تصاویر با شفافیت و وضوح بسیار بالا خاتمه می‌یابند. به‌منظور درک مقایسه‌ی پهنای باند مورد نیاز محتوای تصویری با دیگر کاربردهای اینترنتی سه کاربر انسانی مختلف را به‌صورت ذیل در ذهن خود تصور نمایید که هرکدام از آنها از کاربردهای اینترنتی متفاوت بهره می‌گیرند. نخستین کاربر فرهاد است که به‌سرعت تصاویر قرار گرفته روی صفحه‌ی فیس‌بوک دوستش را مرور می‌کند. فرض کنید فرهاد هر 10 ثانیه یک تصویر جدید را تماشا می‌کند و این عکس‌ها هرکدام دارای میانگین حجم برابر 200kbyte هستند. فرض کنید هر یک کیلو بایت معادل 8000 بیت باشد. کاربر دوم میترا مشغول شنیدن آهنگ‌های مورد علاقه‌ی خود از طریق اینترنت و در قالب رایانش ابری روی تلفن هوشمند خویش است. فرض کنید میترا به تعداد زیادی فایل MP3 به‌صورت پی‌درپی گوش می‌کند و هر یک از آنها با نرخ 128kbps رمزگذاری شده‌اند. سرانجام کاربر سوم بهرام است که مشغول تماشای تصاویر ویدیویی رمزگذاری شده با نرخ 2Mbps است. اگر فرض شود طول زمان ارتباط هر سه کاربر معادل 4000 ثانیه و تقریباً 67 دقیقه باشد آنگاه جدول ۷-۳ مقایسه‌ی سرعت انتقال و اندازه‌ی بایت‌های انتقال یافته برای این سه کاربر را نشان می‌دهد. از ارقام جدول ملاحظه می‌شود پخش تصویر بیشترین پهنای باند را به خود اختصاص می‌دهد زیرا سرعت انتقال آن 10 برابر بیشتر از فیس‌بوک و کاربردهای پخش موسیقی است. بنابراین اگر یک برنامه‌ی کاربردی ویژه‌ی ویدیو طراحی می‌شود، نخستین و مهم‌ترین موضوع در نظر گرفتن نیازهای نرخ انتقال بالا برای این‌گونه محتواها می‌باشد.

جدول ۷-۳ مقایسه‌ی نیازهای سرعت سه کاربرد اینترنتی.

	سرعت انتقال (bps)	تعداد بایت انتقال یافته در 67 دقیقه
صفحه‌ی فیس‌بوک فرهاد	160kbps	80Mbytes
دریافت فایل موسیقایی میترا	128kbps	64Mbytes
تماشای محتوای ویدیویی بهرام	2Mbps	1Gbytes

با توجه به محبوبیت ویدیو و نرخ انتقال بالای آن باعث تعجب نیست که شرکتی مانند Cisco پیش‌بینی نماید: پخش و ذخیره‌ی تصاویر ویدیویی تا پایان سال 2015 میلادی تقریباً 90 درصد ترافیک مصرفی اینترنت را در سراسر جهان شامل خواهد شد.

یک ویژگی مهم دیگر محتوای تصویری امکان فشرده‌سازی آن است که سبب توازن بین کیفیت ویدیویی و سرعت انتقال می‌شود. یک ویدیو معمولاً زنجیره‌ای از تصاویر است که با سرعت ثابتی به‌نمایش در می‌آید. به‌عنوان مثال با سرعت 24 یا 30 تصویر بر ثانیه. یک تصویر رمزگذاری شده‌ی دیجیتالی بدون فشرده‌سازی شامل آرایه‌ای از نقاط (پیکسل‌های) رنگی است که در آن هر نقطه‌ی رنگی در قالب چند بیت تعریف می‌شود تا روشنایی و رنگ خاصی را تعریف نماید. دو نوع افزونگی در ویدیو وجود دارد و هر دو مورد استفاده‌ی **فشرده‌سازی تصویر** قرار می‌گیرند. این دو عبارتند از:

افزونگی فضایی که عبارت است از فضای سفید گسترده که طبیعت ساختاری یک تصویر را تشکیل می‌دهد و بدون تأثیر چشمگیر بر کیفیت تصویر می‌تواند فشرده‌سازی شود و دیگری **افزونگی زمانی** است که توصیف تکرار ویژگی‌هایی از یک تصویر به تصویر نتیجه شده است. به‌عنوان مثال اگر یک تصویر و تصویر نتیجه شده از آن عیناً همانند باشند آنگاه دلیلی وجود ندارد که تصویر نتیجه شده مجدداً رمزگذاری شود بلکه بهینه است هنگام رمزگذاری مشخص شود که تصویر نتیجه شده دقیقاً همان تصویر اصلی است. در حال حاضر الگوریتم‌های فشرده‌سازی موجود قادرند عملیات فشرده‌سازی یک تصویر را با هر سرعت دلخواه انجام دهند، بدیهی است هرچه نرخ فشرده‌سازی بیشتر باشد کیفیت تصویر و تجربه تماشای آن ویدیو نیز بهتر خواهد شد.

از فشرده‌سازی می‌توان به‌منظور خلق **نسخه‌های چندگانه** از یک ویدیو مشابه هرکدام با سطح کیفیت متفاوت نیز بهره گرفت. به‌عنوان مثال از فشرده‌سازی می‌توان بهره گرفت تا سه نسخه از یک ویدیو را با سرعت‌های 300kbps , 1Mbps و 3Mbps رمزگذاری نمود. در این فرآیند برای کاربران این امکان فراهم می‌آید تا بنابر پهنای باند در اختیار خود، تصمیم به گزینش نسخه‌ی مطلوب جهت تماشا بگیرند. کاربران بهره‌مند از ارتباط‌های اینترنتی سرعت بالا ممکن است نسخه‌ی 3Mbps را گزینش نمایند. بدیهی است تصاویر یک کاربرد ویدیو کنفرانس هنگام جریان و پخش آن قابل فشرده‌سازی می‌شود تا بهترین کیفیت ویدیویی را مشروط بر پهنای باند مطلوب در انتها به انتهای ارتباط بین کاربران تأمین نماید.

۲- مکالمات صوتی و تصویری از طریق اینترنت

مکالمه‌های صوتی بی‌درنگ از طریق اینترنت اصطلاحاً **تلفن اینترنتی** نامیده شده است زیرا از نگاه کاربر بسیار مشابه خدمات تلفنی شبکه‌های راهیابی شده‌ی مداری سنتی است. نام دیگر این‌گونه مکالمه‌ها **انتقال صدا از طریق پروتکل اینترنت (VOIP)** می‌باشد. مکالمه‌های ویدیویی نیز همین حالت را دارند با این تفاوت که تصاویر افراد مشغول مکالمه به صدای آنها نیز افزوده شده است. اغلب سیستم‌های مکالماتی صوتی و تصویری در حال حاضر برای کاربران امکان ایجاد کنفرانس‌های سه یا چندین نفره‌ی صوتی و تصویری را نیز فراهم می‌آورند. مکالمه‌های صوتی ـ تصویری امروز به شکل گسترده‌ای با بهره‌گیری از اینترنت در جریان است. شرکت‌هایی مانند اسکایپ، QQ و GoogleTalk میزبانی صدها میلیون کاربر روزانه را برعهده دارند.

در بحث پیرامون نیازهای خدمات کاربردی در فصل ۲ و به ویژه در شکل ۲-۴ محورهایی معرفی شدند که در کنار آنها نیازهای کاربردها را می‌توان طبقه‌بندی کرد. دو محور مهم این موارد یکی موضوعات زمانبندی و دیگری میزان پذیرش اتلاف داده می‌باشد که از نقطه نظر کاربردهای صوتی و تصویری بسیار با اهمیت است. علت این اهمیت به واسطه‌ی حساسیت مکالمه‌های صوتی و تصویری در برابر **تاخیر** است. برای یک مکالمه‌ی بین دو یا چند نفر تاخیر حاصل از لحظه‌ی صحبت یا حرکت یک طرف تا لحظه‌ای که این عملیات در سمت دیگر مکالمه تاثیر خود را نشان دهد بایستی کمتر از چندصد میلی ثانیه باشد. برای صدا تاخیرهای کمتر از 150 میلی‌ثانیه از سوی انسان قابل درک نیست. بنابراین تاخیرهای بین 150 میلی‌ثانیه تا 400 میلی‌ثانیه قابل قبول است. تاخیرهای بالاتر از این مقدار ناگوار و غیر قابل تفکیک از سوی مخاطب می‌باشد.

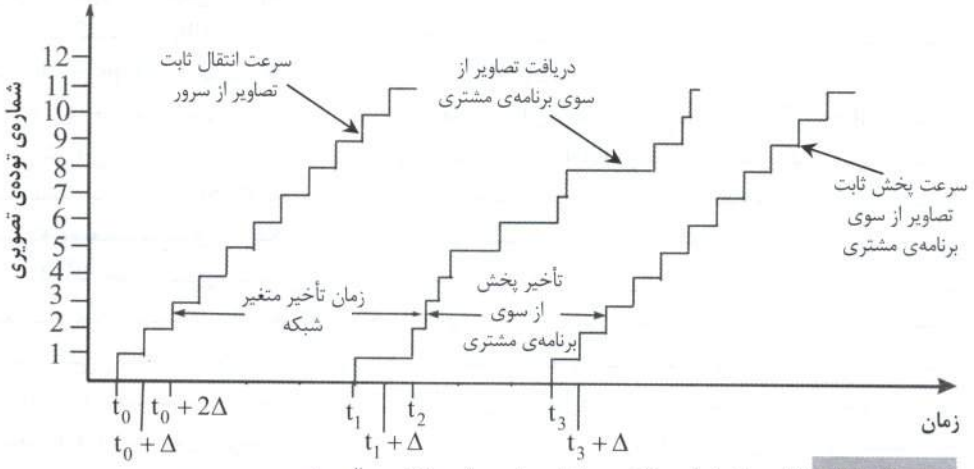
از طرف دیگر کاربردهای چند رسانه‌ای گفتاری کمتر در برابر **اتلاف** حساس هستند. اتلاف‌های

اتفاقی به‌ندرت در بازپخش صدا و تصویر تأثیرگذار هستند و این اتلاف‌ها غالباً به‌صورت نسبی یا کامل پنهان هستند. حساسیت در برابر تاخیر می‌تواند مقاوم در برابر اتلاف نیز باشد و اظهار این عبارت با کاربردهای انعطاف‌پذیر داده مانند وب‌گردی، ایمیل، شبکه‌های اجتماعی و تلنت متفاوت است. برای کاربردهای انعطاف‌پذیر تاخیرهای دراز مدت ناخوشایند هستند اما آسیب‌رسان نیستند. در هر صورت آنچه اهمیت دارد تمامیت و یکپارچگی داده‌ی انتقال یافته است.

۳- پخش سیال تصاویر ذخیره شده:

برای کاربردهای پخش سیال ویدیو، تصاویر ویدیویی از پیش ثبت شده روی سرورها قرار می‌گیرند و کاربران درخواست‌های خود را به‌این سرورها به‌منظور تقاضای تماشای بی‌درنگ آنها ارسال می‌کنند. کاربر می‌تواند یک فایل تصویری را از ابتدا تا انتها بدون وقفه تماشا نماید و یا قبل از پایان یافتن آن را متوقف نماید. ممکن است یک کاربر توقف موقت یا تغییر مکان به آینده وگذشته‌ی صحنه‌های فیلم را نیز انجام دهد. پخش سیال ویدیویی را می‌توان به‌سه گروه طبقه‌بندی نمود: (۱) **پخش سیال UDP**، (۲) **پخش سیال HTTP** و (۳) **پخش سیال HTTP تطبیقی**. اگرچه همگی این سیستم‌ها در عمل مورد استفاده قرار می‌گیرد اما اکثریت سیستم‌های امروزی از دو رویکرد (۲) و (۳) بیشتر استفاده می‌کنند. ویژگی مشترک هر سه گونه پخش سیال ویدیویی اشاره شده در آن است که از ذخیره‌سازی کاربرد سمت مشتری به‌منظور دوری جستن از تأثیر تاخیرهای متغیر انتها به انتها و نیز تغییرات پهنای باند در دسترس بین سرور و مشتری بیشتر استفاده می‌کند. برای پخش سیال ویدیو به هر دو صورت زنده (بی‌درنگ) و ذخیره‌سازی شده معمولاً کاربران می‌توانند تاخیرهای اولیه‌ی چند ثانیه‌ای را بین هنگام صدور تقاضای برنامه‌ی مشتری و زمان پخش ویدیو در سمت مشتری تحمل نمایند. این موضوع باعث می‌شود هنگامی که تصاویر شروع به ورود در سمت برنامه‌ی مشتری می‌کنند، برنامه‌ی مشتری لازم نیست بی‌درنگ پخش آن را آغاز نماید بلکه می‌تواند ذخیره‌ای از تصاویر را در یک کاربرد نگهداری نماید. هنگامی که برنامه‌ی مشتری ذخیره‌ای از محتوای تصاویر را به اندازه‌ی چند ثانیه انجام دهد و در طول ذخیره‌سازی اقدام به پخش آن نکند برنامه‌ی مشتری می‌تواند پخش آن را آغاز نماید. دو مزیت مهم عملکرد **ذخیره‌سازی مشتری** عبارتند از: (۱) ذخیره‌سازی برنامه‌ی مشتری می‌تواند تغییرات در تأخیر برنامه‌ی مشتری ـ سرور را جذب نماید. زیرا اگر قطعه‌ی خاصی از تصویر شامل تاخیر گردد تا زمانی که دوره‌ی ذخیره‌سازی و پخش تصاویر دریافتی به‌پایان نرسیده است چنین تاخیر طولانی قابل لمس نمی‌باشد. (۲) اگر پهنای باند بین دو برنامه‌ی مشتری و سرور موقتاً به کمتر از مقدار مورد نیاز سرعت مصرفی ویدیو افت نماید آنگاه یک کاربر می‌تواند از بازپخش بدون وقفه‌ی دریافتی کماکان تا زمانی که حافظه‌ی ذخیره‌سازی مشتری تهی نشده است، لذت ببرد. شکل ۷-۳۴ ذخیره‌سازی مشتری ـ سرور را نشان می‌دهد. در این مثال ساده فرض کنید رمزگذاری تصاویر با یک نرخ ثابت انجام شود و هر توده‌ی ویدیو شامل قاب‌هایی از تصاویر باشد که در یک فاصله‌ی زمانی ثابت و یکسان Δ بایستی پخش شود. سرور نخستین توده‌ی تصویری را در زمان t_0، توده‌ی دوم را در زمان $t_0 + \Delta$، سومین توده را در $t_0 + 2\Delta$ و در نهایت توده‌ی n را در $t_0 + n\Delta$ انتقال می‌دهد. هنگامی که مشتری آغاز به بازپخش نماید هر توده بایستی به‌اندازه‌ی Δ یکای زمانی بعد از توده پیشین خود بازپخش شود تا زمانبندی اصلی ویدیوی ضبط شده را تولید نماید. به‌دلیل تاخیرهای متغیر انتها به انتها توده‌های گوناگون تصویری تاخیرهای متفاوت را تجربه می‌کنند. نخستین توده‌ی تصویری در زمان t_1 وارد برنامه‌ی مشتری می‌شود و توده‌های بعدی به‌ترتیب در زمان‌های t_2 و غیره وارد خواهند شد. تاخیر

شبکه برای توده‌ی i فاصله‌ی افقی بین زمانی است که توده از سوی سرور انتقال یافته است و زمانی که آن توده از سوی مشتری دریافت شده است. آشکار است که تاخیر شبکه از یک توده‌ی ویدیویی به توده‌ی دیگر متفاوت است. در این مثال اگر قرار باشد برنامه‌ی مشتری بی‌درنگ پس از دریافت توده‌ی نخست در زمان t_1 آغاز به پخش نماید آنگاه توده‌ی دوم در زمان بازپخش $t_1 + \Delta$ وارد نخواهد شد. در این وضعیت بازپخش ویدیو ممکن است منتظر دریافت توده‌ی یک گردد و یا آنکه از توده‌ی یک صرفنظر نماید. در هر دو حالت اختلال در پخش به‌وجود می‌آید و درنتیجه نامطلوب است. اما اگر برنامه‌ی مشتری تاخیر بازپخش را به‌اندازه‌ی زمان t_3 به‌تعویق اندازد که در این فاصله‌ی زمانی توده‌های 1 تا 6 دریافت شده‌اند آنگاه بازپخش متناوب می‌تواند با **همه‌ی** توده‌های دریافت شده قبل از زمان بازپخش آنها ادامه یابد.



شکل ۷-۳۴ ♦ تاخیر بازپخش برنامه‌ی مشتری در روش پخش سیال ویدیو.

۴- پخش سیال UDP

در این بخش بحث مختصری پیرامون پخش سیال UDP ارایه می‌شود. در پخش سیال UDP برنامه‌ی سرور انتقال ویدیو را با سرعتی دنبال می‌کند که امکانات دریافت تصاویر از سوی برنامه مشتری سازگار باشد. به‌همین منظور سرعت خروج تصاویر از طریق UDP با نرخ ثابت انجام می‌گیرد. به‌عنوان مثال اگر قدرت مصرف، نرخ معادل 2Mbps باشد و هر بسته‌ی UDP نیز تعداد 8000 بیت از تصویر ویدیویی را حمل نماید، آنگاه برنامه‌ی سرور نسبت به انتقال یک بسته‌ی UDP به هر سوکت در زمانی معادل: (8000bit/2Mbps)=4ms اقدام می‌کند. در فصل ۳ آموختیم پروتکل UDP از مکانیزم کنترل تراکم بی‌بهره است و بنابراین سرور می‌تواند تزریق بسته‌ها را به شبکه براساس سرعت مصرف تصاویر بدون محدودیت کنترل سرعت از سوی پروتکل TCP انجام دهد. پخش سیال UDP از حافظه‌ی کوچکی در سمت مشتری بهره می‌گیرد که اندازه‌ی ظرفیت آن کمتر از یک ثانیه از تصویر است.

قبل از انتقال بسته‌های ویدیویی به UDP، برنامه‌ی سرور این بسته‌ها را در قالب قطعه‌های لایه‌ی انتقال بسته‌بندی می‌کند که مخصوص حمل صدا و تصویر با استفاده از پروتکل بی‌درنگ (RTP) بنابر مستند [RFC 3550] طراحی شده‌اند و یکی از ویژگی‌های دیگر پخش سیال UDP آن است که علاوه بر جریان تصویری بین سرور و مشتری هر دو برنامه‌ی سرور و مشتری به‌صورت موازی یک ارتباط

کنترل را نگهداری می‌کنند که از طریق آن مشتری فرمان‌های مربوط به وضعیت ارتباط و تغییرات احتمالی آن نظیر توقف، باز پخش مجدد، تغییر موقعیت و غیره را ارسال می‌کند. این ارتباط کنترلی از زوایای زیادی شبیه به ارتباط کنترل FTP در فصل ۲ می‌باشد. پروتکل RTSP از جمله پروتکل‌های باز (آزاد) پرطرفدار برای چنین ارتباط‌هایی می‌باشد.

اگر چه پخش سیال UDP در بسیاری از سیستم‌های آزاد و در اختیار عموم و نیز محصولات خصوصی مورد استفاده قرار گرفته است اما از سوی سه نقطه ضعف اساسی در تهدید است:(۱) به‌علت آنکه مقدار پهنای باند در دسترس بین سرور و مشتری غیر قابل پیش‌بینی و متغیر است بنابراین پخش سیال UDP با سرعت ثابت موفق به امکان پخش مستمر فایل صوتی ـ تصویری UDP نخواهد شد. به‌عنوان مثال حالتی را درنظر بگیرید که نرخ مصرف ویدیو 1Mbps است و پهنای باند در دسترس مشتری ـ سرور نیز بیش از 1Mbps می‌باشد و هر چند دقیقه یک بار پهنای باند در دسترس برای مدت چند ثانیه کمتر از 1Mbps می‌گردد. در چنین حالتی یک سیستم پخش سیال UDP که در حال پخش ویدیو با نرخ 1Mbps از طریق RTP/ UDP است احتمالاً تجربه‌ی تصویری ضعیفی را برای کاربر فراهم می‌کند زیرا قاب‌های تصویری بی‌درنگ پس از کاهش پهنای باند در دسترس به کمتر از 1Mbps حذف و یا ساکن (بی‌حرکت) می‌شوند. (۲) ضعف دوم پخش سیال UDP در نیاز آن به یک سرور کنترل رسانه مانند یک سرور RTSP به‌منظور پردازش در خواست‌های تعاملی بین برنامه‌های مشتری و سرور است که وضعیت مشتری را رهگیری نماید. به سخن دیگر نقطه‌ای از بازپخش مشتری در ویدیو به این معنی که آیا پخش تصویر ویدیویی متوقف شده است و یا آنکه در حال پخش است. عملیات رهگیری وضعیت برنامه‌ی مشتری برای **هریک** از جریان‌های در حال پخش مشتری انجام می‌گیرد. این موضوع هزینه‌ی کلی و نیز پیچیدگی به کارگیری یک سیستم پرحجم ویدیو بر مبنای درخواست را افزایش می‌دهد. (۳) ضعف سوم آن است که بسیاری از فایروال‌ها (دیواره‌های آتش) به‌منظور مسدود کردن ترافیک UDP پیکربندی می‌شوند و نتیجه‌ی آن ممانعت دریافت تصاویر UDP برای کاربرانی است که در عقب این فایروال‌ها قرار می‌گیرند.

۵- پخش سیال HTTP

در پخش سیال HTTP تصاویر در یک سرور HTTP در قالب یک فایل معمولی با یک URL خاص ذخیره‌سازی می‌شود. هنگامی که یک کاربر مایل به تماشای ویدیو می‌باشد، برنامه‌ی مشتری یک ارتباط TCP را با برنامه‌ی سرور برقرار می‌سازد و یک تقاضای HTTP را درقالب GET برای آن URL گسیل می‌کند. پس از آن سرور، فایل ویدیو را درون پیام پاسخ HTTP ارسال می‌دارد و این عمل را با سرعت ممکن آن لحظه برمبنای اجازه‌ی کنترل جریان و کنترل تراکم TCP انجام می‌دهد. در سمت مشتری بایت‌ها در یک بافر متعلق به برنامه‌ی کاربرد مشتری جمع‌آوری می‌شوند. هنگامی که تعداد بایت‌ها در این بافر بیشتر از آستانه‌ی از پیش تعیین شده باشد برنامه‌ی مشتری آغاز به بازپخش تصاویر می‌کند. به‌این صورت خاص که برنامه‌ی مشتری قاب‌های تصویر را از بافر برنامه‌ی کاربردی مشتری اخذ و سپس آنها را از فشرده‌سازی خارج و آنگاه روی صفحه‌ی نمایشگر کاربر به نمایش می‌گذارد.

در فصل ۳ آموختیم هنگام انتقال یک فایل از طریق TCP، نرخ انتقال سرور به مشتری می‌تواند به سبب مکانیزم‌های کنترل تراکم TCP به‌گونه‌ی چشمگیری تغییر نماید. به‌ویژه آنکه تغییر نرخ انتقال در قالب یک نمودار دندانه‌ای مطابق شکل ۳-۵۳ در ارتباط با کنترل تراکم TCP صورت می‌گیرد. علاوه

بر این موارد به‌دلیل روش انتقال مجدد TCP، تأخیر بسته‌ها می‌تواند به‌صورت چشمگیری افزایش یابد. به‌دلیل این ویژگی‌های TCP بینش سنتی دهه‌ی 1990 میلادی آن بود که پخش سیال تصویر از طریق TCP هرگز ممکن نخواهد بود. اما در گذر ایام طراحان سیستم‌های پخش سیال تصویری دریافتند کنترل تراکم TCP و مکانیزم‌های انتقال مطمئن داده اگر برنامه‌ی مشتری از روش دریافت تصاویر قبل از پخش و نیز ذخیره‌سازی آنها بهره‌گیری نماید لزوماً مانع بازپخش مستمر تصاویر نخواهد شد.

بهره‌گیری از HTTP از طریق TCP امکان عبور ویدیو از فایروال و NAT را نیز به‌سادگی امکان‌پذیر می‌سازد. زیرا ین دو مورد پیکربندی شده‌اند تا اغلب ترافیک‌های UDP را مسدود نمایند اما اجازه‌ی عبور ترافیک HTTP را می‌دهند. روش پخش سیال پروتکل HTTP نیاز به استفاده از برنامه‌ی سرور کنترل رسانه مانند RTSP را برطرف می‌سازد که نتیجه‌ی مستقیم آن کاهش هزینه‌ی کاربردهای این‌گونه از طریق اینترنت است. به‌سبب همه‌ی این ویژگی‌ها بسیاری از کاربردهای پخش تصاویر در حال حاضر مانند یوتیوب و Netfix از پخش سیال HTTP روی TCP به‌عنوان پروتکل زیر ساخت پخش سیال ویدیو استفاده می‌کنند.

- **پیش دریافت ویدیو**

بخش پیشین نشان داد بهره‌گیری از ذخیره‌سازی تصاویر در سمت برنامه‌ی مشتری تأثیر تاخیرهای انتها به انتها و نیز تغییر در پهنای باند در دسترس را کاهش می‌دهد. در مثال شکل ۷-۳۴ برنامه‌ی سرور تصاویر ویدیو را با نرخ مورد نیاز بازپخش ارسال می‌کند. اما برای پخش سیال تصاویر **ذخیره‌سازی شده** برنامه‌ی مشتری می‌تواند بارگیری تصاویر را با سرعت **بالاتر** از نرخ مصرفی انجام دهد و بنابراین **پیش دریافت**[1] قاب‌های ویدیویی مورد نیاز برای آینده‌ی خود را تأمین نماید. تصاویر دریافتی در این روش در بافر برنامه‌ی کاربردی مشتری ذخیره‌سازی می‌شود. به‌دلیل مکانیزم پرهیز از تراکم TCP این پروتکل تلاش می‌کند از همه‌ی امکانات پهنای باند موجود بین سرور و مشتری بهره‌گیری نماید.

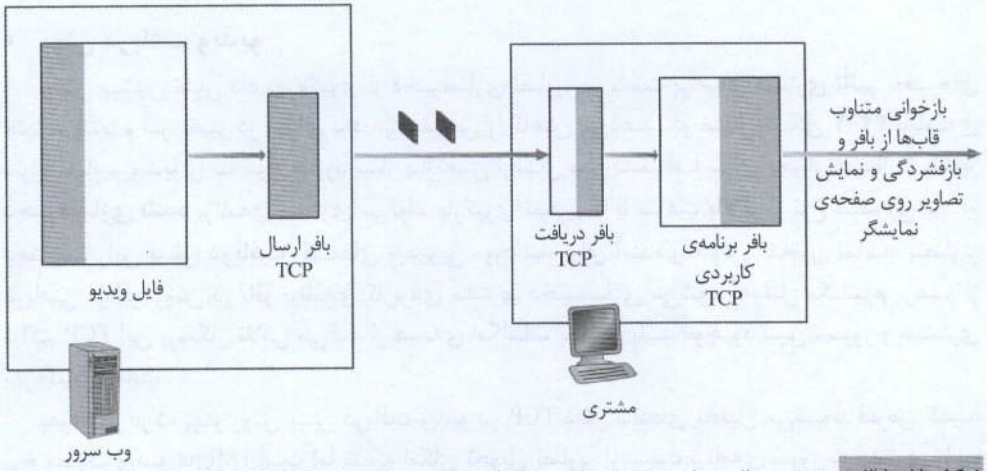
به‌منظور درک بهتر روش پیش دریافت ویدیو در TCP مثال ساده‌ای تحلیل می‌شود. فرض کنید نرخ مصرف ویدیو 1Mbps است اما شبکه امکان تحویل تصاویر از سمت برنامه‌ی سرور به مشتری را با سرعت ثابت 1.5Mbps دارد. در این شرایط برنامه‌ی مشتری نه تنها قادر به پخش ویدیو با زمان تأخیر کوتاه است بلکه قادر به افزایش مقدار تصویر ذخیره‌سازی شده‌ی برابر 500kbps نیز می‌باشد. در این فرایند اگر در آینده برنامه‌ی مشتری داده‌ای را با نرخ کمتر از 1Mbps در یک فاصله‌ی زمانی کوتاه دریافت نماید آنگاه برنامه‌ی مشتری قادر است باز پخش مستمر تصاویر را با تکیه بر منبع ذخیره‌ی خود در بافر انجام دهد. قبلاً نشان داده شد هنگامی که میانگین نهفتگی TCP تقریباً دو برابر سرعت رسانه است پخش سیال ویدیو از طریق TCP سبب کاهش ایستایی تصویر و نیز تاخیرهای در اثر ذخیره‌سازی ویدیو می‌شود.

۶- بافرهای برنامه‌ی کاربرد مشتری و TCP

تعامل بین دو برنامه‌ی مشتری و سرور برای پخش سیال HTTP در شکل ۷-۳۵ نشان داده شده است. در سمت برنامه‌ی سرور بخشی از فایل ویدیو که به‌رنگ سفید نشان داده شده است تاکنون به‌درون سوکت سرور منتقل شده است در حالی که بخش دیگر که رنگ تیره دارد باقی مانده‌ی فایل است که

1- Prefetching

بایستی گسیل شود. **پس از عبور از دریچه‌ی سوکت** بایت‌های مربوط در بافر ارسال TCP ذخیره می‌شوند و سپس به شبکه‌ی اینترنت براساس مطالب فصل ۳ گسیل می‌شوند. شکل ۷-۳۵ نشان می‌دهد که بافر ارسال TCP کاملاً اشغال شده است و بنابراین به‌صورت مقطعی از ارسال بایت‌های بیشتر از سوی سرور به‌درون سوکت ممانعت به‌عمل می‌آید. در سمت برنامه‌ی مشتری برنامه‌ی کاربردی مشتری یعنی پخش کننده‌ی رسانه (مانند: Real player) بایت‌های دریافتی از سوی بافر را از طریق سوکت برنامه‌ی مشتری خوانده و این بایت‌ها را به درون بافر برنامه‌ی کاربردی مشتری منتقل می‌کند. همزمان برنامه‌ی کاربردی مشتری به‌صورت متناوب قاب‌های تصویری را از بافر برنامه کاربردی مشتری اخذ و پس از انبساط (باز فشردگی) قاب‌ها، آنها را روی صفحه‌ی نمایش کاربر به تصویر می‌کشد. بدیهی است اگر حجم بافر برنامه‌ی کاربرد بزرگتر از فایل ویدیو باشد آنگاه تمامی فرآیند انتقال بایت‌ها از حافظه‌ی سرور به بافر برنامه‌ی کاربردی مشتری معادل بارگیری یک فایل معمولی از طریق HTTP تلقی می‌شود و برنامه‌ی مشتری با سرعتی که TCP به آن اجازه می‌دهد اطلاعات تصویری را از برنامه‌ی سرور اخذ می‌کند!



شکل ۷-۳۵ ♦ پخش سیال تصاویر ذخیره‌سازی شده از طریق پروتکل‌های HTTP/TCP.

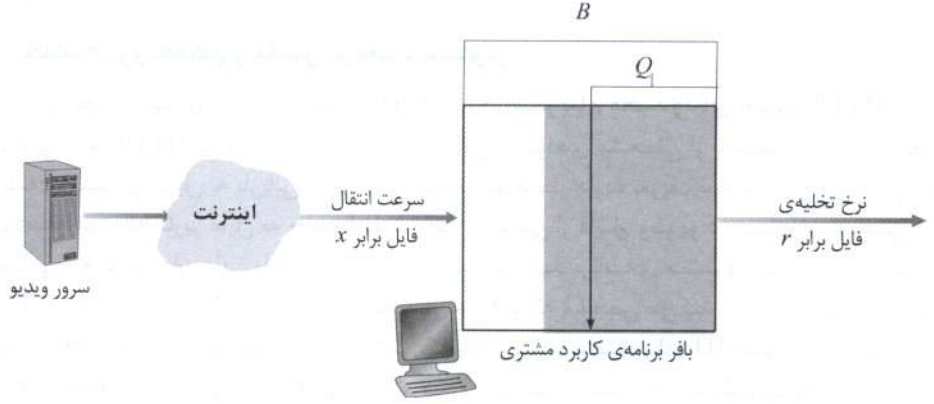
اکنون حالتی را درنظر بگیرید که کاربر دریافت ویدیو را در زمان فرآیند پخش متوقف می‌کند. در طول زمان توقف، بیت یا بایت‌های اطلاعات تصویری از بافر برنامه‌ی کاربرد مشتری خارج نخواهند شد حتی اگر بیت‌های اطلاعات به فرآیند ورود خود به بافر از طریق برنامه‌ی سرور ادامه دهند. اگر بافر برنامه‌ی کاربرد مشتری محدود باشد سرانجام اشباع خواهد شد و در نتیجه سبب ایجاد یک فشار معکوس در تمامی مسیر به‌سوی سرور می‌شود. به‌صورت خاص هنگامی که بافر برنامه‌ی کاربردی مشتری کاملاً اشباع شود هیچ بایت اطلاعاتی از بافر دریافت برنامه‌ی مشتری TCP خارج نخواهد شد و بنابراین آن هم کاملاً اشغال می‌شود. هنگامی که بافر ارسال TCP کاملاً پُر شود برنامه‌ی سرور قادر به ارسال بایت‌های بیشتر به درون سوکت نخواهد شد. بنابراین اگر کاربر پخش ویدیو را متوقف نماید ممکن است برنامه‌ی سرور مجبور به توقف فرآیند انتقال گردد که در این حالت سرور تا زمان آغاز پخش مجدد تصاویر از سوی کاربر مسدود می‌شود.

در حقیقت حتی هنگام بازپخش متعارف یک ویدیو که در آن فعالیت توقف موقت روی نمی‌دهد نیز اگر بافر برنامه‌ی کاربردی اشباع گردد فشار معکوس اشاره شده باز هم سبب پر شدن بافرهای TCP

می‌شود که نتیجه‌ی آن مجبور نمودن سرور به کاهش سرعت انتقال است. به‌منظور تعیین سرعت نتیجه شده از این فرآیند توجه به این نکته ضروری است، هنگامی که برنامه‌ی کاربرد مشتری تعداد f بیت را خارج می‌کند باعث ایجاد یک فضا با ظرفیت f بیت نیز در سمت بافر برنامه‌ی کاربرد مشتری می‌شود. این موضوع خود اجازه می‌دهد برنامه‌ی سرور f بیت افزون دیگر را نیز ارسال نماید. بنابراین سرعت انتقال سرور نمی‌تواند بیشتر از نرخ مصرف تصاویر در سمت مشتری باشد. به سخن دیگر **بافر برنامه‌ی کاربرد مشتری به‌صورت غیرمستقیم محدودیتی را روی سرعت ارسال ویدیو از سوی سرور به مشتری هنگامی تحمیل می‌کند که پخش از طریق پروتکل HTTP در حال انجام است.**

۷- تحلیل پخش سیال تصاویر

از نوعی الگوسازی ساده می‌توان به‌منظور پی‌بردن به جزئیات تأخیر آغاز بازپخش تصاویر و نیز ساکن شدن آنها در اثر کاهش تدریجی بافر برنامه‌ی کاربرد بهره گرفت. همان‌گونه که در شکل ۷-۳۶ نشان داده شده است اگر فرض شود B ظرفیت تعداد بیت بافر برنامه‌ی کاربردی مشتری باشد و Q تعداد بیت‌های لازم به‌منظور ذخیره‌سازی در بافر قبل از آغاز بازپخش تصاویر از سوی مشتری و نیز فرض شود r نرخ مصرف تصاویر از سوی برنامه مشتری باشد و به‌عبارتی دیگر اگر تصور شود $Q < B$ و r نیز نرخ دریافت بیت‌ها از بافر برنامه‌ی کاربرد مشتری در هنگام بازپخش می‌باشد. آنگاه اگر سرعت انتقال قاب‌های تصویری 30 قاب در ثانیه و هریک از آنها نیز در قالب حجم 100000 بیت فشرده‌سازی شده باشند، مقدارمحاسبه شده‌ی r برابر 3Mbps خواهد شد. به‌منظور پی‌بردن به عمق مطلب از بافرهای ارسال و دریافت TCP صرفنظر می‌شود.



شکل ۷-۳۶ ♦ تحلیل ذخیره‌سازی در سمت برنامه‌ی مشتری برای پخش تصاویر

فرض کنید برنامه‌ی سرور نرخ انتقال ثابت x را به‌منظور ارسال بیت‌ها در شرایطی اختیار می‌کند که بافر مشتری اشباع نشده است. بدیهی است این فرض بسیار ساده‌انگاری است زیرا نرخ ارسال TCP به‌دلیل فرآیند کنترل تراکم متغیر است. فرض کنید در زمان $t = 0$ بافر برنامه‌ی کاربرد تهی است و تصاویر آغاز به‌ورود در بافر برنامه‌ی کاربرد مشتری می‌کنند. اکنون می‌توان این پرسش را مطرح کرد در چه زمان $t = t_p$ بازپخش تصاویر آغاز می‌شود؟ و هنگام وقوع این رخداد در چه زمان $t = t_f$ بافر کاربرد مشتری اشباع می‌شود؟

ابتدا زمان t_p تعیین می‌شود. این زمان هنگامی است که تعداد Q بیت وارد بافر برنامه‌ی کاربرد مشتری شده‌اند و فرآیند بازپخش نیز آغاز شده است. یادآوری می‌شود که ورود بیت‌ها به بافر برنامه‌ی کاربرد مشتری در سرعت x انجام می‌شود و **هیچ** یک از بیت‌ها از این بافر قبل از آغاز بازپخش تصاویر خارج نخواهند شد. بنابراین مقدار زمان لازم برای انباشت Q بیت که همان زمان تاخیر ذخیره‌سازی آغازین است برابر $t_p = Q / x$ خواهد بود.

اکنون نوبت تعین t_f است یعنی زمانی که بافر برنامه‌ی کاربرد مشتری کاملاً اشباع می‌شود. ابتدا به این نکته توجه می‌شود که $x < r$ است به‌عبارت دیگر اگر نرخ ارسال سرور کمتر از سرعت مصرف ویدیو باشد، آنگاه بافر مشتری هرگز کامل و اشباع نخواهد شد! بلکه در زمان t_p بافر با نرخ r آغاز به تخلیه شدن می‌نماید و فقط با سرعت $x < r$ است که اشباع خواهد شد. سرانجام بافر مشتری به‌صورت کامل تخلیه می‌شود و در آن هنگام تصویر به‌صورت ثابت و ساکن روی صفحه‌ی نمایش ظاهر می‌شود درحالی که بافر مشتری در انتظار t_p ثانیه‌ی بعدی برای انباشت Q بیت از اطلاعات تصویر است. **بنابراین هنگامی که نرخ قابل دسترس در شبکه کمتر از سرعت ویدیو باشد آنگاه فرآیند بازپخش بین تناوب‌های مستمر بازپخش و ساکن شدن تصاویر به‌صورت مستمر جابه‌جایی انجام می‌دهد.** اکنون می‌توان زمان t_f را برای حالت $x > r$ محاسبه نمود. در این حالت با آغاز در زمان t_p حجم بافر از Q به B افزایش یافته و سرعت این افزایش $x - r$ است زیرا تخلیه‌ی بیت‌ها با سرعت r انجام می‌شود اما ورود آنها با نرخ x بیت برثانیه برمبنای شکل ۷-۳۶ است. بدیهی است **هنگامی که سرعت در دسترس در شبکه بیش از نرخ تصاویر است پس از یک تاخیر ذخیره‌سازی آغازین، کاربر تا انتهای اطلاعات ویدیویی می‌تواند از فرآیند مستمر بازپخش تصاویر لذت ببرد.**

۸- خاتمه‌ی زودهنگام و تغییر موقعیت تصاویر

در اغلب اوقات سیستم‌های پخش سیال HTTP از یک **سرپیام محدوده‌ی بایت HTTP** در پیام تقاضای HTTPGET استفاده می‌کنند تا از طریق آن محدوده‌ی مشخصی از بایت‌ها را که در آن لحظه برنامه‌ی مشتری تمایل به بازیابی آنها از فایل تصویر مورد نظر دارد، تعریف نمایند. این مهم از آن جهت سودمند است که کاربر مایل به جابه‌جایی یا پَرش به نقاطی از فایل ویدیو در زمان آینده می‌باشد. هنگامی که کاربر جابه‌جایی به مکان جدید را تقاضا می‌کند برنامه‌ی مشتری یک پیام درخواست HTTP جدید را با سرپیام محدوده‌ی بایت ارسال می‌کند که مشخص می‌کند از کدام بایت فایل سرور بایستی اقدام به‌ارسال اطلاعات نماید. هنگامی که سرور پیام تقاضای HTTP جدید را دریافت نمود آنگاه در موقعیتی قرار می‌گیرد که می‌تواند از هر درخواست پیشین صرفنظر نموده و در عوض آن بایت‌هایی را گسیل نماید که آغاز آنها با بایت مشخص شده در محدوده‌ی بایت تقاضا شده قرار گرفته است.

اکنون که در حال بررسی موضوع تغییر موضع تصاویر هستیم به‌صورت خلاصه این موضوع نیز تشریح می‌شود که هنگام انجام تغییر مواضع از سوی یک کاربر به نقطه‌ای در آینده و یا هنگامی که عملیات خاتمه‌ی زود هنگام بازپخش از سوی وی تقاضا می‌شود برخی از اطلاعات انتقال یافته از پیش که بازپخش نشده‌اند هرگز رویت نمی‌شوند و نتیجه‌ی حاصل آن اتلاف پهنای باند شبکه و نیز منابع سرور خواهد بود. به‌عنوان مثال فرض کنید بافر مشتری با تعداد B بیت اطلاعات در زمان t_0 پس از گذشت ویدیو اشباع می‌شود و در این هنگام کاربر به لحظه‌ای از زمان مانند $t > t_0 + B / r$ در فایل

تصاویر موقعیت ایجاد می‌کند و سپس ویدیو را از آن نقطه تا انتها مشاهده می‌کند. در این حالت تمامی B بیت اطلاعات موجود در بافر رویت نخواهند شد و پهنای باند شبکه و منابع سرور برای انتقال B بیت اطلاعات کاملاً تلف شده است. در شبکه‌ی اینترنت همواره پهنای باند تلف شده‌ی قابل ملاحظه‌ای به‌دلیل فرآیند خاتمه‌ی زودهنگام ویدیو پدید می‌آید که از نقطه نظر اقتصادی به‌ویژه برای پیوندهای بی‌سیم بسیار گران تمام می‌شود. به‌همین علت اغلب سیستم‌های پخش سیال تنها حجم محدودی از بافر در سمت برنامه‌ی کاربرد مشتری را مجاز به استفاده می‌کنند و یا آنکه مقدار تصاویر از پیش واکشی شده را با بهره‌گیری از سرپیام محدوده‌ی بایت در پیام تقاضای HTTP محدود می‌کنند.

خاتمه‌ی زود هنگام و نیز موقعیت تصاویر بسیار شباهت به‌حالتی دارد که یک نفر اقدام به آماده‌سازی یک وعده‌ی غذایی بزرگ می‌کند اما فقط بخش اندکی از آن را میل می‌کند و باقی مانده‌ی غذا را تحویل زباله‌دان می‌دهد و بنابراین منابع غذایی و زمان مصرف شده برای آماده‌سازی آن را عملاً برباد می‌دهد.

۷-۱۹ پخش سیال تطبیق‌پذیر و DASH

اگر چه پخش سیال HTTP به‌صورت گسترده در عمل از طریق سامانه‌هایی چون یوتیوب مورد استفاده قرار گرفته است اما از نقاط ضعف عمده‌ای نیز برخوردار است. از جمله همه‌ی برنامه‌های کاربردی مشتری بدون توجه به تغییرات مهم در مقدار پهنای باند در دسترس به‌مشتری از رمزگذاری مشابه تصویری بهره‌مند می‌شوند. این برابری در هر دو حالت یعنی هنگامی که برنامه‌های کاربردی مشتری در دو سوی ارسال و دریافت متفاوت هستند و نیز در گذر زمان برای مشتری‌های مشابه وجود د ارد. این مورد سبب توسعه‌ی نوع جدیدی از پخش سیال برپایه‌ی HTTP شده است که اصطلاحاً به آن **پخش سیال تطبیق‌پذیر پویا از طریق HTTP** یا به عبارت دیگ**DASH** [1] گفته می‌شود. در روش DASH تصاویر ویدیویی در قالب نسخه‌های گوناگون رمزگذاری می‌شوند که هر نسخه از نقطه‌نظر سرعت متفاوت می‌باشد و بنابراین از سطح کیفیت متفاوتی نیز بهره‌مند است. برنامه‌ی مشتری به‌صورت پویا متقاضی توده‌های ویدیویی می‌شود که از لحاظ طول زمانی شامل چندین ثانیه براساس نسخه‌های مختلف است. هنگامی که میزان پهنای باند در دسترس قابل توجه باشد برنامه‌ی مشتری به‌صورت طبیعی توده‌هایی را گزینش می‌کند که متعلق به نسخه‌های سریع‌تر هستند و نیز هنگامی که پهنای باند در دسترس پایین باشد به‌صورت طبیعی گزینش خود را از طریق نسخه‌های با نرخ کمتر انجام می‌دهد. از طریق پیام‌های تقاضای HTTPGET برنامه‌های مشتری توده‌های گوناگون را در هرلحظه‌ی زمانی گزینش می‌کنند.

از یک سو DASH امکان نرخ دستیابی متناوب به اینترنت را برای برنامه‌های مشتری به‌منظور پخش تصاویر در نرخ‌های گوناگون رمزگذاری فراهم می‌آورد و سبب می‌شود تا مشتریان کند سرعت 3G قادر به دریافت نسخه‌های سرعت پایین و درنتیجه کیفیت پایین‌تر تصویری باشند و ارتباط‌های از نوع کابل‌های نوری نیز قادر به دریافت نسخه‌های با کیفیت بسیار بالا گردند و از سوی دیگر DASH به برنامه‌های مشتری اجازه می‌دهد پهنای باند در دسترس را در صورتی که سرعت انتها به انتها در طول ارتباط تغییر نماید مطابق میل خود سازگار نمایند. این ویژگی اخیر برای کاربران سیار که مرتب در

1- Dynamic Adaptive Streaming over HTTP

حالت جابه‌جایی قرار می‌گیرند بسیار پراهمیت است زیرا آنها همواره ملاحظه می‌کنند که پهنای باند در دسترس آنها با توجه به جابه‌جایی آنها نسبت به پایانه‌های مبداء در نوسان و تغییر است. شرکت Comcast به‌عنوان مثال از یک سیستم پخش سیال تطبیق‌پذیر بهره‌گیری می‌کند که در آن هرفایل منبع ویدیویی در 8 تا 10 قالب‌بندی MPEG-4 رمزگذاری می‌شود و اجازه می‌دهد تا قالب‌های تصویری با بالاترین کیفیت به‌سوی یک برنامه‌ی مشتری پخش شوند در حالی که عملیات تطبیق‌پذیری در واکنش به تغییرات شبکه و شرایط تجهیزات صورت می‌پذیرد.

با بهره‌گیری از DASH هر نسخه‌ی ویدیویی در سرور HTTP با یک URL متفاوت ذخیره‌سازی می‌شود. برنامه‌ی سرور HTTP در این حالت بهره‌مند از یک **فهرست فایل** است که یک URL را برای هر نسخه همراه با سرعت انتقال بیت متناسب با آن تامین می‌کند. در آغاز برنامه‌ی مشتری از فهرست فایل، تقاضای نسخه‌ها را می‌کند و پس از کسب اطلاعات لازم در مورد نسخه‌های گوناگون، توده‌ای را در هر لحظه برمبنای مشخص نمودن یک URL و نیز محدوده بایت آن در پیام تقاضای HTTPGET گزینش می‌کند. هنگامی که مشغول بارگیری توده‌ها می‌باشد برنامه‌ی مشتری اقدام به‌اندازه‌گیری پهنای باند دریافتی می‌کند و یک **الگوریتم تعیین سرعت** را به اجرا در می‌آورد تا توده‌ی مورد نظر بعدی را گزینش نماید. بدیهی است اگر برنامه‌ی مشتری مقدار قابل توجهی از تصاویر ویدیویی ذخیره‌سازی شده را در اختیار دارد و اگر پهنای باند اندازه‌گیری شده‌ی دریافتی نیز بزرگ باشد آنگاه توده‌ای را گزینش می‌کند که متعلق به نسخه‌ی با نرخ پایین‌تر است. بنابراین پروتکل DASH اجازه می‌دهد مشتری با آزادی عمل تمام در میان سطوح با کمیت‌های مختلف به فعالیت خود ادامه و جابه‌جایی انجام دهد. از آنجا که یک افت ناگهانی سرعت انتقال از طریق جابه‌جایی بین نسخه‌های مختلف ممکن است سبب تضعیف کیفیت قابل ملاحظه در ویدیو شود آنگاه کاهش نرخ انتقال ممکن است از طریق بهره‌گیری از نسخه‌های چندگانه‌ی واسطی امکان‌پذیر گردد تا یک انتقال را به نرخی هدایت نماید که سرعت مصرف برنامه‌ی مشتری را به کمتر از پهنای باند دریافت در دسترس آن تنزل دهد. هنگامی که شرایط شبکه بهبود می‌یابد آنگاه برنامه‌ی مشتری توده‌هایی را گزینش می‌کند که متعلق به نسخه‌های با سرعت بالاتر هستند.

از طریق نظارت پویا روی پهنای باند در دسترس و نیز سطح بافر برنامه‌ی مشتری و نیز تطبیق نرخ انتقال با نسخه‌ی مطلوب از فایل تصویر، پروتکل DASH یک بازپخش مستمر را با بهترین سطح کیفیت ممکن بدون ایجاد توقف در حرکت قاب‌ها و یا صرفنظر نمودن پاره‌ای از آنها امکان‌پذیر می‌سازد. علاوه بر همه‌ی این موارد از آنجا که برنامه‌ی مشتری به جای برنامه‌ی سرور از هوشمندی لازم به‌منظور تعیین توده‌ی بعدی جهت ارسال بهره‌مند است آنگاه این روش مقیاس‌پذیری در سمت سرور را نیز محقق می‌سازد. سودمندی دیگر روش DASH آن است که برنامه‌ی مشتری می‌تواند از تقاضای محدوده بایت HTTP بهره‌گیری نماید تا به‌گونه‌ای دقیق مقدار از پیش واکشی شده‌ی تصاویر موجود در بافر خویش را کنترل نماید.

بحث پیرامون DASH را با این نتیجه‌گیری به‌پایان می‌رسانیم که در بسیاری از پیاده‌سازی‌ها نه تنها برنامه‌ی سرور نسخه‌های انبوهی از فایل‌های ویدیویی را ذخیره‌سازی می‌کند بلکه به‌صورت مجزا بسیاری از نسخه‌های صوتی را نیز نگهداری می‌کند. هر نسخه‌ی صوتی بهره‌مند از سطح کیفیت، نرخ انتقال و نیز URL خویش است. در اغلب این پیاده‌سازی‌ها برنامه‌ی مشتری به‌صورت پویا هر دو توده‌ی تصویر و صدا را گزینش می‌کند و به گونه‌ی موضعی نیز بازپخش صدا و تصویر را همزمان (سنکرون) می‌کند.

۷-۲۰ شبکه‌های توزیع محتوا (CDN)

در حال حاضر بسیاری از شرکت‌های رسانه‌های تصویری اینترنت جریان‌های تصویری چندین مگابیت بر ثانیه را برمبنای تقاضا بین میلیون‌ها کاربر روزانه توزیع می‌کنند. به‌عنوان مثال یوتیوب با مخزنی از صدها میلیون فایل تصویری هر روز در سراسر گیتی صدها میلیون جریان تصویری را بین کاربران خود توزیع می‌کند. پخش سیال تمامی این ترافیک به کلیه‌ی نقاط گیتی به مفهوم تامین بازپخش مستمر و تعامل‌گرایی بسیار بالا به‌وضوح چالشی بسیار عظیم است.

برای یک شرکت رسانه‌ی تصویری اینترنتی شاید ساده‌ترین رویکرد تامین خدمات پخش ویدیویی در قالب ساختن یک مرکز داده‌ای عظیم تک، ذخیره‌سازی همگی فایل‌های تصویری در آن مرکز داده و سپس پخش جریان‌های تصویری مستقیماً از مرکز داده به مشتریان در سراسر گیتی باشد. اما سه چالش مهم در این رویکرد وجود دارد. نخست آنکه اگر برنامه‌ی مشتری در فاصله‌ی قابل تأملی از مرکز داده قرار گرفته باشد آنگاه بسته‌های ارسالی تصویر از سمت سرور به مشتری مجبور به عبور از بین بسیاری از پیوندهای ارتباطی و احتمالاًعبور از میان بسیاری از ISPها خواهند بود که برخی از این ISPها احتمالاً در قاره‌های متفاوت اسکان یافته‌اند. اگر یکی از این پیوندها نهفتگی را تامین نماید که کمتر از نرخ مصرف فایل تصویری است آنگاه نهفتگی انتها به انتها نیز کمتر از نرخ مصرف خواهد شد و نتیجه‌ی مستقیم آن تاخیرهای طولانی است که سبب ایستایی تصاویر می‌شوند و برای کاربر ناخوشایند خواهد بود. یادآوری می‌شود در فصل ۱ نهفتگی انتها به انتها از یک جریان از طریق نهفتگی در پیوند تنگراه کنترل و مدیریت می‌شد. احتمال وقوع چنین رویدادی در این‌گونه انتقال‌ها نیز هرچه بر تعداد پیوندها در مسیر انتها به انتها افزوده می‌شود، افزایش می‌یابد. ضعف دوم آن است که یک فایل ویدیویی پربیننده احتمالاً به تعداد دفعات زیادی از طریق پیوندهای ارتباطی مشابه ارسال خواهد شد. این موضوع نه تنها سبب اتلاف پهنای باند شبکه می‌شود بلکه شرکت‌های رسانه‌ای تصویری اینترنت هزینه‌ی تامین امکانات برای ISPهایی که به مرکز داده متصل شده‌اند و نیز هزینه‌ی ارسال بخشی از همان بایت‌ها را به درون اینترنت به دفعات پرداخت می‌کنند. ضعف سوم این راهکار آن است که در یک مرکز داده‌ی تک، اگر مرکز داده یا پیوندهای آن به سمت اینترنت دچار اختلال شده و در نتیجه قادر به توزیع **هر** جریان تصویری نباشد آنگاه خود توصیف‌گر یک عیب و خرابی خواهد شد. به‌منظور تامین چالش توزیع حجم وسیعی از داده‌های تصویری به کاربران پراکنده در سراسر گیتی تقریباً همه‌ی شرکت‌های مهم پخش سیال تصاویر از امکانات **شبکه‌های توزیع محتوا (CDN)** [1] بهره می‌گیرند. یک CDN، سرورهای ساکن در چندین موقعیت جغرافیایی را مدیریت نموده و نسخه‌هایی از فایل‌های تصویری و دیگر انواع محتوای وب را که شامل اسناد، تصاویر و فایل‌های صوتی است، روی سرورها ذخیره‌سازی و تلاش می‌کند تا درخواست هرکاربر را به یک موقعیت CDN که تامین کننده‌ی بهترین تجربه‌ی تماشای تصویر برای کاربر است، هدایت نماید. شبکه‌های CDN ممکن است در مالکیت خصوصی قرار داشته باشند، به‌عنوان مثال CDN متعلق به شرکت گوگل تصاویر ویدیویی متعلق به یوتیوب متعلق به خود و دیگر انواع از محتواها را توزیع می‌کند. این امکان وجود دارد که یک CDN در مالکیت **ثالث** قرار گرفته باشد و توزیع محتوا را به نمایندگی از سوی چندین تامین کننده‌ی محتوا تأمین نماید. به‌عنوان مثال شبکه‌ی توزیع محتوا متعلق به Akamai یک CDN ثالث است که

1- Content distribution networks

محتواهای متعلق به Hulu و Netfix و ده‌ها تامین‌کننده‌ی محتوای دیگر را توزیع می‌کند. معمولاً از دو نوع معماری به‌منظور جایگذاری مکانی سرورها استفاده می‌شود. این دو معماری عبارتند از:

۱- نفوذ عمیق[1]

یک معماری که طلایه‌دار آن شرکت Akamai می‌باشد اصطلاحاً به **نفوذ عمیق** درون شبکه‌ی دستیابی یک ISP معروف شده است. در این روش از سرورهای خوشه‌ای در ISP های دستیابی سراسر گیتی استفاده می‌شود. رویکرد Akamai در این معماری بهره‌گیری از سرورهای خوشه‌ای تقریباً در 1700 مکان جغرافیایی است. هدف همجواری با کاربران است و در نتیجه بهبودی مقدار تاخیر تجربی کاربر و نهفتگی بیشتر از طریق کاهش تعداد پیوندها و مسیریاب‌های بین کاربر انتهایی (میزبان) و خوشه‌های CDN، در این شرایط کاربر محتوای مورد نظر خود را از طریق آن تامین می‌کند. به‌دلیل درصد توزیع بسیار بالا در این طرح مسئولیت نگهداری و مدیریت خوشه‌ها نیز بسیار دشوار می‌شود.

۲- انتقال به خانه[2]

فلسفه‌ی دوم طراحی که از سوی شرکت Limelight و بسیاری دیگر از تأمین‌کنندگان CDN مورد استفاده قرار گرفته است اصطلاحاً **انتقال ISP به خانه** است، به این صورت که با ساختن خوشه‌های بزرگ در تعداد مناطق کمتری مانند 10 ناحیه از مناطق کلیدی و سپس ارتباط این خوشه‌ها به یکدیگر با استفاده از شبکه‌های پرسرعت خصوصی کاربران در مجاورت خوشه‌ها قرار گیرند.

تفاوت این رویکرد با روش قبل آن است که به‌جای آنکه به درون ISPهای دستیابی نفوذ شود این‌گونه CDNها هر یک از سرورهای خوشه را در موقعیتی قرار می‌دهند که همزمان در نزدیکی PoPهای بسیاری از طبقه‌ی 1 از ISPها قرار می‌گیرند. به‌عنوان مثال در ایالات متحده‌ی آمریکا در فاصله چند کیلومتر از PoPهای هر دو شرکت AT&T و Verzion در یک شهر بزرگ خوشه‌های مربوط به سرورهای CDN جایگذاری شده‌اند. در مقایسه با طراحی نفوذ عمیق این رویکرد سبب کاهش سرآیندهای نگهداری و مدیریت می‌شود اما این کاهش قطعاً تأخیرهای بیشتر و نهفتگی کمتر را برای کاربران به ارمغان می‌آورد.

هنگامی که خوشه‌ها در مکان خود در یکی از معماری‌های بالا قرار گیرند شبکه‌ی CDN نسخه‌ی پشتیبان دومی را از محتوا در سراسر خوشه‌های خود نگهداری می‌کند. ممکن است یک CDN تمایلی به قرار دادن نسخه‌ای از هر فایل تصویری در هر خوشه نداشته باشد زیرا برخی از فایل‌های تصویری به ندرت تماشا می‌شوند و یا آنکه فقط در برخی از کشورها محبوبیت دارند. در حقیقت بسیاری از CDNها فایل‌های ویدیویی را به درون خوشه‌های خود تزریق نمی‌کنند بلکه به جای سیاست عرضه از یک رویکرد ساده‌ی تقاضا بهره می‌گیرند. به سخن دیگر اگر یک برنامه‌ی مشتری درخواست فایل تصویری از خوشه‌ای را نماید که تصاویری را ذخیره‌سازی نکرده است آنگاه خوشه آن فایل تصویری را از یک مرکز ذخیره‌سازی (مخزن) یا خوشه‌ای دیگر بازیابی می‌کند و سپس نسخه‌ای از آن را هنگامی‌که در حال پخش سیال تصویر به مشتری است به‌صورت همزمان در جایگاه خود ذخیره‌سازی می‌کند. این رویکرد بسیار شبیه به نهان‌سازی وب است که در فصل ۲ تشریح گردید.

1- Enter deep
2- Bring home

هنگامی‌که ظرفیت ذخیره‌سازی خوشه اشباع گردد تصاویر ویدیویی که علاقه‌مندان چندانی ندارند حذف خواهند شد.

- **عملیات CDN**

با شناخت دو رویکرد مهم بهره‌گیری از CDN اکنون به چگونگی عملکرد یک CDN می‌پردازیم. هنگامی‌که یک مرورگر در میزبان کاربر فرمان بازیابی یک فایل تصویری را از طریق تعریف URL آن صادر می‌کند یک CDN بایستی تقاضا را به گونه‌ای تفسیر نماید که امکان موارد ذیل میسر گردد.

۱- شناسایی یک خوشه سرور مناسب برای آن مشتری در آن لحظه.

۲- هدایت درخواست مشتری به یک سرور در آن خوشه.

در ادامه ملاحظه خواهد شد چگونه یک CDN یک خوشه‌ی مناسب را شناسایی می‌کند اما ابتدا چگونگی عملیات تلاقی و سپس هدایت یک تقاضا تحلیل می‌شود.

اغلب شبکه‌های CDN از امکانات یک DNS به‌منظور تلاقی و هدایت درخواست‌ها بهره می‌گیرند. یک بحث جالب پیرامون استفاده از امکانات DNS به صورتی است که در ادامه توضیح داده می‌شود. فرض کنید یک تأمین کننده‌ی محتوا مانند NetCinema از یک CDN ثالث بهره‌گیری می‌کند با نام KingCDN تا محتوای فایل‌های تصویری را بین کاربران خود توزیع نماید. روی صفحه‌ی وب NetCinema هر یک از فایل‌های ویدیویی شرکت بهره‌مند از یک URL مستقل شده‌اند که شامل رشته‌ی: Video و شناسه‌ای انحصاری برای آن ویدیو می‌باشند. به‌عنوان مثال Rostam7 در قالب: http://Video.netcinema.com/6Y7B23V اختصاص می‌یابد. بر مبنای آنچه شکل ۷-۳۷ نشان می‌دهد 6 گام ذیل برداشته می‌شود تا توزیع محتوا میسر گردد:

۱- کاربر صفحه‌ی وب متعلق به NetCinema را بازدید می‌کند.

۲- هنگامی که کاربر روی پیوند: http://Video.netcinema.com/6Y7B23V کلیک می‌کند میزبان کاربر یک تقاضای DNS به‌سوی Video.netcinema.com گسیل می‌کند.

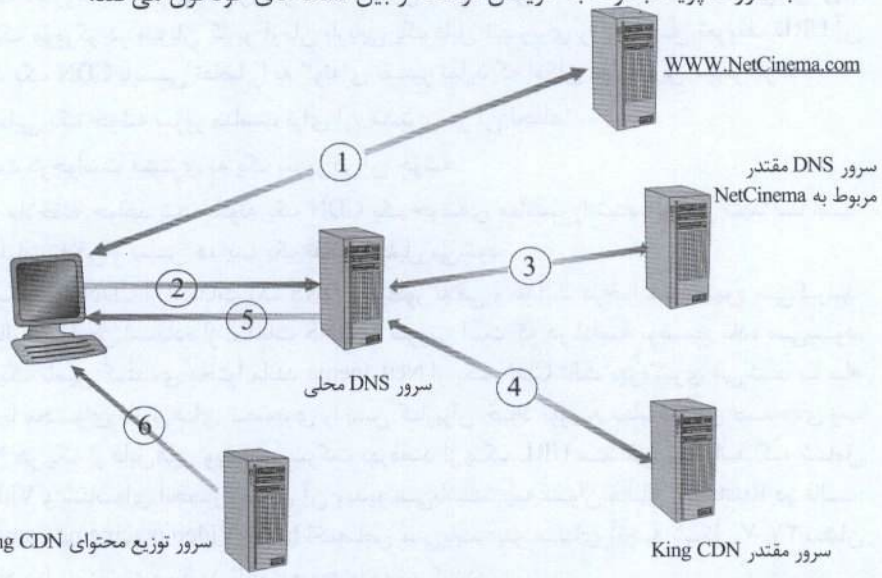
۳- سرور DNS محلی کاربر (LDNS) تقاضای DNS را به سمت یک سرور DNS مقتدر برای NetCinema گسیل می‌کند و در نتیجه آن سرور رشته‌ی Video در نام میزبان Video.netcinema.com را مشاهده می‌کند. به‌منظور انتقال تقاضای DNS به KingCDN به جای بازگشت یک آدرس IP سرور مقتدر DNS مربوط به NetCinema، یک نام میزبان در قلمرو KingCDN را به DNS بازگشت می‌دهد. به‌عنوان مثال نام میزبانی چون: 1105.kingcdn.com

۴- از این نقطه به بعد تقاضای DNS وارد زیرساخت خصوصی DNS مربوط به KingCDN می‌شود. در این هنگام سرور LDNS کاربر تقاضای بعدی خود را که اکنون برای: 1105.kingcdn.com است ارسال می‌کند و سیستم DNS مربوط به KingCDN سرانجام آدرس IP یک سرور محتوای KingCDN را به LDNS بازگشت می‌دهد. در این لحظه است که سیستم DNS وابسته به KingCDN امکانات لازم را برای سرور CDN و برنامه‌ی مشتری فراهم می‌کند تا محتوای تقاضا شده در اختیار برنامه‌ی مشتری قرار گیرد.

۵- پس از آن سرور LDNS آدرس IP گره‌ی CDN که محتوا را در اختیار دارد به میزبان کاربر تحویل می‌دهد.

۶- هنگامی که برنامه‌ی مشتری، آدرس IP مناسب برای یک سرور محتوای KingCDN را دریافت

نماید یک ارتباط مستقیم TCP را با سرور آن آدرس IP برقرار می‌سازد و یک پیام تقاضای HTTPGET را برای دریافت آن فایل تصویری صادر می‌کند. اگر از پروتکل DASH بهره‌گیری شود سرور ابتدا فهرستی از فایل‌ها را به همراه یک فهرست از URLها به مشتری گسیل می‌کند. هر یک از URLها وابسته به نسخه‌ای از فایل تصویری است و برنامه‌ی مشتری به‌صورت پویا مبادرت به گزینش توده‌ها از بین نسخه‌های گوناگون می‌کند.



شکل ۷-۳۷ ♦ سیستم **DNS** تقاضای یک کاربر را به یک سروی **CDN** هدایت می‌کند.

• روش‌های گزینش خوشه‌ها

در مرکزیت هر بهره‌گیری از CDN یک **استراتژی گزینش خوشه** وجود دارد که هدف آن هدایت فعال برنامه‌های مشتری کاربران به سمت یک خوشه سرور و یا یک مرکز داده در CDN است. همان‌گونه که ملاحظه شد شبکه‌ی CDN از طریق جدول DNS جستجوی مشتری، آدرس IP مربوط به سرور LDNS مشتری را یادگیری می‌کند. پس از این یادگیری CDN نیازمند گزینش خوشه‌ای مناسب برمبنای این آدرس IP می‌باشد. به‌طور کلی شبکه‌های CDN همواره از روش‌های انحصاری خود به‌منظور گزینش خوشه‌ها بهره می‌گیرند. در ادامه‌ی مطالب این زیربخش تعدادی از این رویکردها که هر یک نقاط قوت و ضعف ویژه‌ی خود را دارند تحلیل می‌شود.

ساده‌ترین استراتژی اختصاص مشتری به خوشه‌ای است که از لحاظ **جغرافیایی نزدیک‌تر** است. با بهره‌گیری از پایگاه داده‌های اقتصادی از نقطه‌نظر موقعیت اجتماعی مانند Quova و یا Max-Mind هر آدرس IP متعلق به یک LDNS به یک موقعیت جغرافیایی نگاشت می‌شود. هنگامی که یک تقاضای DNS از سوی یک LDNS خاص دریافت می‌شود شبکه‌ی CDN نزدیک‌ترین خوشه را از لحاظ موقعیت جغرافیایی گزینش می‌کند. به عبارت دیگر خوشه‌ای که حداکثر در فاصله‌ی چند کیلومتری از LDNS قرار گرفته است. چنین راهکاری می‌تواند برای انبوهی از کاربران سودمند باشد. اما برخی از مشتریان ممکن است از نقطه‌نظر مسیر گزینشی شبکه‌ی اینترنت نزدیک‌ترین به خوشه نباشند اگرچه از لحاظ جغرافیایی به خوشه‌ی سرورها نزدیک‌ترین باشند. علاوه بر آن یک ضعف ذاتی همه‌ی

رویکردهای مبتنی بر DNS آن است که برخی از کاربران انتهایی پیکربندی شده‌اند که از LDNSهای دوردست بهره گیرند و در این حالت مکان LDNS ممکن است به مراتب از موقعیت مکانی برنامه‌ی مشتری فاصله‌دار باشد. اضافه بر همه‌ی موارد اشاره شده این استراتژی ساده از تغییرات زمان‌های تأخیر و پهنای باند در دسترس در گذر زمان برای مسیرهای اینترنتی صرفنظر می‌کند و همواره یک خوشه را به یک مشتری خاص اختصاص می‌دهد.

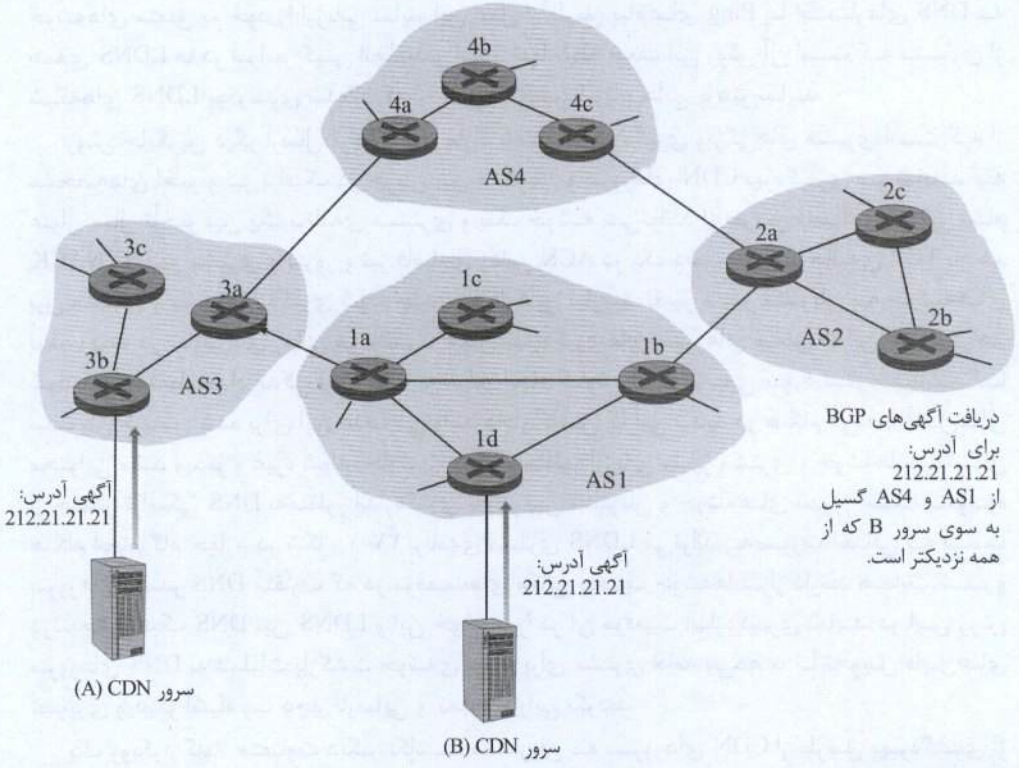
به‌منظور تعیین بهترین خوشه برای یک مشتری بر مبنای شرایط **ترافیک جاری**، شبکه‌ی CDN می‌تواند به‌صورت متناوب **اندازه‌گیری‌های زمان واقعی** از زمان‌های تأخیر و افت کارایی را بین خوشه‌ها و مشتریان انجام دهد. به‌عنوان مثال یک CDN می‌تواند به‌صورت متناوب هر یک از خوشه‌های متعلق به خود را ارزیابی نماید. این عمل از طریق پیام‌های Ping یا جستارهای DNS به همه‌ی LDNS ها در سراسر گیتی انجام‌پذیر است. تنها نقطه ضعف این روش آن است که بسیاری از شبکه‌های LDNS پیکربندی شده‌اند که از پاسخ به چنین ارزیابی‌هایی پرهیز نمایند.

روش جایگزین دیگر ارسال ترافیک‌های مازاد به‌منظور اندازه‌گیری ویژگی‌های مسیری است که از مشخصه‌های اخیر و نیز ترافیک در جریان بین مشتریان و سرورهای CDN بهره‌گیری می‌نمایند. به عنوان مثال تأخیر بین یک برنامه‌ی مشتری و یک خوشه می‌تواند از طریق فاصله‌ی زمانی پیام SYNACK بین مشتری و سرور و نیز فاصله‌ی زمانی ACK در یک دست‌دهی سه‌جانبه‌ی TCP باز هم بین مشتری و سرور اندازه‌گیری شود. چنین راهکارهایی نیازمند تغییر مسیر مشتریان به خوشه‌های نیمه بهینه در تناوب‌های زمانی می‌باشد. تا امکان اندازه‌گیری‌های مسیرهای متعلق به این خوشه فراهم شود. اگرچه فقط تعداد اندکی از تقاضاها به‌منظور انجام کاوش این ارزیابی مورد نیاز می‌باشد اما مشتریان گزینش شده برای این هدف می‌توانند دچار کاهش کارایی شدید در هنگام دریافت فایل‌های محتوایی مانند ویدیو و غیره شوند. جایگزین دیگر به‌منظور ارزیابی مسیر مشتری و خوشه‌ها بهره‌گیری از جستار ترافیکی DNS به‌منظور اندازه‌گیری تأخیر بین مشتریان و خوشه‌های سرور است. به‌ویژه هنگام انجام گام چهارم در شکل ۷-۳۷ برنامه‌ی مشتری LDNS می‌تواند به‌صورت اتفاقی به سمت سرورهای مقتدر DNS متفاوت که در موقعیت‌های مکانی مختلف خوشه‌ها قرار دارند هدایت شده و درنتیجه ترافیک DNS بین LDNS و این خوشه‌ها را در آن موقعیت اندازه‌گیری نماید. در این روش سرورهای DNS به عملیات بازگشت خوشه‌ی بهینه برای مشتری ادامه می‌دهند تا تحویل فایل‌های تصویری و دیگر اشیاء وب دچار نارسایی و کمبود کارایی نگردند.

یک رویکرد کاملاً متفاوت دیگر نگاشت مشتریان به سرورهای CDN از طریق بهره‌گیری از **پروتکل اینترنت هر فراخوانی**[1] است که [RFC1546] مبنای آن می‌باشد. اندیشه‌ی اصلی IPanycast وادار نمودن مسیریاب‌ها به هدایت بسته‌های اطلاعاتی مشتریان براساس محاسبه‌ی پروتکل BGP به نزدیک‌ترین خوشه است. لازم به یادآوری است روش **هر فراخوانی IP** به معنی انتقال به نزدیک‌ترین روتر یا مسیریاب است. در شکل ۷-۳۸ نشان داده شده است هنگامی که مرحله‌ی پیکربندی IPanycast در جریان است شبکه‌ی CDN آدرس IP **مشابهی** را به هر یک از خوشه‌ها اختصاص می‌دهد و از BGP **استاندارد بهره‌گیری می‌کند** تا این آدرس IP را از هر یک از موقعیت‌های مختلف خوشه **آگهی** نماید. هنگامی که مسیریاب BGP چندین آگهی مسیر را دریافت نماید که همگی متعلق به آدرس IP مشابه هستند همه‌ی این آگهی‌ها را به عنوان تأمین مسیرهای

1- IP anycast

گوناگون برای یک مکان فیزیکی قلمداد می‌کند درحالی که واقعیت آن است که آگهی‌های اشاره شده برای مسیرهای گوناگون مربوط به موقعیت‌های فیزیکی **متفاوت** است. با تعقیب این فرآیند استاندارد عملیاتی مسیریاب BGP آنگاه **بهترین** مسیر به آدرس IP را بر مبنای روش گزینش مسیر موضعی خود انجام می‌دهد. به‌عنوان مثال نزدیک‌ترین را براساس شمارش AS-hop تعیین می‌کند. به سخن دیگر اگر یکی از مسیرهای BGP که وابسته به موقعیت است تنها فقط به فاصله‌ی یک گره‌ی AS از مسیریاب قرار دارد و همه‌ی مسیرهای BGP دیگر که وابسته به موقعیت‌های متفاوت هستند به اندازه دو یا چندین گره‌ی AS فاصله دارندآنگاه مسیریاب BGP به‌صورت خاص مسیر بسته‌ها را در جهت موقعیتی هدایت می‌کند که فقط نیازمند گذر از **یک** AS باشند.



شکل ۷-۳۸ ♦ بهره‌گیری از IPanycast به‌منظور هدایت مشتریان به نزدیک‌ترین خوشه CDN.

پس از انجام این مرحله‌ی ابتدایی پیکربندی، شبکه‌ی CDN می‌تواند وظیفه‌ی اصلی خود که توزیع فایل‌های محتوایی است را انجام دهد. هنگامی که هر مشتری تمایل به تماشای ویدیویی داشته باشد سیستم DNS مربوط به شبکه‌ی CDN یک آدرس anycast را بدون توجه به موقعیت مکانی مشتری بازگشت می‌دهد. هنگامی که برنامه‌ی مشتری بسته‌ای به آن آدرس IP گسیل می‌کند بسته‌ی اشاره شده به **نزدیک‌ترین** خوشه براساس محاسبه‌ی جداول گسیل از پیش پیکربندی شده به‌وسیله‌ی BGP فرستاده می‌شود. برتری این روش آن است که یافتن خوشه‌ای را انجام می‌دهد که به کاربر نزدیک‌تر است در حالی که روش‌های پیشین خوشه‌ای از سرورها را می‌یافتند که به برنامه‌ی مشتری LDNS

نزدیک‌تر بود. در هر صورت استراتژی فراخوانی IP بازهم طبیعت پویای اینترنت در مقابل مقیاس‌های کوتاه‌مدت را درنظر نمی‌گیرد.

علاوه بر همه‌ی موارد اشاره شده‌ی موضوعات مهم وابسته به شبکه مانند تأخیر، اتلاف و کارایی پهنای باند عوامل مهم دیگری نیز وجود دارند که بایستی در هنگام طراحی یک استراتژی گزینش خوشه مورد توجه قرار گیرند. یکی از این عوامل بار ترافیک روی خوشه‌ها می‌باشد. برنامه‌های مشتری ممکن است به‌گونه‌ای گزینش شوند که از یک ISP خاص به‌منظور حمل ترافیک CDN به مشتری استفاده شود و هزینه‌های متفاوت ساختارها رادر روابط ساختمانی بین ISPها و اپراتورهای خوشه منظور نماید.

• زیرساخت شبکه‌ی CDN شرکت گوگل

به‌منظور پشتیبانی از گروه انبوهی از خدمات ابری از جمله جست‌وجو، جی‌میل، تقویم، محتواهای تصویری یوتیوب، نقشه‌خوانی، اسناد و شبکه‌های اجتماعی شرکت گوگل از یک شبکه‌ی خصوصی گسترده به همراه یک زیرساخت شبکه‌ی توزیع محتوا (CDN) بهره گرفته است. زیرساخت شبکه CDN شرکت گوگل بهره‌مند از سه رده خوشه‌های سرور است که در ذیل تشریح می‌شوند:

- تعداد 8 اَبَر مرکز داده که 6 فقره از آنها در ایالات متحده‌ی آمریکا قرار دارند و دو فقره‌ی باقی مانده در قاره‌ی اروپا اسکان یافته‌اند. هر یک از این مراکز داده‌ی عظیم نزدیک به 100,000 سرور را پشتیبانی می‌کنند. این مراکز داده مسئولیت خدمات رسانی پویای محتوا که اغلب حالت خصوصی نیز دارند را برعهده دارند. موارد خدمات رسانی آنها شامل نتایج کاوش‌ها و پیام‌های gmail است.
- در حدود 30 خوشه‌ی سرور با بهره‌گیری از استراتژی **انتقال به خانه** وجود دارند که هرکدام شامل تعداد 100 تا 500 سرور می‌شوند. خوشه‌ها از لحاظ موقعیت مکانی در سراسر گیتی توزیع شده‌اند و هر موقعیت به‌گونه‌ی خاص در مجاورت POPهایی قرار دارد که بهره‌مند از ISPهای رده‌بندی چندگانه در سطح 1 می‌باشند. این خوشه‌ها مسئولیت خدمات‌رسانی به محتواهای ایستا یعنی، غیرپویا شامل فایل‌های تصویری یوتیوب را برعهده دارند.
- بخش دیگر این زیرساخت شامل صدها خوشه از نوع **نفوذ عمیق** است که هر خوشه از لحاظ موقعیت مکانی درون یک ISP دستیابی اینترنت قرار دارد. در این‌جا یک خوشه شامل ده‌ها سرور درون یک قفسه‌ی (راک) تک است. این سرورهای نفوذ عمیق تقسیم‌بندی‌ها و به عبارتی تمایزسازی TCP در فصل ۳ را انجام می‌دهند و به خدمت محتواهای غیرپویا شامل صفحات وب ایستا در بدنه‌ی نتایج جست‌وجوهای اینترنتی درآمده‌اند.

همه‌ی این مراکز داده و مکان‌های خوشه‌ها با شبکه‌ی خصوصی گوگل پیوند خورده‌اند و تشکیل یک شبکه‌ی کامپیوتری فوق‌العاده هیجان‌انگیز را به‌عنوان بخشی از یک AS عظیم داده‌اند. هنگامی‌که یک کاربر درخواست جست‌وجویی را صادر می‌کند در اغلب موارد ابتدا آن درخواست از طریق ISP محلی به نزدیک‌ترین سرور نهان نفوذ عمیق گسیل می‌شود و از آنجا محتوای ایستا بازیابی می‌گردد. هنگامی‌که سرور نهان مشغول تأمین محتوای موردنظر است این سرور هم‌چنین تقاضایی را از طریق شبکه‌ی خصوصی گوگل به یکی از مراکز داده‌ی عظیم که در بالا به آنها اشاره رفته است گسیل می‌کند. از این مراکز داده نتایج جست‌وجوهای شخصی‌سازی شده بازیابی می‌شود. برای دریافت یک فایل تصویری یوتیوب محتوای ویدیویی ممکن است از یکی از سرورهای

نهان **انتقال به خانه** دریافت گردد. در حالی که بخش‌هایی از اطلاعات صفحه‌ی وب که در اطراف آن ویدیو قرار گرفته‌اند ممکن است از سرور نهان نفوذ عمیق نزدیک به برنامه‌ی مشتری دریافت گردد. بدیهی است آگهی‌های تجاری که ممکن است وابسته به فایل ویدیویی باشند نیز از مراکز داده‌ی اشاره شده بازیابی می‌شوند. به‌طور خلاصه به جز برای ISPهای محلی **خدمات ابری گوگل** در قالب کلی از طریق یک زیرساخت شبکه که مستقل از اینترنت عمومی است تأمین می‌شود.

۷-۲۱ موضوعات مطالعاتی: شبکه‌های Netfix، یوتیوب و Kankan

موارد مربوط به این فصل و نیز پخش سیال فایل‌های تصویری ذخیره‌سازی شده را با مطالعه پیرامون سه بهره‌گیری مقیاس وسیع و بسیار موفق به پایان می‌بریم. این شبکه‌های محتوای موفق عبارتند از: Netfix، یوتیوب و Kankan. در این بخش پایانی ملاحظه خواهد شد همه‌ی این سیستم‌ها از رویکردهای گوناگون به‌منظور رسیدن به هدف خود بهره‌گیری می‌کنند در حالی که همگی آنها اصول بنیادی تشریح شده در این فصل را راهنمای خود قرار داده‌اند.

- **Netfix**

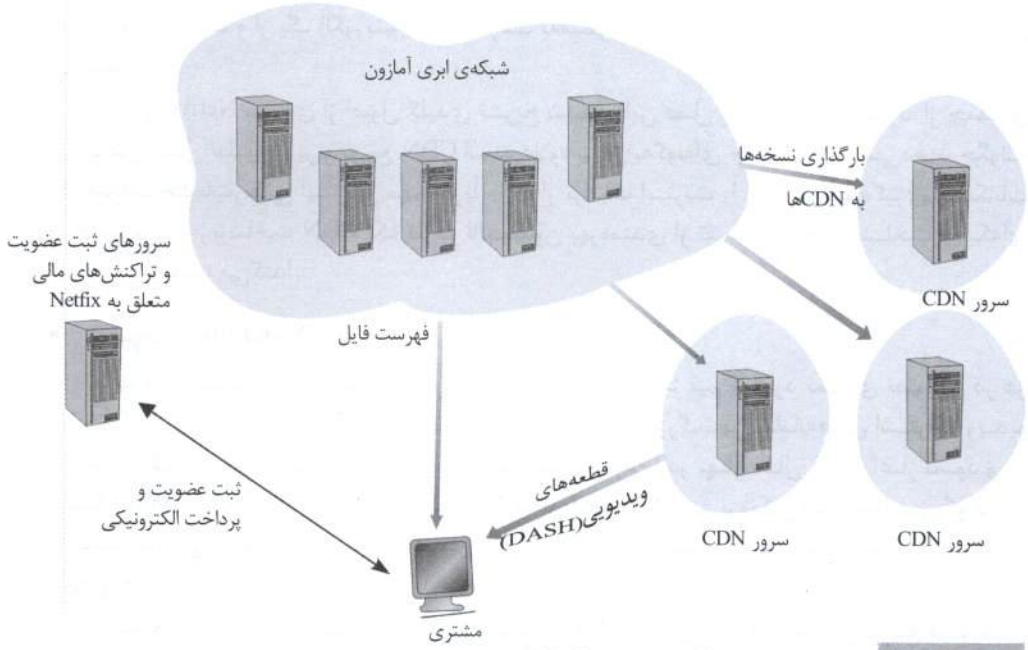
با تأمین تقریباً 30 درصد از ترافیک اینترنت بارگیری در ایالات متحده‌ی آمریکا در سال 2011 میلادی شبکه‌ی Netfix تبدیل به یکی از پیشتازان عرصه‌ی تأمین خدمات برای رسانه‌های تصویری آنلاین و برنامه‌های تلویزیونی آمریکا شده است. به‌منظور تسریع در به‌کارگیری خدمات خود در مقیاس وسیع Netfix از خدمات ابری ثالث و شبکه‌ی CDN بهره‌گیری نموده است. این شرکت یکی از موارد جالب در میان شرکت‌های توزیع محتوا است که از خدمات وسیع آنلاین به‌منظور اجاره‌ی سرورها، پهنای باند، ذخیره‌سازی و خدمات پایگاه‌های داده از شرکت‌های ثالث استفاده نموده است و این در حالی است که Netfix بهره‌مند از هیچ زیرساخت ویژه‌ی خود نمی‌باشد. به‌منظور رسیدن به اهداف درازمدت خود Netfix روش‌های بسیاری را به کار گرفته است که در بخش‌های پیشین این فصل تحلیل شدند از جمله‌ی این موارد توزیع ویدیو با استفاده از چندین CDN و نیز پخش سیال تطبیق‌پذیر از طریق HTTP است.

شکل ۷-۳۹ معماری اصلی سامانه‌ی پخش سیال تصویری Netfix را نشان می‌دهد. این سامانه بهره‌مند از چهار مؤلفه‌ی اصلی است: (۱) سرور ثبت نام عضویت و پرداخت، (۲) شبکه‌ی ابری آمازون، (۳) چندین تأمین‌کننده‌ی CDN و (۴) مشتریان. در زیرساخت سخت‌افزاری خود Netfix سرورهای ثبت عضویت و پرداخت را نگهداری می‌کند که عملیات مربوط به اطلاعات ثبت حساب‌های مشتریان جدید و نیز پرداخت‌های اعتباری الکترونیکی آنها را نگهداری می‌کنند، به استثنای این عملیات پایه‌ای Netfix خدمات آنلاین خود را نیز از طریق به‌کارگیری ماشین‌های مجازی در شبکه‌ی ابری آمازون به اجرا در می‌آورد. برخی از عملیاتی که در شبکه‌ی ابری آمازون انجام می‌شوند عبارتند از:

- **هضم محتوی.** قبل از آنکه Netfix قادر به توزیع یک فایل تصویری به مشتریان خود باشد ابتدا بایستی فایل تصویری را اصطلاحاً پردازش و هضم نماید. شبکه‌ی Netfix نسخه‌های اصلی فیلم‌ها را از تهیه‌کنندگان محتوا دریافت و سپس آنها را روی میزبان‌های مستقر در شبکه‌ی ابری آمازون بارگذاری می‌کند.
- **پردازش محتوا.** نرم‌افزارها و به‌عبارت دیگر ماشین‌های مجازی در ابر آمازون قالب‌بندی‌های متفاوت و انبوهی را برای هر یک از فیلم‌ها تهیه می‌کنند. این قالب‌بندی‌های

متفاوت مناسب برنامه‌های کاربردی پخش تصاویر گوناگون است که روی انبوهی از برنامه‌های مشتری کاربران مستقر در رایانه‌های شخصی، کامپیوترهای همراه، گوشی‌های هوشمند و تبلت‌ها و حتی کنسول‌های متصل به دستگاه‌های TV و بازی نصب شده‌اند. برای هر یک از این قالب‌ها نسخه‌های مناسب و متفاوت خلق می‌شود که بهره‌مند از سرعت‌های (bps) گوناگون هستند. این موضوع سبب می‌شود با بهره‌گیری از DASH پخش سیال تطبیق‌پذیر از طریق HTTP فراهم آید.

- **نسخه‌های قابل بارگذاری روی شبکه‌های CDN**. هنگامی‌که همه‌ی نسخه‌های یک فیلم خلق شدند آنگاه میزبان‌های مستقر در شبکه‌ی ابری آمازون نسخه‌های اشاره شده را روی شبکه‌های توزیع محتوا (CDN) بارگذاری می‌کنند.



شکل ۷-۳۹ ♦ سامانه‌ی پخش سیال تصویری Netfix.

به‌منظور تحویل فایل‌های تصویری به مشتریان متقاضی، Netfix از فناوری گسترده‌ی CDN بهره‌گیری می‌کند تا لحظه‌ی نگارش و ویراست ششم کتاب شبکه‌های کامپیوتری «رویکرد از فراز به فرود» Netfix از تعداد 3 شرکت توزیع محتوای ثالث (CDN) به‌صورت همزمان استفاده نموده است. این 3 عبارتند از: Akamai، LimeLight و Level-3.

اکنون که عناصر معماری Netfix توصیف شدند نگاه عمیق‌تری می‌اندازیم به تعامل بین برنامه‌های مشتری و سرور گوناگون که در تحویل محتوای تصویری دخالت مستقیم دارند، صفحات وب که به‌منظور مرور کتابخانه‌ی فایل‌های تصویری Netfix وجود دارند همگی از طریق سرورهای مستقر در شبکه‌ی ابری آمازون خدمات‌رسانی می‌شوند. هنگامی‌که کاربر یک فیلم را گزینش می‌کند و سپس بر روی کلید: PlayNow کلیک می‌کند برنامه‌ی مشتری کاربر یک فهرست فایل را نیز از سرورهای ابر آمازون دریافت می‌دارد. این فهرست فایل شامل انواع اطلاعات است. از جمله‌ی این اطلاعات فهرست

رتبه‌بندی CDN و آدرس‌های URL برای نسخه‌های مختلف یک فیلم است که برای بازپخش DASH مورد استفاده قرار می‌گیرد. فهرست رتبه‌بندی CDN از طریق Netfix تعریف می‌شود و ممکن است از یک فرآیند پخش به فرآیند بعدی تغییر نماید. به‌صورت ویژه مشتری، شبکه‌ی توزیع محتوایی را گزینش می‌کند که بهره‌مند از بالاترین امتیاز رتبه‌بندی در فهرست فایل‌ها می‌باشد. پس از گزینش یک CDN از سوی برنامه‌ی مشتری شبکه‌ی CDN با اعمال نفوذ در سیستم DNS مشتری را به سوی سرور خاص در یک CDN مشخص بر مبنای موارد تشریح شده در قبل هدایت می‌کند. برنامه‌ی مشتری و سرور CDN سپس از طریق بهره‌گیری از DASH با یکدیگر تعامل می‌کنند. معمولاً برنامه‌ی مشتری از یک سرپیام محدوده‌ی بایت در پیام تقاضای HTTPGET استفاده می‌کند تا توده‌هایی از نسخه‌های گوناگون فیلم را تقاضا نماید. روش Netfix استفاده از توده‌هایی است که تقریباً طول زمانی آنها 4 ثانیه است. هنگامی که توده‌ها در حال بارگیری هستند برنامه‌ی مشتری نهفتگی دریافتی را اندازه‌گیری می‌کند و از یک الگوریتم تعیین سرعت به‌منظور محاسبه‌ی کیفیت توده‌ی بعدی بهره‌گیری می‌نماید.

شبکه‌ی Netfix بسیاری از اصول کلیدی تشریح شده در این فصل را به کار می‌گیرد. از جمله‌ی آنها پخش سیال تطبیق‌پذیر و توزیع CDN است. علاوه بر آن به‌گونه‌ای جذاب نشان می‌دهد چگونه یک شرکت خدمات‌رسانی اینترنتی مهم تقریباً 30% از ترافیک اینترنت را از طریق بهره‌گیری از امکانات شبکه‌ی ابری و زیرساخت CDN یک طرف ثالث بدون بهره‌مندی از تقریباً هیچ زیرساخت شبکه‌ای ویژه‌ی خود تولید می‌کند!

- **یوتیوب (YouTube)**

تقریباً با نیم‌میلیارد فایل تصویری ویدیویی در مخزن خود و نیز نیم میلیارد تماشای تصویری در هر روز بنابر آمار انتهای سال 2011 میلادی شرکت یوتیوب یکی از بزرگ‌ترین سامانه‌های اشتراک ویدیو در سراسر گیتی به‌شمار می‌آید. یوتیوب خدمات‌رسانی خود را در بهار سال 2005 آغاز نمود و در زمستان 2006 میلادی در مالکیت شرکت گوگل قرار گرفت. اگرچه پروتکل‌ها و طراحی گوگل و یوتیوب خصوصی هستند اما از طریق چندین تلاش اندازه‌گیری مستقل از یکدیگر می‌توان به یک درک اولیه از چگونگی عملکرد یوتیوب پی برد.

همانند Netfix شبکه‌ی یوتیوب بهره‌گیری گسترده‌ای از فناوری CDN به‌منظور توزیع فایل‌های تصویری خود می‌کند. اما برعکس Netfix شرکت گوگل از شبکه‌ی CDN ثالث استفاده نمی‌کند بلکه از شبکه‌ی توزیع محتوای خصوصی خود به‌منظور توزیع فیلم‌ها و محتواهای تصویری یوتیوب بهره‌گیری می‌کند. گوگل چندین خوشه سرور در صدها مکان جغرافیایی نصب نموده است. از زیرمجموعه‌ای در حدود 50 مکان به‌منظور توزیع تصاویر ویدیویی YouTube بهره‌برداری می‌کند. شرکت گوگل هم‌چنین از DNS به‌منظور هدایت تقاضاهای مشتریان به یک خوشه‌ی خاص بر مبنای توضیحات ارایه شده در همین فصل استفاده می‌کند. در اغلب موارد استراتژی گزینش خوشه از سوی گوگل هدایت یک برنامه‌ی مشتری به خوشه‌ای است که اندازه‌ی RTT بین مشتری و خوشه‌ی سرور در کمترین میزان ممکن قرار می‌گیرد. به‌منظور برقراری توازن بار در سراسر خوشه‌ها برخی اوقات برنامه‌ی مشتری از طریق DNS به خوشه‌ی دورتر هدایت می‌شود. علاوه بر آن اگر یک خوشه بهره‌مند از فایل تقاضا شده نمی‌باشد به‌جای واکشی آن فایل از نقطه‌ای دیگر و ارسال آن به مشتری، خوشه ممکن است یک پیام تغییر جهت HTTP را بازگشت دهد و بنابراین برنامه‌ی مشتری را به سوی خوشه‌ی دیگری هدایت نماید.

شبکه‌ی یوتیوب از پخش سیال HTTP بهره‌گیری می‌کند و در اغلب اوقات تعداد اندکی از نسخه‌های گوناگون را برای یک فایل ویدیویی در اختیار می‌گذارد که هر یک از آنها از نرخ انتقال متفاوت و سطح کیفیت وابسته به آن سرعت بهره‌مند هستند. تا پایان سال 2011 یوتیوب از پروتکل پخش سیال تطبیق‌پذیر مانند DASH استفاده نکرده است بلکه به جای آن از کاربر می‌خواهد تا به‌صورت دستی نسخه‌ی مورد نظر را گزینش نماید. به‌منظور صرفه‌جویی در پهنای باند و منابع سرور که ممکن است در جابه‌جایی اطلاعات فایل و یا خاتمه‌ی زودهنگام نمایش فایل تلف شوند یوتیوب از سرپیام تقاضای محدوده‌ی بایت HTTP استفاده می‌کند تا جریان انتقال داده را پس از آنکه حجم مطلوبی از فایل تصویری واکشی شده است را محدود نماید.

هر روز چندین میلیون فایل ویدیویی روی یوتیوب بارگذاری می‌شود. نه‌تنها ویدیوهای یوتیوب از طریق HTTP به‌صورت مشتری به سرور پخش می‌شوند بلکه کاربران یوتیوب به بارگذاری فایل‌های تصویری خود از سمت برنامه‌های مشتری به سرور از طریق HTTP نیز اقدام می‌کنند. شبکه‌ی یوتیوب هر یک از فایل‌های دریافتی را پردازش و سپس آنها را تبدیل به قالب ویدیویی یوتیوب نموده و نسخه‌های چندگانه‌ای را با سرعت‌های مختلف بارگیری خلق می‌کند.این فرآیند به‌طور کامل درون مراکز داده‌ای انحصاری گوگل صورت می‌پذیرد. بنابراین تمایزی کاملاً متفاوت با Netfix دارد که خدمات خود را به‌طور کامل از طریق زیرساخت‌های ثالث تأمین می‌نمود. شرکت گوگل تمامی بدنه‌ی خدماتی یوتیوب را درون زیرساخت وسیع و گسترده‌ی مراکز داده‌ی خود انجام می‌دهد که بایستی به این زیرساخت شبکه‌ی CDN خصوصی و شبکه‌ی ارتباطات جهانی گوگل به مراکز داده و خوشه‌های CDN را نیز اضافه نمود.

- **شبکه‌ی Kankan**

ملاحظه شد در هر دو شبکه‌ی Netfix و یوتیوب سرورهای از طریق شبکه‌های توزیع محتوای خصوصی یا ثالث عملیات پخش تصاویر به مشتریان را هدایت می‌نمودند. دو شبکه‌ی Netfix و یوتیوب نه‌تنها بایستی هزینه‌ی سخت‌افزاری این سرورها را به‌صورت مستقیم و یا غیرمستقیم از طریق مالکیت و یا اجاره‌ی آنها تأمین نمایند بلکه هزینه‌ی پهنای باند مورد استفاده‌ی سرورها را برای توزیع فایل‌ها نیز بایستی تأمین نمایند. با در نظر گرفتن مقیاس خدمات ارایه شده و نیز مقدار پهنای باند مصرفی چنین بهره‌برداری مشتری – سرور بسیار پرهزینه خواهد شد.

پایان بخش مطالب این بخش توصیف رویکردی کاملاً متفاوت برای تأمین ویدیو براساس تقاضا از طریق اینترنت و در یک مقیاس وسیع است. رویکردی که اجازه می‌دهد تأمین کننده‌ی خدمات به‌صورت چشمگیری هزینه‌های پهنای باند و زیر ساخت خود را کاهش دهد. ممکن است تاکنون به این نتیجه رسیده باشید که این رویکرد قطعاً از پروتکل P2P در عوض مشتری – سرور تحویل فایل‌های تصویری را انجام می‌دهد. حدس شما کاملاً درست است. بهره‌گیری از پروتکل P2P به‌منظور تحویل محتواهای تصویری با موفقیت چشمگیری از سوی چندین شرکت چینی ازجملهKankan که از سوی شرکت زونلای مالکیت و عملیاتی شده است و نیز PPTV با نام قبلی PPLive و PPS با نام قبلی PPStream انجام گرفته است. در حال حاضر یکی از تأمین‌کنندگان محتواهای ویدیویی براساس تقاضا در چین با بهره‌گیری از پروتکل P2P شرکت KanKan است. با بیش از 20 میلیون کاربر انحصاری که تماشای فیلم‌ها را در هر ماه شامل می‌شوند.

در سطح وسیع پخش سیال ویدیو از طریق P2P بسیار مشابه بارگیری فایل از طریق پروتکل

تندآب بیت است که در فصل ۲ به آن اشاره رفت. هنگامی که یک نظیر مایل به تماشای یک فیلم می‌باشد او ابتدا با یک رهگیر که ممکن است متمرکز یا با استفاده از DHT بر پایه نظیر به نظیر باشد، تماس برقرار می‌کند و از طریق رهگیر نظیرهای موجود در سیستم که نسخه‌ای از فایل ویدیو را در اختیار دارند شناسایی می‌نماید. این نظیر سپس تقاضای توده‌هایی از فایل ویدیو را می‌کند که به موازات دیگر نظیرهای دارنده‌ی همین فایل قرار گرفته است. این روش اندکی با رویکرد بارگیری در تندآب بیت(Bit Torrent) متفاوت است اما در هر صورت تقاضاها به‌گونه‌ی ترجیحی برای توده‌هایی صادر می‌شود که بایستی درآینده باز پخش شوند تا امکان پخش مستمر آنها ضمانت گردد.

طراحی Kankan به‌گونه‌ای است که از یک رهگیر و DHT خاص خود بهره‌گیری می‌کند تا محتوا را شناسایی نماید. حجم‌های وسیع از توده‌های فایل تصویری برای ویدیوهای محبوب از وجود ده‌ها هزار نظیر بهره‌گیری می‌کنند که به‌گونه‌ی ویژه‌ای بزرگ‌تر از عظیم‌ترین حجم امکان‌پذیر در پروتکل بیت ترنت هستند. پروتکل‌های استفاده شده Kankan به‌منظور برقراری ارتباط‌های بین نظیر و رهگیر، بین نظیر و DHT و در بین نظیرها همگی خصوصی هستند. آنچه جذابیت این فناوری را بیشتر می‌کند توزیع توده‌های ویدیویی بین نظیرها می‌باشد که شبکه‌ی Kankan با بهره‌گیری از پروتکل UDP در هر نقطه‌ی امکان‌پذیر از آن استفاده نموده است. نتیجه‌ی مستقیم آن خلق مقدار وسیعی ترافیک UDP در درون شبکه‌ی اینترنت چین می‌باشد.



تمرین‌های فصل هفتم

۱- منظور از تعامل‌گرایی برای جریان ذخیره‌سازی شده صوتی و تصویری چیست؟ این مفهوم در ارتباط با تعامل بی‌درنگ صوتی و تصویری چگونه تعریف می‌شود؟

۲- دو شکل ۷-۱ و ۷-۲ دو روش پخش رسانه‌ی ذخیره سازی شده را نشان می‌دهند چه نقاط قوت و ضعفی را برای هریک می‌توان برشمرد؟

۳- نسبت‌های متداول فشرده‌سازی برای کاربردهای تصویری و صوتی چیست؟ برمبنای مطالب بخش ۷-۱ توضیح دهید.

۴- چه تفاوتی مابین تأخیر انتها به انتها و ناپایداری بسته می‌شناسید؟ علل وقوع ناپایداری بسته چیست؟

۵- علت آن‌که بسته‌ی دریافت شده پس از زمان اختصاص یافته برای بازپخش تلف شده به‌شمار می‌آید چیست؟

۶- نقش پروتکل DNS در یک CDN چیست؟ آیا DNS به منظور پشتیبانی CDN بایستی دچار تغییرات شود؟ چه نوع اطلاعاتی را در صورت لزوم یک CDN بایستی در اختیار DNS قرار دهد.

۷- در بخش ۷-۸ سه نوع بسته‌ی RTCP توصیف شده‌اند به اختصار اطلاعات درون هریک از انواع بسته‌ها را توصیف کنید.

۸- نقش یک ثبات SIP چیست؟ تفاوت یک ثبت‌کننده‌ی SIP با عامل خانه در یک IP سیار چیست؟

۹- چگونه جریان‌های RTP گوناگون در تعامل‌های مختلف از سوی گیرنده شناسایی می‌شوند؟ چگونه جریان‌های مختلف درون یک تعامل مشابه شناسایی می‌شوند؟ بسته‌های RTP و RTCP چگونه از یکدیگر بازشناسی می‌شوند.

۱۰- چه نوع اطلاعاتی مورد نیاز برای ابعادگذاری شبکه می‌باشد تا از طریق آن QoS موردنظر قابل دستیابی باشد.

۱۱- مثالی را از صف‌هایی که شما در طول زندگی روزانه‌ی خود از FIFO، الویت، RR و WFQ تجربه نموده‌اید نام ببرید.

۱۲- چالش‌های وابسته به الگوهای Intserv و ذخیره‌سازی برمبنای جریان ذخیره‌سازی از منابع چیست؟

۱۳- با جستجو در وب دو سامانه را شناسایی نمائید که از امکانات پخش صدا و یا تصویر ذخیره‌سازی شده استفاده می‌کنند. برای هر سامانه مشخص نمائید از کدام یک از موارد ذیل استفاده می‌شود:

الف) RTSP ب) متافایل

ج) RTP د) ارسال صدا و تصویر از طریق UDP یا TCP

۱۴- آیا بافر دریافت TCP و بافر مشتری رسانه‌ی پخش کارکرد مشابه دارند؟ اگر خیر چگونه تعامل انجام می‌دهند.

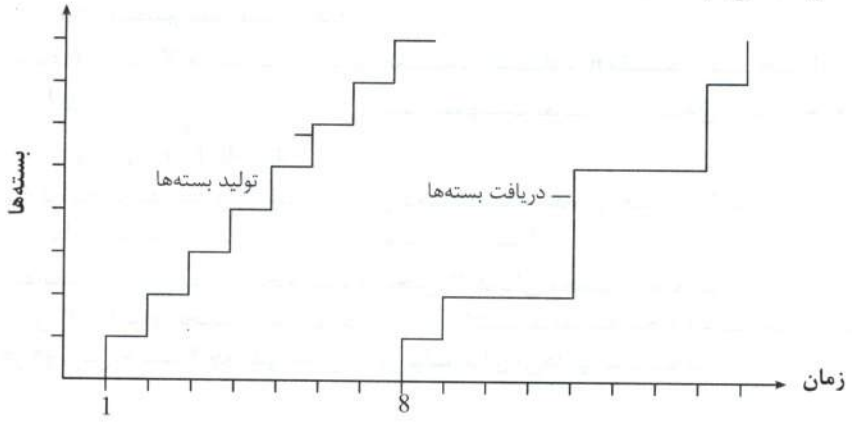
۱۵- دو بافر مشتری نشان داده شده در شکل ۷-۳ را در نظر بگیرید اگر فرض شود سیستم پخش از امکانات سومی بهره می‌گیرد به عبارت دیگر رسانه را به درون سوکت بی‌درنگ تزریق می‌کند و از پهنای باند TCP بسیار بزرگ‌تر از d در بیش‌تر مواقع باشد. چنان‌چه بافر مشتری تنها قادر به نگهداری یک سوم از رسانه باشد آن‌گاه توضیح دهید چگونه x(t) و محتوای بافر مشتری در طول زمان تغییر می‌یابند؟

۱۶- در مثال تلفن اینترنت در بخش ۷-۵ اگر h مجموع تعداد بایت‌های سرپیام باشد که بایستی به هر گروه افزوده شود و این مورد شامل سرپیام‌های TCP و UDP نیز می‌شود آن‌گاه:

الف) ارزش h هنگام استفاده از RTP چیست؟

ب) اگر فرض شود یک داده گرام IP در هر 20 ثانیه ارسال شود سرعت انتقال برای داده گرام تولیدی از سوی یک طرف این کاربرد چیست؟

۱۷- بادرنظر گرفتن شکل ذیل که مشابه شکل ۷-۵ است. یک فرستنده فایل صوتی بسته‌بندی شده‌ای را به‌صورت متناوب در زمان t=1 گسیل می‌کند. نخستین بسته در گیرنده در زمان t=8 دریافت می‌شود.



الف) بدون درنظر گرفتن تأخیرهای احتمالی بازپخش حساب کنید تأخیرهای از فرستنده به گیرنده برای بسته‌های از 2 تا 8 چه مقدار است؟ توجه داشته باشید هر قطعه خط عمودی و افقی در شکل دارای طولی معادل 1، 2 و یا 3 یکای زمان می‌باشد؟

ب) اگر بازپخش صدا بی‌درنگ پس از دریافت نخستین بسته در گیرنده در زمان t=8 آغاز گردد. کدام‌یک از نخستین 8 بسته‌ی ارسالی در زمان مطلوب خود جهت بازپخش به گیرنده وارد نخواهد شد؟

ج) اگر بازپخش صدا در زمان t=9 آغاز شود کدام‌یک از نخستین 8 بسته‌ی ارسالی در زمان مطلوب خود جهت بازپخش به گیرنده وارد نخواهد شد؟

د) حداقل تأخیر بازپخش درگیرنده که نتیجه‌ی آن ورود همه‌ی 8 بسته‌ی نخست در زمان مطلوب جهت بازپخش می‌باشد، چه‌مقدار است؟

۱۸- با در نظرگیری مجدد شکل پرسش ۱۷ که زمان‌های انتقال بسته‌های صوتی و دریافت آن‌ها را نشان می‌دهد:

الف) تأخیر پیش‌بینی شده برای بسته‌های از 2 تا 8 را با بهره‌گیری از رابطه‌ی شماره‌ی (۷-۱) برای d_i با فرض آن که ارزش u=0.1 می‌باشد، محاسبه نمائید.

ب) مطلوب است محاسبه‌ی انحراف از تأخیر برای میانگین زمان بسته‌های 2 تا 8 با بهره‌گیری از رابطه‌ی شماره‌ی (۷-۲) برای v_i با فرض آن که ارزش u=0.1 می باشد؟

۱۹- با توجه به مطالب بخش ۷-۵ به منظور محاسبه‌ی میانگین تأخیر d_i، اگر فرض شود u=0.1 و مقدار r_1-t_1 تازه‌ترین نمونه‌ی تأخیری و r_2-t_2 نمونه‌ی بعدی آن و غیره باشد آن‌گاه:

الف) برای یک کاربرد صوتی اگر 4 بسته در گیرنده‌ای با زمان‌های تأخیر نمونه‌ی r_4-t_4 ، r_3-t_3 ، r_2-t_2 و r_1-t_1 وارد شده، زمان تأخیر d را براساس چهار نمونه‌ی اشاره شده محاسبه و آن را نشان دهید.

ب) رابطه‌ی به‌دست آمده را برای تعداد n نمونه‌ی تأخیری تعریف نمائید.

ج) برای رابطه‌ای که در بخش (ب) تعریف می‌نمائید اگر مقدار n به سمت بی‌نهایت میل کند رابطه‌ی نتیجه شده را نشان دهید و نتیجه‌گیری خود را از آن شرح دهید.

۲۰- برای مثال تلفن اینترنتی بخش ۷-۵ اگر t_i برچسب زمانی بسته‌ی دریافتی i (iامین بسته) باشد و اگر r_i زمانی است که این بسته دریافت می‌شود با فرض آن که d_n میانگین تأخیر تقریبی پس از دریافت بسته‌ی n باشد و به دنبال دریافت نخستین بسته زمان تأخیر برابر $d_1 = r_1 - t_1$ تنظیم شده باشد آن‌گاه:

الف) اگر لازم باشد برای همه‌ی مقادیر n مقدار تأخیر d_n برابر $d_n = \frac{(r_1 - r_1 + r_2 - t_2 + \ldots + r_n - t_n)}{n}$ باشد مطلوبست تعریف یک رابطه‌ی خود مرجع برای d_n برمبنای r_n ، d_n-1 و t_n.

ب) توضیح دهید چرا برای تلفن اینترنتی تأخیر توصیف شده در بخش ۷-۵ مناسب‌تر از تأخیر نشان داده شده در بخش (الف) این پرسش می‌باشد؟

۲۱- مقایسه‌ای را بین مراحل انجام شده در پخش ۷-۵ برای محاسبه‌ی میانگین تأخیر با آن‌چه در بخش ۳-۱۱ برای تخمین زمان دوره‌ی رفت و برگشت تعریف شد انجام دهید. نقاط مشترک هر دو روش چیست؟ چه تفاوت‌هایی را می‌توانید برای آن‌ها فهرست نمائید؟

۲۲- با توجه به این نکته که CDN باعث افزایش ظرفیت پیوند در یک شبکه نمی‌شود. با فرض آن‌که CDN از پیوندهای موجود برای توزیع محتوا در بین گره‌های CDN استفاده می‌کند چگونه CDN کارایی نشان داده شده از سوی میزبان‌ها را بهینه می‌کند؟ نمونه‌ای را در توضیحات خود بیان نمائید.

۲۳- تشابه RTSP و HTTP در چیست؟ آیا RTSP نیز دارای روش‌هایی می‌باشد؟ آیا از HTTP می‌توان به منظور تقاضای یک جریان (پخش) استفاده نمود؟

۲۴- تفاوت RTSP و HTTP در چیست؟ آیا HTTP درون باند است یا خارج از باند؟ آیا RTSP اطلاعات وضعیت یک مشتری را نگهداری می‌کند؟ پاسخ خود برای حالت توقف و سپس پخش ارایه دهید.

۲۵- با یادآوری روش‌های FEC در تلفن اینترنتی بخش ۷-۵ اگر فرض شود نخستین روش به تولید یک گروه افزونه برای هر 4 گروه اصلی بپردازد و اگر روش دوم از نرخ پائین سرعت برای رمزگذاری بهره‌گیری نماید که سرعت انتقال آن 25 درصد سرعت انتقال نامی جریان باشد آن‌گاه:

الف) چه مقدار پهنای باند اضافی برای هر روش لازم می‌باشد؟ مقدار زمان تأخیر بازپخش اضافه شده برای هر روش چه مقدار است؟

ب) عملکرد دو روش چگونه است اگر در هر گروه 5 بسته‌ای نخستین بسته‌ی آن تلف شود؟ کدام روش کیفیت صوتی بهتری دارد؟

ج) اگر در هر گروه 2 بسته‌ای نخستین بسته تلف شود کدام یک از دو روش بهینه عمل می‌کند و کدام‌یک کیفیت صوتی بهتری را در این حالت دارد؟

۲۶- یک تعامل RTP با تعداد 4 کاربر را در نظر بگیرید. اگر همگی آن‌ها ارسال و دریافت بسته‌های RTP به درون آدرس چندپخش مشابهی را انجام دهند و هر کاربر تصاویر را با سرعت 100kbps ارسال دارد، آن‌گاه:

الف) آیا RTCP ترافیک خود را محدود می‌کند؟ سرعت محدودیت ترافیک در این حالت چه‌قدر است؟

ب) به یک گیرنده چه مقداری از پهنای باند RTCP اختصاص می‌یابد؟

ج) به یک فرستنده چه مقداری از پهنای باند RTCP اختصاص پیدا می‌کند؟

۲۷- با توجه به شکل ذیل به هریک از پرسش‌های ذیل پاسخ دهید:

الف) با فرض به‌کارگیری FIFO زمان خروج از صف را برای هر بسته‌ی از 2 تا 12 مشخص نمائید. برای هر بسته تأخیر بین ورود و ابتدای شیار انتقال آن چه‌قدر است؟ میانگین این تأخیر برای همه‌ی 12 بسته چه مقدار است؟

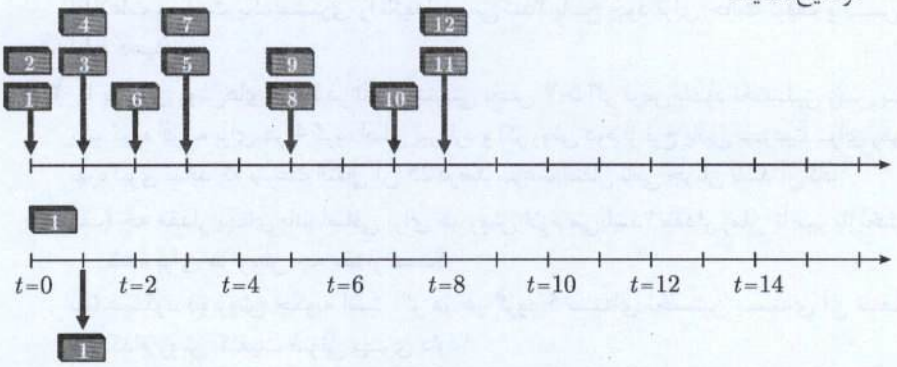
ب) اگر خدمات الویت‌دار فرض شود و بسته‌های با شماره‌ی فرد دارای الویت بالا و بسته‌های با شماره‌ی زوج الویت پائین داشته باشند آن‌گاه زمان خروج از صف برای هر بسته‌ی از 2 تا 12 چه‌قدر است؟ برای هر بسته در این حالت تأخیر بین ورود و ابتدای شیار انتقال چه‌قدر است؟ میانگین تأخیر برای همه‌ی 12 بسته چه مقدار است؟

ج) اگر خدمات گردش نوبتی درنظر گرفته شود و بسته‌های 1، 2، 3، 6، 11 و 12 از گروه یک باشند و بسته‌های 4، 5، 7، 8، 9 و 10 از گروه 2 حساب کنید زمان خروج از صف برای هر

بسته‌ی از 2 تا 12 در این حالت چیست؟ برای هر بسته تأخیر بین ورود و خروج آن چه مقدار است؟ میانگین تأخیر برای همه‌ی 12 بسته چه مقدار است؟

د) اگر خدمات از نوع WFQ فرض شود و بسته‌های شماره‌ی فرد متعلق به گروه یک و بسته‌های با شماره‌ی زوج متعلق به گروه 2 باشند و اگر گروه یک دارای WFQ با وزن 2 و گروه 2 دارای WFQ با وزن یک باشد، آن‌گاه برای هر بسته تأخیر بین ورود و خروج آن چه مقدار است؟ میانگین تأخیر برای همه‌ی 12 بسته چه مقدار است؟

هـ) مشاهدات خود را در مورد میانگین تأخیر هر یک از 4 مورد FCFS، RR، الویت و WFQ توضیح دهید.



۲۸- یک جریان بسته وقتی مطلوب مشخصات دلو نشت کننده‌ی (r ، b) عمل می‌کند که سرعت ریزش b و میانگین نرخ r در آن به صورتی باشد که تعداد بسته‌های وارد شده به ظرف نشت کننده در هر فاصله‌ی زمانی t برای همه‌ی حالت‌های t کوچک‌تر از rt+b باشد. اگر یک جریان بسته می‌تواند مطابق سطل نشت کننده عمل نماید در صورتی‌که در انتظار یک ناظر کنترل دلو نشت کننده با پارامترهای r و b می‌باشد؟ پاسخ خود را توجیه کنید.

۲۹- فرض کنید یک سیستم کنترل WFQ به یک بافر که سه گروه را پشتیبانی می‌کند، اعمال شده است. اگر اوزان اختصاص یافته به این سه گروه به ترتیب 0.5 ، 0.25 و 0.25 باشد آن‌گاه:

الف) فرض کنید هر گروه دارای تعداد انبوهی بسته در بافر است ترتیب خدمات رسانی به این سه گروه چگونه بایستی انجام شود تا اوزان WFQ عنوان شده قابل دستیابی باشند. بدیهی است برای روش گردش نوبتی ترتیب طبیعی سرویس‌دهی به‌صورت: ...123123123 می‌باشد.

ب) اگر فرض شود کلاس‌های 1 و 2 حامل تعداد انبوهی بسته در بافر باشند و گروه 3 حامل هیچ بسته‌ای نباشد آن‌گاه ترتیب سرویس‌دهی سه کلاس چگونه بایستی انجام شود تا اوزان WFQ اشاره شده در پرسش قابل دستیابی باشند؟

۳۰- نشان دهید تا زمانی که $r_i < \frac{Rw_1}{\Sigma w_j}$ باشد آن‌گاه dmax قطعاً شامل حداکثر تأخیری خواهد شد که هر بسته در جریان 1 در یک صف WFQ همواره تجربه خواهد نمود.

۳۱- به نظر شما آیا بهتر است پخش ذخیره شده‌ی صدا و تصویر بر فراز کدام‌یک TCP و یا UDP انجام شود؟

۳۲- آیا چالش تأمین QoS را می‌توان از طریق پهنای باند کافی برطرف نمود؟ منظور از پهنای باند کافی ارتقاء ظرفیت همه‌ی پیوندها به‌گونه‌ای می‌باشد که محدودیت سرعت سبب نگرانی نباشد.



فصل ۸

امنیت در شبکه‌های کامپیوتری

قبلاً در زیربخش ۱۷ از فصل یک برخی از متداول‌ترین تهاجم‌های اینترنتی، شامل حمله‌های بدافزاری، انسداد خدمات، جاسوسی، به خطا انداختن منابع اطلاعاتی و اعمال تغییرات در پیام‌ها و نیز حذف احتمالی آن‌ها معرفی شدند. اگرچه از آن مقطع تاکنون مطالب و موضوعات قابل ملاحظه‌ای را در ارتباط با شبکه‌های کامپیوتری آموخته‌ایم، اما هنوز چگونگی ایمن‌سازی شبکه از این تهاجمات بررسی و تحلیل عمیق نشده‌اند. با توجه به تجربیات به‌دست آمده در ارتباط با پروتکل‌های شبکه‌های کامپیوتری و ا ینترنت اکنون در موقعیتی قرار گرفته‌ایم که قادر به مطالعه‌ی عمیق موارد مربوط به ارتباطات ایمن و به‌ویژه حفاظت شبکه‌های کامپیوتری در برابر نفوذ مهاجمان می‌باشیم.

به‌منظور تحقق این مهم از دو فرد Alice,Bob که در فصل‌های مختلف معرفی شده‌اند برای برقراری تعامل‌های ایمن و حراست شده بهره می‌گیریم. بدیهی است اگرچه در طول مطالب فصل‌ها از Alice,Bob به عنوان دو شخص، نامبرده شده است اما در عمل این دو می‌توانند دو مسیریاب به‌منظور مبادله‌ی ایمن اطلاعات جداول مسیریابی خود باشند و یا آن‌که ممکن است یک برنامه‌ی مشتری که مایل به برقراری یک ارتباط انتقال اطلاعات ایمن با یک برنامه‌ی سرور است، باشند. علاوه بر این موارد این دو ممکن است دو کاربرد پست الکترونیکی (ایمیل) باشند که مایل به تبادل ایمن مطالب خویش می‌باشند. در ادامه‌ی مطالب از این دو گاهی به اختصار با نام A و B نام برده می‌شود.

در مطالب اشاره شده در این پیش گفتار تمایل A و B به برقراری یک ارتباط ایمن مورد تاکید قرار گرفت در صورتی‌که مفهوم ارتباط ایمن مشخص نگردید. برقراری یک تعامل حفاظت شده در شبکه می‌تواند شامل موارد متعددی باشد. از جمله‌ی این موارد می‌توان به عناوین ذیل اشاره نمود:

- پنهان‌سازی اطلاعات مبادله شده بین A و B از نفر ثالث.
- آشکارسازی و شناسایی نفر ثالث از سوی A و B هنگامی که آن‌ها مشغول مبادله‌ی اطلاعات می‌باشند.

در بخش نخست این فصل روش‌های بنیادی مربوط به چگونگی رمزگذاری اطلاعات به‌منظور فراهم آوری یک ارتباط رمزگذاری شده و اعتبارسنجی طرف ارتباط به‌منظور اطمینان از یکپارچه ماندن اطلاعات مبادله شده تحلیل می‌شود.

در بخش دوم فصل اصول بنیادی رمزگذاری به‌منظور طراحی پروتکل‌های ایمن‌سازی شبکه تحلیل می‌شود و بار دیگر در ارتباط با موضوعات طراحی پروتکل‌های ایمن‌سازی شبکه از رویکرد فراز به فرود بهره گرفته می‌شود. پروتکل‌های ایمن‌سازی در هریک از چهار لایه‌ی بالاتر شبکه تحلیل خواهند شد که آغاز آن از لایه‌ی کاربرد می‌باشد. چگونگی ایمن‌سازی تبادل ایمیل، ارتباط‌های TCP و نیز اعمال پوشش امنیتی در لایه‌ی شبکه و تأمین امنیت در یک شبکه‌ی LAN بی‌سیم از جمله‌ی این موارد به شمار می‌آیند. در بخش سوم فصل اعمال مسایل امنیتی که عبارت است از حفاظت شبکه‌های سازمانی در برابر تهاجمات تحلیل می‌شوند. نگاه عمیق و تحلیلی بر سیستم‌های فایروال و آشکارسازی نفوذ در شبکه‌های سازمانی از جمله‌ی موضوعات با اهمیت این بخش را تشکیل خواهد داد.

۸-۱ امنیت شبکه چیست؟

مطالعه پیرامون امنیت شبکه را با مثال دو فرد A و B که بارها در این کتاب از آن‌ها نام برده شده است آغاز می‌کنیم. اگر Alice و Bob مایل به برقراری یک تعامل و ارتباط امن مابین خود باشند آن‌گاه این پرسش مطرح می‌شود که منظور از ارتباط امن چیست؟ قطعاً A مایل است فقط B قادر به درک پیام‌های ارسالی از سوی وی باشد اگرچه آن‌ها ارتباط مابین خویش را از طریق یک رسانه‌ی تهی از امنیت انجام می‌دهند که ممکن است فرد ثالث دیگری به شنود تعامل بین A و B بپردازد. از طرف دیگر B نیز مایل است از این موضوع اطمینان حاصل نماید که پیام‌های دریافتی از جانب او قطعاً به وسیله‌ی A گسیل شده است و بازهم Alice علاقه‌مند است که مطمئن از ارتباط با Bob باشد به عبارت دیگر فرد بیگانه‌ای خود را در جایگاه B قرار ندهد. از همه مهم‌تر A و B هر دو بسیار علاقه‌مند می‌باشند که محتوای پیام‌های آن‌ها در طول مسیر انتقالی دستخوش تغییر و یا جعل نگردد و سرانجام هر دو مایل به حصول اطمینان از این موضوع می‌باشند که همواره قادر به برقراری ارتباط با یکدیگر خواهند بود به عبارت دیگر مانعی جهت جلوگیری از تعامل آن‌ها با یکدیگر نخواهد بود. با توجه به همه‌ی موارد اشاره شده ویژگی‌های یک ارتباط مطلوب و ایمن را می‌توان چنین خلاصه نمود:

- **محرمانگی.** تنها فرستنده و گیرنده‌ی موردنظر بایستی قادر به درک محتوای پیام ارسال شده باشند. از آن‌جا که نامحرمان در مسیر ارتباط ممکن است قادر به شنود موارد حامل پیام باشند بنابراین لازم است که پیام به‌گونه‌ای **رمزگذاری**[1] شود که در صورت شنود پیام از سوی یک نامحرم محتوای پیام برای وی قابل درک نباشد. این ویژگی محرمانه ماندن پیام شاید متداول‌ترین مفهوم از نقطه‌نظر عبارت **از تباط ایمن** به‌شمار می‌آید. موضوعات مربوط به روش‌های رمزگذاری و رمزگشایی در زیربخش ۸-۲ همین فصل بررسی شده‌اند.
- **اعتبارسنجی نقطه‌ی انتهایی.** فرستنده و گیرنده هر دو بایستی قادر به شناسایی و اعتبارسنجی طرف دیگر ارتباط باشند. ارتباط رودررو بین افراد این دشواری را به سادگی از طریق شناسایی به‌وسیله‌ی روش رؤیت نمودن، برطرف می‌سازد اما در مواقعی که تبادل

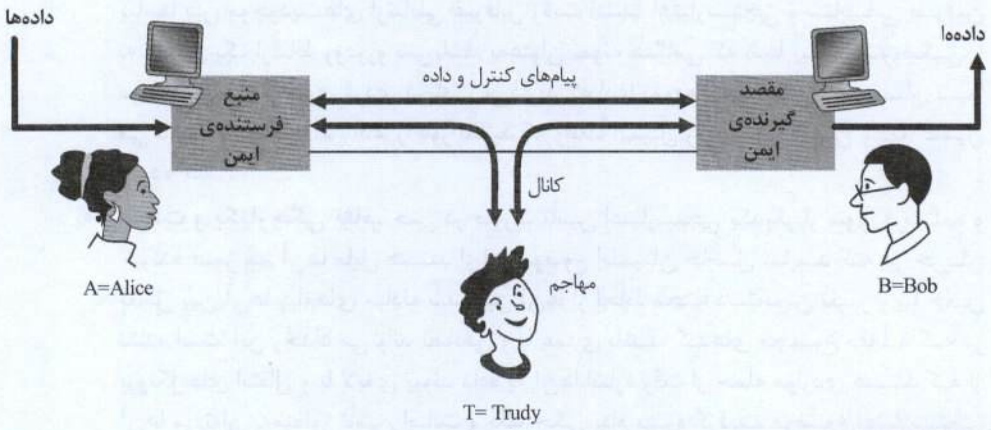
1- Encryption

پیام‌ها بین موجودیت‌های ارتباطی غیرقابل رؤیت است اعتبارسنجی و شناسایی طرفین به‌سادگی یک ارتباط رودررو نمی‌باشد. به‌عنوان نمونه هنگامی که شما پیام الکترونیکی را در قالب ایمیل از سوی فردی دریافت می‌دارید که ادعا نموده است، یکی از دوستان شما می باشد آیا بایستی این ادعا را باور نمائید که واقعاً ایمیل را دوست واقعی شما گسیل نموده است؟

- **اصالت و یکپارچگی پیام**. حتی در صورت تأمین اعتبارسنجی یکدیگر از سوی فرستنده و گیرنده هنوز هم آن‌ها مایل هستند از این موضوع اطمینان حاصل نمایند که در جریان تعامل بین آن‌ها پیام‌های مبادله شده بین آن‌ها از لحاظ محتوا دستخوش تغییر و یا جعل نشده است. این رخداد می‌تواند تصادفی و یا عمدی باشد. کدهای مجموع مقابله که در پروتکل‌های انتقال و یا لایه‌ی پیوند داده به آن‌ها اشاره رفت از جمله مواردی هستند که از آن‌ها می‌توان به‌منظور تأمین اصالت و یکپارچگی پیام بهره گرفت. موضوع اعتبارسنجی نقطه‌ی انتهایی و یکپارچگی پیام در بخش ۸-۳ مورد مطالعه قرار می‌گیرد.
- **امنیت عملیات**. تقریباً همه‌ی نهادها مانند شرکت‌ها، دانشگاه‌ها و نظایر آن‌ها دارای شبکه‌هایی می‌باشند که به اینترنت عمومی متصل شده‌اند. این شبکه‌ها به‌سادگی می‌توانند از سوی مهاجمان بالقوه از طریق دستیابی به اینترنت عمومی در معرض تهدید قرار گیرند. مهاجمان از طریق جایگذاری کرم‌های مخرب در میزبان‌های شبکه به تمامی اطلاعات محرمانه‌ی یک شرکت می‌توانند دسترسی داشته باشند و یا پیکربندی درونی آن شبکه را نگاشت نمایند و یا آن‌که حملات از نوع DoS را انجام دهند. در بخش‌های انتهایی همین فصل ملاحظه خواهد شد چگونه تجهیزات عملیاتی مانند دیواره‌های آتش و سیستم‌های تشخیص مهاجم به‌منظور جلوگیری از تهاجم به شبکه‌ی یک شرکت مورد بهره‌برداری قرار می‌گیرد. یک فایروال معمولاً بین شبکه‌ی شرکت و یا یک نهاد و اینترنت عمومی قرار می‌گیرد و از این طریق دستیابی بسته‌ها را از سوی یک شبکه و یا به طرف آن کنترل می‌کند. یک سیستم تشخیص مهاجم از طرف دیگر به بازرسی عمیق درونی بسته می‌پردازد و مدیر شبکه را در ارتباط با فعالیت‌های مشکوک باخبر می‌سازد.

اکنون که مفهوم امنیت شبکه آشکار شد به بررسی نوع اطلاعاتی که یک مهاجم و یا نامحرم ممکن است دستیابی نماید و نیز عملیاتی که از سوی وی ممکن است انجام گیرد، می‌پردازیم. شکل ۸-۱ سناریویی را نشان می دهد که در آن A به عنوان فرستنده مایل به ارسال پیامی به B می‌باشد. به‌منظور تبادل ایمن پیام‌ها با توجه به سه مورد اساسی محرمانگی، اعتبارسنجی نقطه‌ی انتهایی و یکپارچگی پیام، A و B هر دو به تبادل پیام‌های کنترل و داده می‌پردازند. این تبادل اطلاعاتی بسیار شبیه به مبادله‌ی قطعه‌های کنترل و داده بین فرستنده‌ها و گیرنده‌های TCP می‌باشد. همگی و یا برخی از این پیام‌ها رمزگذاری می‌شوند. همان‌گونه که در زیربخش ۱-۱۷ از فصل یک عنوان گردید یک مهاجم می‌تواند عملیات ذیل را انجام دهد:

- **شنود**. استشمام و ثبت پیام‌های کنترل و داده‌ی یک کانال.
- تغییر، جعل، حذف و یا اضافه نمودن محتوای پیام‌ها.



شکل ۸-۱ ♦ فرستنده، گیرنده و مهاجم به ترتیب Alice، Bob، T-Trudy

در ادامه ملاحظه خواهد شد در صورت فقدان تدابیر دفاعی مناسب این توان‌مندی‌ها سبب خواهد شد، یک مهاجم قادر به انجام انواع گسترده‌ای از تهدیدهای امنیتی به شبکه باشد. از جمله‌ی مهم‌ترین آن‌ها عبارتند از:

- جاسوسی برروی یک ارتباط جهت سرقت رمزهای عبور اینترنتی و داده‌ها
- تقلید و ایفای نقش یک موجودیت جاری در شبکه
- به گروگان گرفتن یک تعامل در جریان
- ممانعت از دسترسی کاربران مشروع شبکه به خدمات آن از طریق سرریزی منابع شبکه و غیره.

گزارش تهاجم‌های موفق انجام شده تا سال 2012 میلادی در کتاب انتشار یافته در این رابطه از سوی مرکز هماهنگی CERT در قالب مستند [CERT 2012] موجود می‌باشد.

با توجه به تهدیدهای واقعی رایج در اینترنت انواع تعامل های بین A و B که نیازمند تعامل ایمن اینترنتی می‌باشند چه مواردی است؟ قطعا A و B دو کاربر انسانی می‌باشند که در محل دو سیستم انتهایی قرار گرفته‌اند به عنوان مثال A و B واقعی علاوه بر آن که مایل به تبادل ایمن ایمیل بین خود می‌باشند ممکن است در حال انجام عملیات تبادل تجاری الکترونیکی نیز باشند به عنوان مثال B ممکن است مایل به انتقال ایمن و محرمانه‌ی اطلاعات کارت اعتبار بانکی خویش به یک وب سرور به‌منظور انجام عملیاتی مانند خرید آنلاین و تعامل با حساب بانکی خویش باشد، از طرف دیگر عوامل نیازمند ارتباط ایمن ممکن است خودبخشی از زیرساخت شبکه باشند. در فصل ۲ به DNS اشاره گردید و یا سیستم‌های عظیم مسیریابی که در حال تبادل اطلاعات مسیریابی هستند و نیازمند ارتباط محرمانه و ایمن بین یکدیگر می‌باشند. کاربردهای مدیریت شبکه از جمله موارد دیگری است که در فصل ۹ بررسی شده‌اند و یک مهاجم از طریق نفوذ به آن‌ها می‌تواند جداول دستیابی DNS و اطلاعات آن‌ها را و یا محاسبات انجام مسیریابی و نیز عملیات مدیریت شبکه را در آن‌ها مورد تهدید قرار داده و فاجعه‌ای باور نکردنی را برروی شبکه‌ی اینترنت سبب شود.

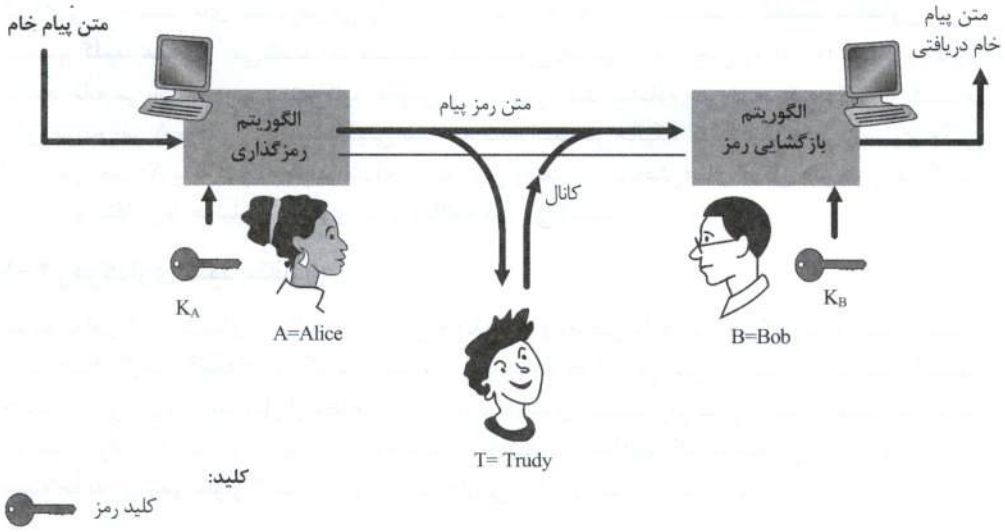
اکنون که چارچوب امنیت شبکه مشخص گردید زمان آن رسیده است که موضوع رمزگذاری و رمزگشایی عمیقاً تحلیل گردد. اگرچه بهره‌گیری از رمزگذاری در تأمین محرمانگی آشکار است اما در

ادامه ملاحظه خواهد شد که این رویکرد به‌منظور تأمین اعتبارسنجی نقاط انتهایی و اصالت پیام نیز نقش حیاتی را بازی می‌کند و به عبارت دیگر رمزگذاری سنگ بنای امنیت شبکه می‌باشد.

۸-۲ اصول رمزگذاری

اگرچه پیشینه‌ی تاریخی اصول رمزگذاری به دوران امپراطوری جولیوس سزار قیصر روم مربوط می‌شود اما روش‌های نوین رمزگذاری شامل بسیاری از موارد که در اینترنت فعلی از آن استفاده می‌شود ریشه در سی سال گذشته دارد. اثر معروف **کان**[1] با عنوان رمزشکن‌ها و علوم اسرار از مصر باستان تا رمزگذاری کوآنتمی اثر **سینگ**[2] قدمت طولانی اصول رمزگذاری را آشکار می‌سازند. آگاهی کامل از اصول و روش‌های رمزگذاری خود نیازمند یک کتاب کامل است بنابراین در این کتاب اصول بنیادی رمزگذاری به‌ویژه آن‌گونه که در اینترنت مورد بهره‌برداری قرار می‌گیرند بررسی و تحلیل خواهند شد. اگرچه نقطه‌ی تمرکز ما در این بخش بهره‌گیری از رمزگذاری به‌منظور محرمانه ماندن اطلاعات است اما روش‌های به کار گرفته شده برای این منظور بنیان‌های اعتبارسنجی، اصالت پیام، مشروعیت و نیز موارد دیگر را نیز در درون خود به ناگزیر تنیده است.

روش‌های رمزگذاری این امکان را فراهم می‌آورد که یک فرستنده داده‌های خود را از چشم یک نامحرم که ممکن است آن داده‌ها را شنود نماید، پنهان کند. بدیهی است گیرنده‌ی اطلاعات نیز بایستی از امکانات لازم به‌منظور بازیابی داده‌های اصلی از اطلاعات پنهان شده برخوردار باشد. شکل ۸-۲ برخی از نمادها و واژگان مهم رمزگذاری را تعریف می‌نماید.



شکل ۸-۲ ♦ عناصر رمزگذاری

اگر فرض شود Alice مایل به ارسال پیامی به Bob می‌باشد آن‌گاه پیام اصلی A در قالب بیان شده

1- Kahn's book
2- Singh's book

اصطلاحاً **متن ساده یا خام**[1] و به بیان دیگر **متن شفاف**[2] نامیده می‌شود. با بهره‌گیری از **الگوریتم رمزگذاری**[3] Alice می‌تواند پیام خام خود را رمز نماید. نتیجه‌ی حاصل از این عملیات **متن رمز شده**[4] می‌باشد که از نقطه‌نظر یک مهاجم و یا سارق اطلاعات غیرقابل درک و مفهوم می‌باشد. جالب توجه آن است، در روش‌های نوین رمزگذاری که اینترنت نیز از آن‌ها بهره گرفته است اصول و روش‌های رمزگذاری داده‌ها کاملاً شناخته شده و در دسترس عموم حتی مهاجمان و سارقان اطلاعات نیز می‌باشد. مستندهای [RFC 2420, RFC 2437, RFC 1321] و نظایر آن‌ها از جمله‌ی این موارد به شمار می‌آیند. بدیهی است هنگامی که روش‌های رمزگذاری داده‌ها برای همگان آشکار باشد آن‌گاه برخی از اطلاعات محرمانه بایستی وجود داشته باشد که مانع رمزگشایی پیام‌های ارسالی به‌وسیله‌ی مهاجمان گردد. در این مقطع است که به‌کارگیری **کلیدهای** رمز طرح می‌شود.

در شکل ۸-۲ فرستنده‌ی پیام یعنی A یک **کلید رمز** K_A را که عبارت است از یک رشته اعداد و یا حروف به‌عنوان ورودی به الگوریتم رمزگذاری معرفی می‌نماید. الگوریتم رمزگذاری با بهره‌گیری از کلید K_A و متن ساده‌ی پیام (m) به عنوان اطلاعات ورودی، یک متن رمز شده‌ای را به عنوان خروجی تولید می‌نماید. عبارت $K_A(m)$ اشاره به متن رمزگذاری شده با بهره‌گیری از کلید K_A برای پیام متن خام m دارد. الگوریتم رمزگذاری مورد استفاده که از کلید بهره گرفته است از طریق چارچوب متن قابل درک می‌باشد. به گونه‌ی مشابه B نیز با تأمین کلید رمز برای **الگوریتم بازگشایی رمز**[5] که متن رمزگذاری شده‌ی $K_A(m)$ و کلید K_B را به عنوان ورودی دریافت می‌دارد سبب می‌شود که آن الگوریتم مجدداً متن خام ارسالی را به عنوان خروجی بازیابی نماید. به عبارت دیگر اگر B پیام رمز شده‌ی $K_A(m)$ را دریافت نماید آن را از طریق محاسبه‌ی $K_B(K_A(m))=m$ رمزگشایی می‌کند. از انواع گوناگونی از سیستم‌های کلید رمز می‌توان بهره گرفت از جمله‌ی این سیستم‌ها **کلید متقارن**[6] و یا سیستم **کلید عمومی**[7] می‌باشند. در سیستم کلید متقارن هر دو کلید متعلق به A و B مشابه یکدیگر اما محرمانه می‌باشند. در سیستم کلید عمومی از یک زوج کلید استفاده می‌شود که یکی از کلیدها برای هر دو نفر A و B و حتی تمام گیتی شناخته شده است در حالیکه کلید دیگر فقط از سوی یکی از دو نفر یعنی A و یا B و نه هر دو شناخته شده می باشد. در زیربخش‌های ذیل هر دو روش کلید عمومی و متقارن با جزئیات بیش‌تری مورد مطالعه قرار می‌گیرند.

۸-۳ رمزگذاری کلید متقارن

تقریباً تمامی الگوریتم‌های رمزگذاری از نوعی جایگذاری و تعویض داده بهره می‌گیرند. به عنوان مثال با در اختیار گرفتن قطعه‌ای از یک متن ساده به محاسبه و جایگذاری متن رمز شده‌ی مناسب به‌منظور جایگزینی آن می‌پردازند. قبل از مطالعه‌ی یک کلید برمبنای سیستم رمزگذاری نوین ابتدا به عصر جولیس سزار بازگشت می‌کنیم تا با بهره‌گیری از الگوریتم ساده‌ی کلید متقارن در آن دوران که اصطلاحاً به آن **رمز سزار** گفته می‌شود درک مطلوبی را از رمزگذاری کلید متقارن به‌دست آوریم.

1- Plain text
2- Clear text
3- Encryption algorithm
4- Cipher text
5- Decryption algorithm
6- Symmetric key systems
7- Public key systems

برای متون انگلیسی زبان رمز سزار از طریق گزینش هر حرف از متن خام پیام و جایگزین نمودن آن با حرفی که به تعداد k حرف از آن فاصله دارد به صورت یک حلقه‌ی بازگشتی استفاده می‌کند. مفهوم حلقه‌ی بازگشتی به این معنی است که به دنبال حرف z حرف a ظاهر می‌شود و این ترتیب مرتباً برمبنای ارزش k تکرارپذیر است. به عنوان مثال اگر k=3 فرض شود آن‌گاه حرف a در یک متن خام براساس رمز سزار تبدیل به حرف رمز d می‌شود. حرفی مانند b در متن ساده‌ی پیام تبدیل به حرف رمز e و غیره خواهد شد. در این نوع رمزگذاری ارزش k به عنوان کلید مورد استفاده قرار می‌گیرد. به عنوان مثال جمله‌ی انگلیسی ذیل را در نظر بگیرید که کلید رمز متقارن آن k=3 در حروف انگلیسی باشد:

Bob. I like to call you. alice

متن رمز شده‌ی این عبارت با کلید k=3 در قالب ذیل تعریف می‌شود:

ere, l olnh wr fdoo brx.dolfh

اگرچه متن رمز شده به نظر غیرقالب تفسیر می‌باشد اما اگر برای مهاجم مشخص باشد که از روش رمزگذاری سزار استفاده شده است در زمان کوتاهی می‌تواند متن رمز شده را بازگشایی نماید زیرا در بین 26 حرف زبان انگلیسی فقط تعداد 25 ارزش برای کلید می‌توان تعریف نمود.

سیستم رمزگذاری سزار را می‌توان با بهره‌گیری از آن‌چه اصطلاحاً **رمز تک الفبا**[1] نامیده شده است بهینه نمود. در این روش هر حرف موجود در حروف الفبا با حرف دیگری در همان مجموعه جایگزین می‌شود. روش جایگزینی حروف به جای بهره‌گیری از یک الگوی معمولی مانند جایگزینی با حرف از طریق ارزش کلید k برای هر حرف با هر حرف دیگری قابل جایگزینی است مشروط بر آن‌که هر حرف یک جایگزین منحصربه فرد خود و برعکس آن را در اختیار داشته باشد. قانون جایگزینی نشان داده شده در شکل ۸-۳ نمونه‌ای از یک گونه رمزگذاری متن خام پیام می‌تواند به‌شمار آید.

حروف خام :	a b c d e f g h i j k l m n o p q r s t u v w x y z
حروف رمز :	m n b v c x z a e d f g h j k i p o l u y b r e w q

شکل ۸-۳ ♦ سیستم تک الفبا

متن خام پیام مثال اشاره شده‌ی قبلی اکنون در سیستم رمزگذاری شکل ۸-۳ به‌صورت ذیل تعریف می‌شود:

nkn, s asfc uk bmgg wky.mgsbc

ملاحظه می‌شود این سیستم رمزگذاری نیز به نوعی غیرعادی می‌باشد. رمزگذاری تک الفبا از طرف دیگر به مراتب در مقابل سیستم رمز سزار بهینه است زیرا در این سیستم یک مهاجم جهت شکستن و بازگشایی رمزهای تک الفبا با تعداد !26 زوج حرف ممکن در برابر تعداد 25 زوج روبه‌رو می‌باشد. این به مفهوم آن است که تعداد 10^{26} ترکیب مختلف در این سیستم موجود می‌باشد. به عنوان مثال اگر از روش جستجوی بهره‌وری نابخردانه (BFS) به‌منظور تلاش برای یافتن همه‌ی 10^{26} زوج حرف ممکن استفاده شود نیازمند زمان قابل توجهی جهت شکستن رمز می‌باشد. بدیهی است با بهره‌گیری از دانش

1- Mono alphabetic cipher

تحلیل آماری زبان متن خام یا ساده‌ی انگلیسی می‌توان دریافت حروفی مانند e و t در یک متن انگلیسی بارها تکرار می‌شوند. از لحاظ آماری این دو حرف شامل احتمال 13 درصد در یک متن پیام و 9 درصد از نقطه‌نظر تکرار در همان متن پیام می‌باشند. علاوه بر این موارد آگاهی از این موضوع که در اغلب موارد کلمه‌های دو یا سه حرفی مانند "in" ، "it" ، "the" ، "ion" ، "ing" و غیره نیز در اغلب موارد بارها تکرار می‌شوند آن‌گاه رمزگشایی این سیستم رمز نیز نسبتاً به آسانی امکان‌پذیر می‌شود. اگر به عنوان مثال مهاجم آگاهی از محتوای پیام داشته باشد آن‌گاه رمزگشایی متن بسیار ساده خواهد بود. به عنوان مثال اگر شخص T از نوع روابط تعاملی بین A و B آگاه باشد آن‌گاه او ممکن است به حضور نام این دو فرد در متن مشکوک گردد و اگر T اطمینان حاصل نماید که این دو نام در متن رمز وجوددارند و نسخه‌ای از آن متن را در اختیار داشته باشد و اسامی هریک از این دو در قالب چهار و سه حرف مختلف از حروف انگلیسی باشد آن‌گاه وی می‌تواند مجموعاً تعداد 7 زوج حرف را از میان 26 زوج حذف نماید و در نتیجه تعداد جستجوی لازم را از 10^{26} به 10^{9} تقلیل دهد که پیشرفت قابل توجهی از لحاظ توان شکستن متن رمز می‌باشد. بدیهی است این تعداد جستجو نیز اگر T از نوع و طبیعت تعامل بین Alice و Bob آگاه باشد بازهم کم‌تر می‌شود زیرا وی به برخی از واژه‌هایی که انتظار تکرار و یا حضور آن‌ها را در متن رمز دارد پی خواهد برد.

هنگامی که سهولت بازگشایی متن رمز شده‌ی A و B برای T درنظر گرفته شود آن‌گاه سه سناریوی گوناگون را براساس نوع اطلاعاتی که مهاجم در اختیار دارد می‌توان شناسایی نمود:

۱- **تهاجم متن رمز**. در برخی موارد مهاجم ممکن است فقط دستیابی به متن رمز شنود شده داشته باشد و اطلاعات ویژه‌ای در مورد محتوای متن ساده‌ی پیام نداشته باشد. در بالا ملاحظه شد چگونه تحلیل آماری می‌تواند به رمزگشایی در یک **تهاجم متن رمز شده**[1] کمک نماید.

۲- **تهاجم متن خام شناخته شده**. قبلاً ملاحظه گردید اگر مهاجم T به نوعی از ظاهر شدن کامل نام A و B در متن رمز شده آگاهی داشته باشد آن‌گاه او می‌تواند زوج حروف متن رمز یا متن ساده برای حروفی مانند a ، l ، i ، c ، e ، b و o را به عنوان مثال برای دو نام Alice و Bob به‌دست آورد. حال اگر T (مهاجم) توانسته باشد نسخه‌ای از متن رمزشده‌ی انتقال یافته بین Alice و Bob را به‌دست آورده باشد و سپس نسخه‌ی رمزگشایی شده‌ی Bob را هم به‌نوعی در اختیار داشته باشد آن‌گاه به‌دست آوردن اطلاعات مبادله شده بین Alice و Bob برای وی به‌سادگی قابل بازگشایی و شناسایی می‌باشد. هنگامی که مهاجم دسترسی به برخی از زوج حروف متون ساده و یا رمز شده داشته باشد آن‌گاه تهدید انجام شده به شبکه را اصطلاحاً **تهاجم متن خام شناخته شده**[2] در سیستم رمزگذاری می‌نامند.

۳- **تهاجم متن ساده‌ی گزینش شده**. هنگامی که مهاجم قادر به گزینش پیام متن خام و دستیابی به الگوی متن رمز شده‌ی مربوط به آن باشد این نوع تهدید را اصطلاحاً **تهاجم متن ساده‌ی گزینش شده**[3] می‌نامند. برای الگوریتم‌های ساده‌ای که تاکنون ملاحظه شده است اگر T به نوعی A را مجبور به گسیل پیامی به صورت ذیل نماید:

the quick brown fox jumps over the lazy dog

1- Cipher text only attack

2- Known plaintext attack

3- Chosen plaintest attack

او می‌تواند روش رمزگذاری انجام شده در این پیام را کاملاً بازگشایی نماید. بدیهی است در روش‌های پیشرفته‌ی رمزگذاری، یک تهاجم از این نوع لزوماً نمی‌تواند به مفهوم شکستن کامل روش رمزگذاری استفاده شده قلمداد شود.

در حدود 500 سال پیش روش‌های سیستم رمزگذاری تک الفبا با روش **رمزگذاری چندین الفبا**[1] بهینه‌سازی گردید. فلسفه‌ی بنیادی این روش رمزگذاری بهره‌گیری از چندین رمز چند الفبا می‌باشد که در آن یک رمز چند الفبایی به رمزگذاری یک حرف در یک مکان معین از متن خام پیام می‌پردازد. بنابراین یک حرف هنگامی که در مکان‌های مختلف در متن پیام خام ظاهر می‌شود به‌گونه‌های مختلف قابلیت رمزگذاری دارد. به عنوان مثال یک روش رمزگذاری چندین الفبایی در شکل ۸-۴ نشان داده شده است. این روش دارای دو رمز سزار به‌صورت k=5 و k=19 مطابق ردیف‌های نشان داده شده در شکل می‌باشد. بدیهی است کاربر سیستم رمز می‌تواند دو رمز C_1 و C_2 را به‌گونه‌ای گزینش نماید که در قالب یک الگوی تکراری به‌صورت:

$$C_1 , C_2 , C_2 , C_1 , C_2$$

بهره‌برداری گردد. به عبارت دیگر نخستین حرف متن خام با استفاده از C_1 رمزگذاری می‌شود و حروف دوم و سوم با بهره‌گیری از C2 و حرف چهارم و پنجم نیز به‌ترتیب از C_1 و C_2 استفاده می‌کنند. پس از آن الگو خود را تکرار می‌کند ودر این حالت حرف ششم از C_1 و حروف هفتم و هشتم از C_2 بهره می‌گیرند و رمزگذاری به‌همین حالت ادامه می‌یابد. بنابراین این متن خام:

"bob, I like to call you. alice" به‌گونه‌ی ذیل با این روش رمزگذاری و تعریف می‌شود:

"ghu, n endx yh hteq rtn tqbhx"

بدیهی است در این عبارت نخستین b در متن خام با بهره‌گیری از C_1 رمزگذاری شده است در حالیکه دومین b با استفاده از C_2. در این مثال کلید رمزگذاری و رمزگشایی پیام آگاهی از دو کلید سزار به ترتیب k=5 و k=19 و نیز الگوی $C_1 , C_2 , C_2 , C_1 , C_2$ می‌باشد.

حروف متن خام:	a b c d e f g h i j k l m n o p q r s t u v w x y z
C_1:(K=5)	f g h i j k l m n o p q r s t u v w x y z a b c d e
C_2:(K=19)	t u v w x y z a b c d e f g h i j k l m n o p q r s

شکل ۸-۴ ♦ سیستم رمز چندالفبایی با بهره‌گیری از دو کلید سزار

- **توده رمزها**

در این مقطع پیاده‌سازی رمزگذاری کلید متقارن در عصر حاضر مورد مطالعه قرار می‌گیرد. دو گروه گسترده را می‌توان برای روش‌های رمزگذاری متقارن شناسایی نمود این دو عبارتند از **جریان رمزها**[2] و **توده‌ی رمزها**[3]. رمزگذاری به روش جریان رمزها در بخش مربوط به امنیت شبکه‌های محلی بی‌سیم (WLAN) در ادامه‌ی همین فصل بررسی خواهند شد. در بخش فعلی تمرکز مطالعه بیش‌تر بر روی

1- Polyalphabetic encryption
2- Stream ciphers
3- Block ciphers

توده رمزها می‌باشد که در بسیاری از پروتکل‌های ایمن اینترنت از قبیل PGP به‌منظور امنیت در تبادل ایمیل‌ها، SSL جهت ارتباط‌های ایمن TCP و IPsec با هدف ایمن‌سازی و انتقال لایه‌ی شبکه مورد بهره‌برداری قرار گرفته‌اند.

در یک سیستم توده رمز پیام رمزگذاری شده در قالب توده‌های k بیت پردازش می‌شوند. به عنوان مثال اگر k=64 باشد آن‌گاه پیام در قالب توده‌های 64 بیت تقسیم‌بندی شده و هر توده به صورت مستقل رمزگذاری می‌شود. به‌منظور رمزگذاری یک توده سیستم رمز از نگاشت یک به یک برای k بیت توده از متن ساده‌ی پیام به k بیت توده از متن رمز بهره می‌گیرد. به‌عنوان مثال اگر فرض شود k=3 می‌باشد آن‌گاه توده رمز 3 بیت ورودی که همان متن خام پیام است را به 3 بیت خروجی یا همان متن رمز شده نگاشت می‌کند. یک روش نگاشت می‌تواند برمبنای جدول ۸-۱ باشد. بدیهی است این یک جدول نگاشت یک به یک است زیرا هر ورودی دارای خروجی متفاوت می‌باشد. این توده‌ی رمز، پیام را در قالب توده‌های 3 بیت تعریف و رمزگذاری می‌کند. به عنوان نمونه متن پیام خام: 010110001111 با بهره‌گیری از توده رمز جدول ۸-۱ به‌صورت پیام رمز: 101000111001 تعریف می‌شود.

جدول ۸-۱ ♦ توده رمز ویژه‌ی 3 بیت

ورودی	خروجی	ورودی	خروجی
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

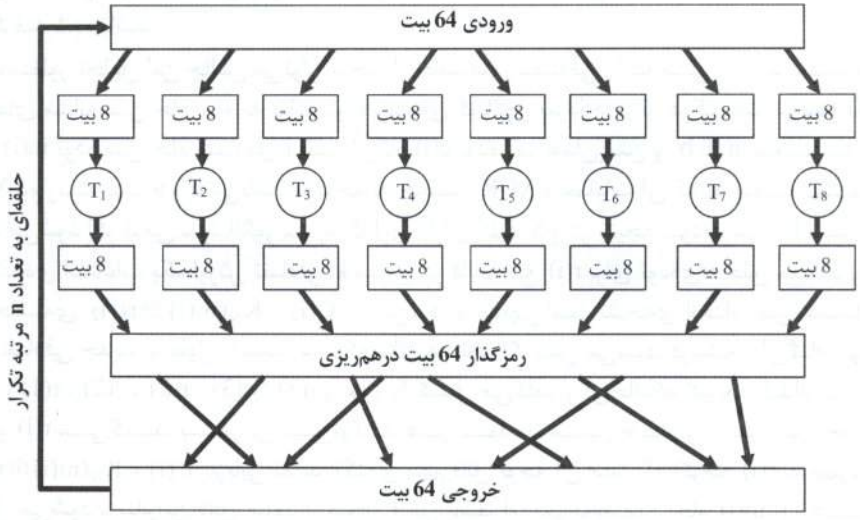
بدیهی است جدول ۸-۱ یکی از روش‌های ممکن نگاشت به‌شمار می‌آید. اکنون پرسش آن است که با این روش تعداد نگاشت‌های ممکن چقدر است؟ در پاسخ به این پرسش توجه به این نکته ضروری است که تعداد نگاشت‌های ممکن در یک سیستم رمز عبارت است از تعداد جایگشت همه‌ی ورودی‌های ممکن. به بیان دیگر در جدول ۸-۱ تعداد ورودی‌های ممکن برابر است با $2^3=8$ و در نتیجه تعداد نگاشت‌ها معادل است با $8!=40320$ روش گوناگون رمزگذاری. از آن‌جاییکه هریک از این جایگشت‌ها مشخص کننده‌ی یک نگاشت می‌باشد بنابراین تعداد 40320 نگاشت گوناگون امکان‌پذیر است. هریک از این نگاشت‌ها را می‌توان به‌صورت یک کلید رمز درنظر گرفت. اگر Alice و Bob هر دو از نگاشت یا همان کلید آگاه باشند آن‌ها می‌توانند پیام‌های مبادله شده بین یکدیگر را رمزگذاری و یا رمزگشایی نمایند.

تهاجم به این رمز با بهره‌گیری از الگوریتم بهره‌وری نابخردانه (BFS)، تلاش در رمزگشایی متن رمز شده برای همه‌ی نگاشت‌های ممکن است. در این حالت فقط با تعداد 40320 نگاشت هنگامی که k=3 می‌باشد شکستن متن رمز شده با استفاده از امکانات پردازش یک PC معمولی به‌سادگی انجام‌پذیر است. به‌منظور جلوگیری از تأثیر تهاجم‌های این گونه معمولاً از توده‌های با اندازه‌ی بزرگ‌تر که شامل k=64 بیت است، استفاده می‌شود. بدیهی است از آنجایی که تعداد نگاشت‌های ممکن برای یک توده رمز به اندازه‌ی k برابر است با $2^k!$ آن‌گاه برای ارزش‌های k مانند k=64 حجم توده رمزهای امکان‌پذیر یک ارزش نجومی را خواهد داشت.

اگرچه توده رمزهای یک جدول کامل مطابق آن‌چه در بالا به آن اشاره رفت روش‌های رمزگذاری

قدرتمندی را با بهره‌گیری از کلید متقارن هنگامی که ارزش k قابل توجه است فراهم می‌آورند اما پیاده‌سازی آن‌ها چالش برانگیز است. به عنوان مثال برای ارزش k=64 و فقط برای یک نگاشت از میان تعداد انبوهی نگاشت دو کاربر مانند Alice و Bob نیازمند نگهداری جدولی هستند که تعداد ارزش‌های ورودی آن 2^{64} است. ملاحظه می‌شود با این تعداد ورودی بسیار انبوه، پیاده‌سازی بسیار دشوار می‌باشد. علاوه بر همه‌ی این موارد اگر Alice و Bob کلیدهای خود را تغییر دهند هریک از آن‌ها بایستی اقدام به تولید مجدد جدول مربوطه نمایند. بنابراین این یک توده رمز جدول کامل که در برگیرنده‌ی نگاشت‌های از پیش تعریف شده بین همه‌ی ورودی‌ها و خروجی‌ها مانند مثال بالا باشد، از نقطه‌نظر پیاده‌سازی غیرقابل تصور خواهد بود.

آن‌چه درعمل پیاده‌سازی می‌شود بهره‌گیری از توابع به وسیله‌ی توده رمزها به‌منظور شبیه‌سازی تصادفی حالت‌ها و نگاشت‌های ممکن جداول می‌باشد. نمونه‌ای از چنین توابعی برای k=64 بیت برمبنای شکل ۸-۵ می‌باشد. تابع نخست یک توده‌ی 64 بیت را به 8 قطعه که در آن هر قطعه 8 بیت است تقسیم می‌کند. هر قطعه‌ی 8 بیت با بهره‌گیری از یک جدول 8 بیت در 8 بیت که اندازه‌ی قابل قبولی دارد پردازش می‌شود. به عنوان مثال نخستین قطعه از سوی جدول T_1 پردازش می‌شود پس از آن 8 خروجی قطعه‌ها مجدداً در قالب یک توده 64 بیت بازیابی می‌شوند. موقعیت‌های 64 بیت حاضر در توده پس از درهم آمیختن به تولید نتیجه‌ی 64 بیت خروجی می‌انجامد. این خروجی آن‌گاه به 64 بیت ورودی بازخورد می‌شود که در نتیجه‌ی آن دوره‌ی بعدی آغاز می‌شود. پس از وقوع تعداد n دوره نتیجه‌ی حاصل از تابع تأمین توده‌ی متن رمز شده‌ی 64 بیت می‌باشد. هدفی که تکرار دوره‌ها دنبال می‌کنند تأثیرگذاری هر بیت ورودی بر روی بیت‌های نهایی خروجی می‌باشد. در صورتیکه فقط از یک دوره استفاده می‌شد یک بیت مفروض ورودی فقط برروی 8 بیت از 64 بیت خروجی تأثیر می‌نمود. کلید این الگوریتم توده رمز تعداد 8 جدول درهم آمیزش شده با فرض عمومی بودن تابع درهم آمیزی می‌باشد.



شکل ۸-۵ ♦ نمونه‌ای از یک توده رمز

در حال حاضر تعداد قابل توجهی از توده رمزهای پرطرفدار مورد استفاده قرار می‌گیرند. از جمله

موارد به کار گرفته شده می‌توان از روش‌های DES[1]، 3DES و AES[2] نام برد. هریک از این استانداردها از توابع به جای جداول از پیش تعریف شده براساس الگوی شکل ۸-۵ بهره می‌گیرند. هریک از این الگوریتم‌ها خود از رشته‌ای از بیت به عنوان یک کلید رمز استفاده می‌برند. به عنوان مثال DES از توده‌های 64 بیت با یک کلید 56 بیت استفاده می‌کند و AES از توده‌های 128 بیت با کلیدهای کاربردی به طول 128، 192 و 256 بهره می‌گیرد. کلید هر الگوریتم تعیین کننده‌ی یک جدول کوچک نگاشت و ترکیبات درون آن الگوریتم است. تهاجم بهره‌وری نابخردانه (BFS) برای هریک از این رمزها تکرار دوره‌ای همه‌ی کلیدها و اعمال الگوریتم رمزگشایی با هر کلید است. بدیهی است با یک کلید به طول n تعداد کلیدهای ممکن برابر 2^n می‌باشد. پژوهش‌های انجام شده بر این تخمین می‌باشند که یک نرم‌افزار یا پردازشگر قادر به شکستن 56 بیت DES در هر ثانیه است به عبارت دیگر همه‌ی 2^{56} کلید در یک ثانیه تحلیل می‌شوند. از طرف دیگر در همین روش برای شکستن کلید 128 بیت AES زمانی معادل 149 تریلیون سال زمان لازم است.

- **زنجیره‌ی توده رمز**

در کاربردهای شبکه معمولاً لازم است پیام‌های طولانی و یا جریان داده‌ی با طول بلند رمزگذاری گردد. اگر یک توده رمز برمبنای آن‌چه در مطالب گذشته عنوان گردید به یک پیام اعمال گردد نتیجه‌ی آن تقسیم پیام در قالب توده‌های k بیت و سپس رمزگذاری مستقل هر توده خواهد بود که دشواری‌های ویژه‌ی خود را به همراه دارد. به‌منظور درک بهتر این موضوع یادآوری این نکته ضروری است که دو یا چندین توده‌ی متن خام ممکن است مشابه یکدیگر باشند. به عنوان مثال متن خام در دو یا چندین توده می‌تواند به صورت "HTTP/1.1" باشد. برای این توده‌های مشابه یک توده رمز به تولید متن رمز مشابه می‌پردازد. یک مهاجم به آسانی می تواند پی به محتوای متن خام پیام برده در صورتی که توده‌های متن رمز مشابه یکدیگر را ملاحظه نماید و ممکن است حتی قادر به رمزگشایی کل پیام از طریق شناسایی توده‌های متن رمز مشابه و بهره‌گیری از اطلاعات آن‌ها در ارتباط با ساختار پروتکل به کار گرفته شده باشد.

به‌منظور تحلیل این چالش می‌توان برخی از حالت‌های تصادفی را با متن رمز ترکیب نمود تا توده‌های مشابه متن خام پیام به تولید توده رمزهای گوناگون بپردازد. برای درک بهتر موضوع اگر فرض شود m(i) توده متن خام شماره‌ی i باشد آن‌گاه c(i) توده رمز همان متن و $a \oplus b$ متعلق به حاصل XOR دو رشته بیت a و b می‌باشد. با توجه به خاصیت XOR و عملیات آن که به صورت بیت به بیت انجام می‌شود اگر فرض شود الگوریتم رمزگذاری دارای کلید S و در نتیجه توده رمز K_S است آن‌گاه فرستنده‌ی اطلاعات یک ارزش تصادفی k بیت را در قالب تابع r(i) برای توده‌ی i خلق می‌کند و سپس به محاسبه‌ی $C(i) = K_S(m(i) \oplus r(i))$ آن می‌پردازد. بدیهی است نتیجه‌ی انجام این عملیات یک عدد تصادفی جدید به طول k بیت است که برای هر توده گزینش می‌شود. فرستنده آن‌گاه توده‌های c(1)، r(1)، c(2)، r(2)، c(3)، r(3) و غیره را گسیل می‌کند. از آن‌جائیکه گیرنده اقدام به دریافت c(i) و r(i) می‌کند. بنابراین می‌تواند هر توده از متن خام را از طریق محاسبه‌ی $C(i) = K_S(m(i) \oplus r(i))$ بازیابی نماید. نکته‌ی مهم قابل توجه آن است که اگرچه r(i) به صورت آشکار گسیل می‌شود و بنابراین قابل شنود از سوی T می باشد. اما نمی‌تواند متن خام m(i) را به‌دست آورد

1- Data Encryption Standard

2- Advanced Encryption Standard

زیرا او از کلید K_S آگاهی ندارد. علاوه بر آن اگر دو توده‌ی متن خام m(i) و m(j) مشابه یکدیگر باشند توده‌های رمز وابسته به آن‌ها یعنی c(i) و c(j) تا زمانی که ارزش‌های تصادفی r(i) و r(j) متفاوت هستند آن‌ها نیز مشابه یکدیگر نخواهند شد.

به عنوان نمونه توده رمز 3 بیت در جدول ۸-۱ را در نظر بگیرید. اگر فرض شود متن خام پیام به صورت 010010010 باشد آن‌گاه اگر Alice این پیام را مستقیماً رمز نماید بدون درنظر گرفتن تصادفی بودن، متن رمز شده‌ی حاصل بر مبنای جدول به صورت 101101101 تعریف می‌شود. اگر مهاجم T این متن رمز را شنود نماید از آن‌جائیکه هریک از سه توده‌ی رمز مشابه یکدیگر می‌باشند، او می‌تواند این نتیجه را به‌دست آورد که هریک از سه توده متن خام نیز مشابه یکدیگرند. اکنون فرض کنید Alice توده‌های تصادفی r(1)=001 ، r(2)=111 و r(3)=100 را تولید نماید و از همان روش بالا جهت تولید متن رمز c(1)=100 ، c(2)=010 و c(3)=000 بهره گیرد. یادآوری این نکته ضروری است که سه توده‌ی متن رمز متفاوت می‌باشند. اگرچه توده‌های متن ساده مشابه یکدیگر هستند. بدیهی است که Alice توده‌های c(1) ، r(1) ، c(2) و r(2) را گسیل می‌کند. با همین استدلال می‌توان نتیجه گرفت Bob نیز متن خام پیام اصلی را با بهره‌گیری از کلید مشترک K_S همین گونه به‌دست می‌آورد.

یک پژوهشگر دقیق ملاحظه خواهد نمود که عامل تصادفی بودن توده‌ها یک چالش را برطرف نموده اما دشواری دیگری را نتیجه می‌دهد. یکی از این دشواری‌ها آن است که Alice بایستی دوبرابر تعداد بیت‌هایی را ارسال نماید که قبلاً گسیل کرده است. برای هر بیت رمز، او بایستی یک بیت تصادفی را ارسال نماید در نتیجه پهنای باند لازم را نیز دوبرابر نماید. به‌منظور دستیابی به هدف، توده رمزها از روش ویژه‌ای بهره می‌گیرند که اصطلاحاً به آن **زنجیره‌ی توده رمز (CBC)**[1] گفته می‌شود. فلسفه‌ی اصلی آن است که فقط یک ارزش تصادفی همراه نخستین پیام ارسال شود و سپس فرستنده و گیرنده را وادار به استفاده از توده‌های رمز شده‌ی محاسبه شده به جای عدد تصادفی نمود. به بیان دیگر CBC به گونه‌ی ذیل عمل می‌نماید:

۱- قبل از رمز نمودن پیام یا جریان داده‌ها فرستنده یک رشته‌ی تصادفی به طول k بیت را تولید می‌کند این رشته اصطلاحاً **بردار مقداردهی اولیه (IV)**[2] نامیده می‌شود. این بردار معمولاً **C(0)** نامیده می‌شود. فرستنده IV را در قالب متن خام به گیرنده گسیل می‌کند.

۲- برای توده‌ی نخست فرستنده ارزش $m(1) \oplus c(1)$ را محاسبه می‌کند به عبارت دیگر XOR نخستین توده از متن خام با IV محاسبه می‌شود سپس نتیجه را از درون الگوریتم توده رمز عبور می‌دهد تا توده متن رمز شده‌ی آن را به‌دست آورد. به عبارت دیگر تابع $c(1) = K_S(m(1) \oplus c(0))$ را محاسبه می‌کند و آن‌گاه فرستنده توده‌ی رمز شده‌ی c(1) را به گیرنده ارسال می‌کند.

۳- برای توده‌ی i فرستنده توده‌ی متن رمز شماره‌ی i را از طریق $c(i) = K_S(m(i) \oplus c(i-1))$ تولید می‌کند.

اکنون تأثیر حاصل از این رویکرد بررسی می‌شود. نخست گیرنده هنوز هم قادر به بازیابی پیام اصلی می‌باشد. هنگامی که گیرنده c(i) را دریافت می‌دارد آن را با کلید K_S رمزگشایی می‌نماید تا از

1- Cipher block chaining

2- Initialization vector

طریق آن $s(i) = m(i) \oplus c(i-1)$ را به‌دست آورد. از آن‌جائیکه گیرنده از c(i-1) آگاهی دارد در نتیجه توده‌ی متن خام را از $m(i) = s(i) \oplus c(i-1)$ به‌دست می‌آورد. نکته‌ی دیگر آن که حتی اگر دو توده‌ی متن خام مشابه یکدیگر باشند متون رمز شده‌ی وابسته به آن‌ها همواره متفاوت خواهد بود. نکته‌ی سوم آن‌که اگرچه فرستنده IV را در قالب متن خام گسیل می‌نماید یک مهاجم هنوز هم قادر به رمزگشایی توده‌های متن رمز شده نمی‌باشد زیرا وی از کلید رمز S آگاهی ندارد. سرانجام فرستنده فقط یک توده‌ی سرآیند را که همان IV است ارسال می‌کند و بنابراین کاربرد پهنای باند را برای پیام‌های طولانی که شامل صدها توده است بسیار اندک افزایش می‌دهد.

به عنوان یک مثال متن رمز توده‌ی 3 بیت در جدول ۸-۱ را برای متن خام 010010010 و IV=c(0)=001 درنظر بگیرید. فرستنده نخست از IV بهره می‌گیرد تا:

$C(1) = K_S(m(1) \oplus C(0)) = 100$ را محاسبه نماید. فرستنده سپس:

و $C(2) = K_S(m(2) \oplus c(1)) = K_S(010 \oplus 100) = 000$

$C(3) = K_S(m(3) \oplus c(2)) = K_S(010 \oplus 000) = 101$ را محاسبه می‌نماید.

بدیهی است گیرنده با آگاهی IV و K_S می‌تواند متن ساده‌ی اصلی را بازیابی نماید. زنجیره‌ی توده رمز (CBC) یک تأثیر مهم نیز دارد، هنگامی که پروتکل‌های ایمن شبکه طراحی می‌شود معمولاً نیازمند تأمین روشی در درون پروتکل به‌منظور توزیع IV از فرستنده به گیرنده می‌باشد در ادامه‌ی همین فصل اعمال این سیاست برای چندین پروتکل بررسی می‌شود.

۸-۴ سیستم رمزگذاری کلید عمومی

بیش از 2000 سال از هنگامی که رمز سزار تعریف شد و نیز تا دهه‌ی 1970 میلادی ارتباط رمزگذاری شده نیازمند آن است که دو سوی ارتباط از یک کلید رمز متقارن مشترک به‌منظور رمزگذاری و رمزگشایی بهره گیرند. یکی از دشواری‌های این رویکرد آن است که دو سوی ارتباط بایستی به نوعی برروی بهره‌گیری از یک کلید مشترک توافق نمایند اما برای انجام این مهم دو طرف ارتباط شاید لازم باشد ابتدا یکدیگر را ملاقات نموده و به‌صورت حضوری بر روی کلید رمز مشترک مابین خود به توافق برسند، پس از آن ارتباط خود را به‌صورت رمز شده انجام دهند. در دنیای شبکه طرفین ارتباط ممکن است هرگز یکدیگر را ملاقات ننموده و نیز هرگز با یکدیگر از طریق شبکه تعامل نداشته باشند. آیا این امکان وجود دارد که ارتباط بین دوطرف به صورت رمز شده و بدون بهره‌گیری از یک کلید رمز مشترک انجام پذیرد؟ و یا به بیان دیگر اطلاعات کلید از قبل به‌گونه‌ی عام آشکار باشد؟

در سال 1976 میلادی دیفی و هلمن از طریق یک **الگوریتم ویژه**[1] که به نام ایشان معروف می‌باشد پیاده‌سازی این مهم را نشان دادند. آن‌ها رویکردی زیرکانه و ظریف را به‌منظور انجام ارتباط‌های امن به نمایش گذاردند که نتیجه‌ی آن توسعه و تکامل سیستم‌های رمزگذاری با بهره‌گیری از کلید عمومی در حال حاضر می‌باشد. در ادامه ملاحظه خواهد شد که سیستم‌های رمزگذاری کلید عمومی دارای چندین ویژگی مهیج است که آن‌ها را سودمند نه فقط برای رمزگذاری بلکه برای اعتبارسنجی و امضای دیجیتالی می‌کند. جالب توجه آنکه اخیراً پیشنهادهای مشابهی در این زمینه در قالب مجموعه‌ای از گزارش‌های محرمانه از سوی پژوهشگران گروه امنیت ارتباط‌های الکترونیکی بریتانیا (CESG) انتشار یافته است. همان‌گونه که انتظار می‌رود ایده‌های بزرگ همواره به‌صورت

1- Diffi-Helman algorithm

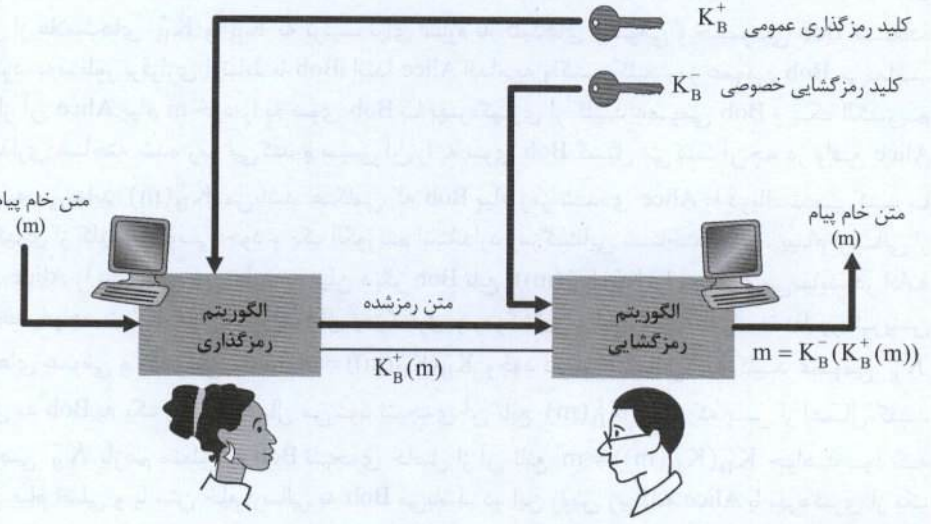
مستقل در بسیاری از مکان‌های گوناگون جغرافیایی ظاهر می‌شوند. بدیهی است پیشرفت‌ها در زمینه کلیدهای رمز عمومی فقط در مجامع محرمانه روی نداده است بلکه اغلب آن‌ها در اختیار عموم و با آگاهی آن‌ها انجام پذیرفته است.

بهره‌گیری از کلید رمزگذاری عمومی از لحاظ مفهومی بسیار ساده است. اگر فرض شود Alice مایل به برقراری ارتباط با Bob است همان‌گونه که در شکل ۸-۶ نشان داده شده است به جای آن‌که Bob و Alice هر دو از یک کلید رمز محرمانه به صورت مشترک بین خودمانند سیستم‌های کلید متقارن استفاده نمایند، Bob به عنوان دریافت کننده‌ی پیام از سوی Alice از دو کلید بهره می‌گیرد. یک **کلید عمومی**[1] که در اختیار هر فردی در هر گوشه‌ی گیتی قرار می‌گیرد و البته مهاجم فرضی ما T را نیز شامل می‌شود و نیز یک **کلید خصوصی**[2] که فقط Bob از آن آگاهی دارد. در ادامه‌ی مطالب این بخش از علامت‌های K_B^+ و K_B^- به ترتیب برای اشاره به کلیدهای عمومی و خصوصی Bob استفاده می‌شود. به‌منظور برقراری ارتباط با Bob، ابتدا Alice اقدام به واکشی کلید رمز عمومی Bob می‌نماید. پس از آن Alice پیام m خود را به سوی Bob با بهره‌گیری از کلید عمومی Bob و یک الگوریتم رمزگذاری شناخته شده رمز می‌کند و سپس آن را به سوی Bob گسیل می‌کند. آن‌چه در واقع Alice محاسبه می‌نماید $K_B^+(m)$ می‌باشد. هنگامی که Bob پیام رمز شده‌ی Alice را دریافت می‌کند با بهره‌گیری از کلید خصوصی خود و یک الگوریتم استاندارد رمزگشایی شناخته شده، پیام ارسالی از سوی Alice را بازگشایی می‌نماید. به بیان دیگر Bob تابع $K_B^-(K_B^+(m))$ را محاسبه می‌نماید. در ادامه ملاحظه خواهد شد که انواع الگوریتم‌های رمزگذاری و رمزگشایی و نیز روش‌هایی به‌منظور گزینش کلیدهای عمومی و خصوصی مانند: $K_B^-(K_B^+(m)) = m$ وجود دارند. هنگامی که کلید عمومی K_B^+ متعلق به Bob به یک پیام m اعمال می‌شود نتیجه‌ی آن تابع $K_B^+(m)$ است که پس از اعمال کلید خصوصی K_B^- بازهم متعلق به Bob نتیجه‌ی حاصل از آن تابع $K_B^-(K_B^+(m)) = m$ خواهد بود که همان پیام اصلی و یا متن خام ارسالی به Bob می‌باشد. در این روش زیرکانه Alice با بهره‌گیری از یک کلید رمز عمومی در دسترس همه و متعلق به Bob پیام رمز شده‌ی خود را به سوی Bob ارسال می‌کند بدون آنکه هیچ یک از آن‌ها الزامی به انتشار کلیدهای رمز داشته باشند! بدیهی است با جایگزینی دو کلید رمزگذاری عمومی و خصوصی با یکدیگر می‌توان به همان نتیجه‌ی شگفت‌آور اشاره شده نیز رسید، به بیان دیگر $K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m$ بهره‌گیری از کلید رمز عمومی بنابراین از لحاظ درک ساده است. دو نگرانی که در ذهن به وجود می‌آید عبارت است از: **(۱)** اگرچه یک مهاجم پیام رمز شده‌ی Alice را شنود می‌کند و آن را غیرعادی قلمداد می‌کند اما مهاجم از دو موضوع آگاهی دارد یکی کلید رمز عمومی Bob است که در اختیار همگی قرار می‌گیرد و دیگری الگوریتم رمزگذاری که Alice از آن برای تشکیل پیام رمز خود بهره گرفته است. در این حالت مهاجم فرضی T می‌تواند یک تهاجم گزینش متن خام را سازماندهی نماید و این عمل را با بهره‌گیری از یک الگوریتم رمزگذاری شناخته شده‌ی استاندارد و کلید رمز عمومی Bob برای هر پیام دلخواهی که T گزینش می‌کند، انجام دهد. مهاجم ممکن است تلاش در رمزگذاری همه‌ی پیام‌ها و یا بخشی از آن‌ها نماید که به نظر وی Alice برای Bob فرستاده است. بدیهی است اگر کلید رمزگذاری عمومی بایستی کارایی

1- Public key

2- Private key

بهینه به‌همراه داشته باشد، گزینش کلید و الگوریتم رمزگذاری و یا رمزگشایی بایستی به‌گونه‌ای عمل نمایند که کار را برای مهاجم دشوار و یا غیرممکن نمایند و از طریق آن‌ها وی نتواند به کلید خصوصی Bob دسترسی حاصل نماید. (۲) نگرانی دوم از آن‌جا نتیجه می‌شود که کلید رمزگذاری Bob حالت عمومی دارد و هر فردی می‌تواند یک پیام رمز شده‌ای را به Bob ارسال نماید و در این میان نه تنها Alice بلکه فرد دیگری که خود را به جای Alice می‌تواند معرفی نماید نیز قادر به گسیل پیام به Bob می‌باشد. در حالتی که از یک کلید محرمانه‌ی مشترک استفاده می‌شد این واقعیت که فرستنده از کلید رمز به‌گونه‌ی انحصاری آگاهی دارد آن کلید نیز شناسایی فرستنده را به گیرنده برعهده می‌گرفت. در حالت کلید عمومی این حالت امنیت وجود ندارد و بنابراین نیاز به یک نوع امضای دیجیتالی می‌باشد، موضوعی که در بخش ۸-۷ تحلیل خواهد شد.



شکل ۸-۶ ♦ سیستم رمزگذاری کلید عمومی

• الگوریتم RSA

اگرچه ممکن است الگوریتم‌های بی‌شماری وجود داشته باشند که نگرانی‌های اشاره شده را برطرف نمایند اما در بین آن‌ها **الگوریتم RSA**[1] که نام خود را از بنیان‌گذارانش یعنی: Ron Rivest، Adi Shamir و L.Adleman گرفته است یکی از مشهورترین الگوریتم‌های به‌کار گرفته شده در سیستم رمزگذاری کلید عمومی می‌باشد. ابتدا چگونگی عملکرد الگوریتم RSA تشریح می‌شود. الگوریتم RSA بهره‌گیری گسترده‌ای از عملیات محاسباتی با بهره‌گیری از محاسبه‌ی تقسیم پیمانه‌ی n دارد.

به‌منظور تشریح این‌گونه محاسبه یادآوری این نکته ضروری است که **X mod n** به مفهوم باقی مانده‌ی تقسیم x بر n می‌باشد. به عنوان نمونه 19mod5=4 می‌باشد. در محاسبات پیمانه‌ی n عملیات متداول جمع، ضرب و توان نیز انجام‌پذیر است اما نتیجه‌ی حاصل از هر عملیات با یک مقدار ارزشی کامل به عنوان باقی مانده قابل جایگزینی است که از تقسیم بر n حاصل می‌شود. جمع و ضرب ریاضی

1- RSA algorithm

پیمانه‌ی n از طریق حقایق ذیل امکان‌پذیر است:

[(a modn) + (b modn)] mod n = (a+b) mod n
[(a modn) - (b modn)] mod n = (a-b) mod n
[(a modn) . (b modn)] mod n = (a.b) mod n

از رابطه‌ی سوم این نتیجه حاصل می‌شود که: $(amodn)^d \ modn=a^d \ modn$ این یک رابطه‌ی بسیار سودمند در مطالب این بخش می‌باشد.

اکنون اگر فرض شود Alice مایل به ارسال یک پیام رمزگذاری شده‌ی RSA به دوست خود Bob برمبنای شکل ۸-۶ می‌باشد. در بحث مربوط به RSA آگاهی از این نکته ضروری است که یک پیام الگویی از بیت‌ها است و هر الگوی بیت انحصاراً می‌تواند از طریق یک عدد کامل در مجاورت با الگوی بیت به نمایش درآید. به عنوان مثال اگر 1001 الگوی بیت یک پیام باشد آن‌گاه این پیام می‌تواند به‌صورت ارزش دسیمال 9 بیان گردد. بنابراین هنگامی که یک پیام با بهره‌گیری از RSA رمزگذاری می‌شود آن پیام معادل است با رمزگذاری ارزش کامل انحصاری که آن پیام را نمایندگی می‌کند. دو عنصر وابسته به RSA وجود دارد، این دو عبارتند از:

- گزینش کلیدهای عمومی و خصوصی.
- الگوریتم‌های رمزگذاری و رمزگشایی.

برای تولید کلیدهای رمز عمومی و خصوصی RSA، گام‌های ذیل از سوی Bob به عنوان گیرنده‌ی پیام به اجرا درمی‌آید:

۱- گزینش دو عدد اول بزرگ p و q. پرسش آن است که بزرگی p و q چقدر است؟ پاسخ آن است که هرچه ارزش p و q بزرگ‌تر باشد شکستن رمز RSA دشوارتر می‌شود اما عملیات رمزگذاری و رمزگشایی نیز به همان نسبت طولانی می‌شود. تجربه‌ی عملی RSA این واقعیت را نشان می‌دهد که حاصل ضرب p و q بایستی در اطراف 1024 بیت باشد.

۲- محاسبه‌ی $n=p.q$ و $z=(p-1)\ (q-1)$

۳- گزینش عددی مانند $e<n$ که دارای ضریب مشترکی به جز یک با z نمی باشد. در این حالت گفته می‌شود e و z به صورت نسبی اعداد اول می‌باشند. حرف e از آن جهت استفاده می‌شود که مفهوم آن کاربرد در **رمزگذاری** است.

۳- یافتن عددی مانند d که حاصل de-1 آن قابلیت تقسیم بدون باقی مانده بر z را دارد. حرف d به این منظور مورد استفاده قرار می‌گیرد زیرا از آن برای **رمزگشایی** استفاده می‌شود. به بیان دیگر با فرض در اختیار داشتن e ارزش d به‌گونه‌ای گزینش می‌شود که رابطه‌ی ذیل محقق گردد:

$$ed \bmod z = 1 \qquad (۱-۸)$$

۴- کلید عمومی که Bob در اختیار عموم قرار می‌دهد یعنی K_B^+ زوج ارزش (n,e) و کلید رمز خصوصی وی یعنی K_B^- زوج ارزش (n,d) می‌باشند.

رمزگذاری انجام شده از سوی Alice و رمزگشایی آن از طرف Bob در قالب گام‌های ذیل صورت می‌گیرد:

- فرض می‌شود Alice مایل به ارسال یک رشته بیت که بیان‌کننده‌ی یک ارزش کامل m است به Bob می‌باشد. بدیهی است $m<n$ فرض شده است. برای رمزگذاری این پیام Alice

عملیات محاسبه‌ی توان m^e را انجام داده و سپس باقی‌مانده‌ی کامل تقسیم m^e بر n را محاسبه می‌کند. به بیان دیگر ارزش رمز شده‌ی c از متن خام پیام (m) متعلق به Alice نتیجه می‌شود:

$$c=m^e \bmod n \qquad (۲-۸)$$

الگوی بیت وابسته به متن رمز c آن‌گاه به سوی Bob گسیل می‌شود.

- متن پیام رمز c به صورت ذیل از طریق Bob رمزگشایی می‌شود:

$$m=c^d \bmod n \qquad (۳-۸)$$

برای محاسبه رابطه‌ی (۸-۳) Bob نیازمند استفاده از کلید خصوصی (n,d) متعلق به خود می‌باشد.

نمونه‌ای ساده از RSA را می‌توان چنین تصور نمود: اگر فرض شود Bob مقدار p=5 و q=7 را گزینش نمایدکه عملاً ارزش‌های بسیار کوچک به‌منظور تأمین امنیت می‌باشند آن‌گاه n=35 و z=24 خواهد بود و Bob به گزینش e=5 می‌پردازد زیرا 5 و 24 دارای ضریب مشترکی نمی‌باشند. سرانجام Bob ارزش d=29 را گزینش می‌کند زیرا 5x29-1 همان (ed-1) خواهد بود که عیناً بر ارزش 24 قابل تقسیم می‌باشد. پس از آن Bob دو ارزش n=35 و e=5 را برای آگاهی عمومی انتشار می‌دهد و ارزش d=29 را محرمانه به عنوان کلید خصوصی خود نگهداری می‌کند. با مشاهده‌ی این دو ارزش عمومی یعنی n=35 و e=5 اگر فرض شود Alice مایل به ارسال حروف l، o، v و e به Bob می‌باشد آن‌گاه تفسیر هر حرف برمبنای یک عدد مابین 1 و 26 که در آن حرف a=1 و حرف z=26 است انجام می‌گیرد و سپس Alice و Bob هر دو عملیات رمزگذاری و رمزگشایی را انجام می‌دهند که به ترتیب مطابق جداول ۸-۲ و ۸-۳ می‌باشد. بدیهی است در این مثال هر چهار حرف به عنوان یک پیام مجزا در نظر گرفته شده است. در یک رویکرد واقعی هریک از چهار حرف تبدیل به معادل 8 بیت ASCII مربوط به خود می‌شوند و آن‌گاه ارزش کامل وابسته به الگوی 32 بیت نتیجه شده رمز می‌شود. چنین رویکرد واقعی، اعدادی را تولید می‌کند که برای درج در این کتاب بسیار طولانی می‌باشند.

با توجه به مثال بسیار ساده‌ی جداول ۸-۲ و ۸-۳ هنوز هم ملاحظه می‌شود که ارقام تولید شده بزرگ می‌باشند. از طرف دیگر قبلاً اشاره شد p و q از لحاظ طول بایستی چند صد بیت درازا داشته باشند آن‌گاه موارد عملی جدی در ارتباط با RSA در ذهن پدیدار می‌شود.

جدول ۸-۲ ♦ رمزگذاری Alice با به‌کارگیری RSA و ارزش‌های e=5 و n=35

Plaintext Letter	m: numeric representation	m^e	Ciphertext $c=m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

چگونه اعداد نخست گزینش می‌شوند؟ ارزش‌های e و d چگونه انتخاب می‌شوند؟ عملیات محاسبه‌ی توان اعداد بزرگ چگونه انجام می‌شود؟ این موارد موضوعاتی است که از چارچوب این کتاب خارج می‌باشد اما در هر حال خوانندگان علاقه‌مند به مرجع [Kaufman 1995] جهت آگاهی بیش‌تر راهنمایی می‌شوند.

جدول ۸-۳ ♦ رمزگشایی Bob با استفاده از RSA و ارزش‌های d=29 و n=35

Ciphertext C	C^d	$m=c^d$ mod n	Plaintext Letter
17	481968572106750915091509141825223071697	12	l
15	127834039403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	100000000000000000000000000000	5	e

- **کلیدهای تعامل**

با اندکی دقت ملاحظه می‌شود فرآیند محاسبه‌ی توان از سوی RSA زمان قابل توجهی را نیاز دارد در صورتی‌که روش DES حداقل 100 مرتبه از لحاظ نرم‌افزاری و مابین 1000 تا 10000 مرتبه از نقطه نظر سخت افزاری سریع‌تر می‌باشد. بنابراین در عمل از RSA در ارتباط با رمزگذاری کلید متقارن استفاده می‌شود. به عنوان مثال اگر Alice مایل به ارسال مقادیر قابل توجهی داده‌ی رمزگذاری شده می‌باشد بایستی عملیات ذیل را انجام دهد: نخست Alice کلیدی را گزینش می‌کند که از آن فقط برای رمزگذاری داده‌ها استفاده می‌شود. این کلید انحصاری رمزگذاری داده‌ها را **کلید تعامل**[1] می‌خوانند و نماد آن نیز K_S می‌باشد. پس از آن Alice بایستی Bob را از این کلید تعامل آگاه نماید از آنجائیکه این کلید مشترک می‌باشد از آن به همراه یک کلید متقارن رمز با DES و یا AES استفاده می‌شود. با بهره‌گیری از کلید عمومی Bob کلید تعامل ارسالی از سوی Alice رمزگذاری می‌شود به بیان دیگر $c=(K_S)^e \bmod n$ محاسبه می‌شود. پس از آن Bob کلید تعامل رمزگذاری شده‌ی RSA یعنی c را دریافت و آن را رمزگشایی می‌نماید تا کلید تعامل K_S را به‌دست آورد. در نتیجه Bob از کلید تعامل استفاده شده از سوی Alice آگاه شده و از آن برای انتقال داده‌ی رمزگذاری شده بهره‌گیری می‌کند.

پرسشی که ممکن است در ذهن ایجاد شود آن است که RSA چگونه کار می‌کند؟ رمزگذاری و رمزگشایی RSA به نوعی شگفت‌انگیز است زیرا چگونه است که با اعمال الگوریتم رمزگذاری و سپس الگوریتم رمزگشایی می‌توان پیام اصلی را بازیابی نمود؟ جهت درک موضوع چگونگی عملکرد RSA بار دیگر n=p.q درنظر گرفته می‌شود که در آن p و q اعداد نخست بزرگ به کار گرفته شده در الگوریتم RSA می‌باشند. یادآور می‌شود در رمزگذاری RSA یک پیام m به‌گونه‌ی انحصاری از طریق یک عدد کامل نمایندگی می‌شود که توانی از e با بهره‌گیری از محاسبه‌ی پیمانه‌ی -n می‌باشد. رابطه (۸-۴) این موضوع را نشان می‌دهد:

$$c=m^e \bmod n \qquad (۴-۸)$$

از طرف دیگر رمزگشایی با بهره‌گیری از محاسبه‌ی توان این ارزش با d و بهره‌گیری از همان روش رابطه‌ی (۸-۴) را نتیجه می‌دهد. حاصل یک گام رمزگذاری به دنبال یک مرحله‌ی رمزگشایی بنابراین رابطه (۸-۵) می‌باشد:

$$(m^e \bmod n)^d \bmod n \qquad (۵-۸)$$

1- Session key

اکنون به تحلیل این کمیت می‌پردازیم. قبلاً اشاره گردید یک ویژگی مهم محاسبه‌ی پیمانه‌ی n- نتیجه‌ی حاصل از رابطه‌ی (۸-۶) برای هر ارزشی از a، n و d می‌باشد.

$$(a \bmod n)^d \bmod n = a^d \bmod n \qquad (۸-۶)$$

با بهره‌گیری از رابطه‌ی $a=m^e$ در رابطه (۸-۶) می‌توان معادله‌ی (۸-۷) را نتیجه‌گیری نمود:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n \qquad (۸-۷)$$

بنابراین کافی است نشان دهیم که $m^{ed} \bmod n = m$ می‌باشد. با بهره‌گیری از ویژگی‌های نظریه‌ی اعداد نیازمند این نتیجه می‌باشیم که به ما نشان دهد اگر p و q اعداد نخست و نیز اگر $n=p.q$ و $z=(p-1)(q-1)$ باشند آن‌گاه $x^y \bmod n$ مشابه است با:

$$[x^{(y \bmod z)}] \bmod [n] \qquad (۸-۸)$$

پس از اعمال نتیجه‌ی حاصل از رابطه‌ی (۸-۸) با مقادیر $x=m$ و $y=ed$ معادله‌ی (۸-۹) نتیجه می‌شود:

$$m^{ed} \bmod n = [m^{(ed \bmod z)}] \bmod [n] \qquad (۸-۹)$$

یادآوری می‌شود e و d به‌گونه‌ای گزینش شده‌اند که $ed \bmod z = 1$ می‌باشد. رابطه‌ی (۸-۱۰) نتیجه‌ی حاصل از این گزینش است:

$$[m^{ed}] \bmod [n] = m^1 \bmod n = m \qquad (۸-۱۰)$$

معادله‌ی (۸-۱۰)نتیجه‌ی مطلوب است. ابتدا با محاسبه‌ی توان e از ارزش کلید که مفهوم آن رمزگذاری است و سپس محاسبه‌ی توان d که رمزگشایی متن رمز شده است ارزش اصلی پیام یعنی m حاصل می‌شود. جالب توجه آن است که اگر فرآیند محاسبه عکس شود یعنی ابتدا عملیات محاسبه‌ی توان d و سپس e انجام شود یعنی نخست رمزگشایی و پس از آن رمزگذاری انجام گردد بازهم ارزش اصلی پیام یعنی m نتیجه می‌شود. معادله‌ی (۸-۱۱) حاصل به‌دست آمده را نشان می‌دهد:

$$(m^d \bmod n)^e \bmod n = [m^{de}] \bmod [n] = [m^{ed}] \bmod [n] \qquad (۸-۱۱)$$
$$= [(m^e \bmod n)^d] \bmod [n]$$

امنیت RSA براین واقعیت پایه‌گذاری شده است که الگوریتم شناخته شده‌ای برای تجزیه‌ی یک عدد وجود ندارد. در این حالت ارزش عمومی n در درون اعداد نخست p و q چنانچه p و q شناخته شده باشند با ارزش عمومی مفروض e به‌سادگی کلید رمز d قابل محاسبه می‌باشد. از طرف دیگر روشن نیست آیا یک الگوریتم سریع برای تجزیه‌ی یک عدد می‌توان طراحی نمود؟ به بیان دیگر ضمانتی برای اطمینان از امنیت RSA وجود ندارد.

الگوریتم رمزگذاری کلید عمومی معروف دیگر عبارت است از الگوریتم دیفی و هلمن می‌باشد. این الگوریتم اگرچه انعطاف‌پذیری RSA را در رمزگذاری پیام‌های با طول متغیر ندارد اما در هر حال از آن می‌توان برای تعریف یک کلید تعامل متقارن جهت کاربرد در رمزگذاری پیام‌ها بهره گرفت.

۸-۵ یکپارچگی پیام و اعتبارسنجی نقطه‌ی انتهایی

در بخش قبل ملاحظه شد چگونه از امکانات رمزگذاری به‌منظور تأمین امنیت دو موجودیت ارتباطی می‌توان بهره گرفت. در زیربخش جاری موضوع محرمانه‌سازی مهم دیگری تحلیل می‌شود که در ارتباط

با حفظ **یکپارچگی پیام**[1] و به عبارت دیگر **اعتبارسنجی** پیام می‌باشد. همراه با حفظ یکپارچگی پیام دو عنوان وابسته به این موضوع نیز تحلیل خواهد شد این دو عبارتند از **امضای دیجیتالی و اعتبارسنجی نقطه‌ی انتهایی**[2]. موضوع یکپارچگی پیام را در قالب مثال Alice و Bob دنبال می‌کنیم. اگر فرض شود Bob پیامی را دریافت نماید که آن پیام خام و یا رمز شده باشد و وی بر این باور باشد که این پیام از سوی Alice ارسال شده است چگونه Bob بایستی از اعتبار این پیام اطمینان حاصل نماید؟ بدیهی است Bob از دو موضوع مهم بایستی اطمینان حاصل نماید:

۱- پیام یقیناً از سوی Alice ارسال شده است.

۲- پیام در طول مسیر خود تا رسیدن به Bob دچار تغییر و جعل نشده است.

در بخش‌های بعدی همین فصل (۸-۱۰ به بعد) ملاحظه خواهد شد که موضوع یکپارچگی پیام برای همه‌ی پروتکل‌های امن شبکه بسیار حیاتی و اثرگذار می‌باشد.

به‌منظور درک بهتر اهمیت موضوع یک شبکه‌ی کامپیوتری را درنظر بگیرید که از الگوریتم مسیریابی وضعیت پیوند (LSA) مانند OSPF با هدف یافتن مسیرهای بین هر زوج روتر در شبکه استفاده می‌کند (فصل ۴). در یک الگوریتم LSA هر مسیریاب بایستی یک پیام وضعیت پیوند (LS) را به همه‌ی روترهای شبکه فراپخش نماید. یک پیام LS متعلق به مسیریاب عبارت است از فهرستی از مجاوران مستقیماً متصل به آن روتر و نیز هزینه‌ی مستقیم این همسایگان، بنابراین هنگامی که یک روتر پیام LS را از همه‌ی مسیریاب‌های دیگر دریافت می‌دارد می‌تواند نقشه‌ی کامل شبکه را تعریف نماید و الگوریتم مسیریابی حداقل هزینه را برای آن اجرا نماید و پس از آن جدول مسیریابی به جلو را برای آن پیکربندی نماید. یک تهاجم نسبتاً ساده بر روی الگوریتم مسیریابی برای فردی مانند T آن است که پیام‌های LS انحرافی و آلوده را که همراه آن‌ها اطلاعات غلط وضعیت پیوند (LS) وجود دارد به سوی هدف مورد نظر انتشار دهد. بنابراین نیاز به یکپارچگی پیام نمایان می‌شود زیرا مسیریاب B یک پیام LS را از روتر A دریافت می دارد، مسیریاب B بایستی اطمینان حاصل کند که مسیریاب A واقعاً پیام را تعریف نموده است و علاوه بر آن هنگام انتقال پیام دچار جعل و یا تغییر نشده است. در این بخش روش‌های ارزیابی یکپارچگی پیام که از سوی پروتکل‌های امن شبکه مورد استفاده قرار می‌گیرند بررسی شده‌اند. قبل از ورود به این موضوعات ابتدا ابزار مهم این عملیات یعنی **توابع درهم‌ریزی رمزگذاری (CHF)**[3] تشریح می‌شود.

• توابع درهم ریزی رمزگذاری (CHF)

همان‌گونه که در شکل ۸-۷ نشان داده شده است یک تابع درهم‌ریزی (HF) با بهره‌گیری از یک ورودی m یک رشته‌ی با طول ثابت H(m) که به آن هاش (hash) گفته می‌شود را تعریف می‌کند. کد مجموع مقابله‌ی اینترنت در فصل (۳) و نیز کدهای CRC فصل (۴) تأمین کننده‌ی این تعاریف می‌باشند. یک **تابع درهم‌ریزی رمز** نیازمند ویژگی اضافی ذیل نیز می‌باشد:

- از نقطه نظر محاسبه یافتن دو پیام گوناگون x و y به‌گونه‌ای که H(x)=H(y) باشد غیرممکن است.

1- Message integrity

2- Digital signitures and end point authentication.

3- Cryptographic hash functions

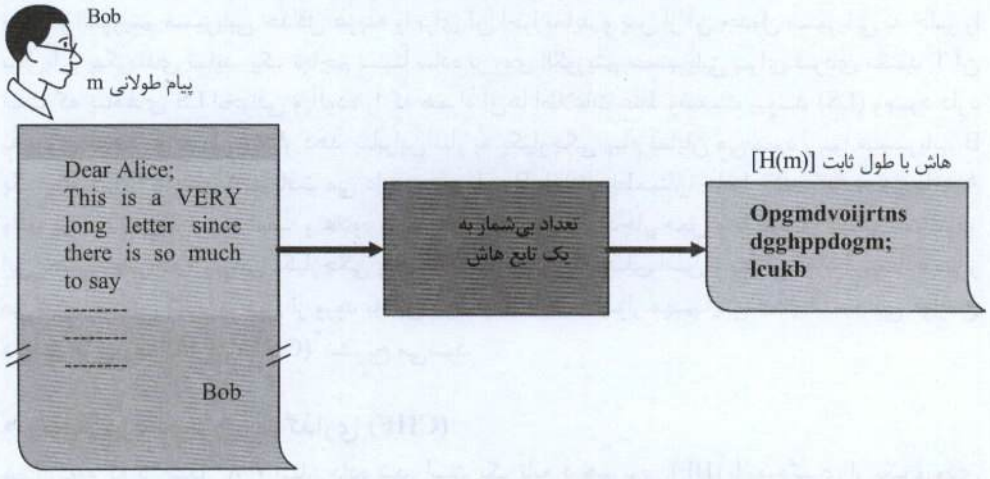
این ویژگی به مفهوم آن است که از لحاظ محاسبه برای یک مهاجم جایگزینی یک پیام به جای پیام دیگری که از سوی یک تابع درهم‌ریزی (HF) محافظت شده است غیرممکن می‌باشد. به بیان دیگر اگر (m, H(m)) به ترتیب پیام و پیام هاش تعریف شده از سوی فرستنده باشد آن‌گاه یک مهاجم قادر به جعل محتوای پیام دیگری مانند y که از ارزش هاش مشابه برخوردار است نسبت به پیام اصلی نمی‌باشد.

از آن‌جایی که یک کد مجموع مقابله مانند کد مجموع مقابله‌ی اینترنت باعث تعریف یک تابع درهم‌ریزی رمز (CHF) خواهد شد که بسیار ضعیف عمل می‌نماید، بنابراین به جای انجام عملیات مکمل‌گیری 1 مانند آن‌چه کد مجموع مقابله‌ی اینترنت انجام می‌دهد به محاسبه‌ی چنین کدی به این گونه می‌پردازیم که هر کاراکتر به عنوان یک بایت در نظر گرفته می‌شود و به آن بایت‌های دیگری در قالب گروه‌های 4 بایتی در هر زمان اضافه می‌گردد. به عنوان مثال اگر Bob مبلغ 100.99$ به Alice بدهکار باشد و یک پیام مانند IOU (به مفهوم من به شما بدهکارم) به Alice ارسال نماید آن‌گاه متن رشته‌ی ارسالی از سوی Bob به Alice چنین است:

"IOU 100.99 BOB"

معادل کدهای ASCII حروف و اعداد به کارگرفته شده بر مبنای سیستم هگزا برای این متن به صورت ذیل است:

49, 4F, 55, 31, 30, 30, 2E, 39, 39, 42, 4F, 42



شکل ۸-۷ ♦ توابع درهم‌ریزی (HF)

بخش بالایی شکل ۸-۸ تعداد 4 بایت کد مجموع مقابله برای این پیام را که به صورت B2، C1، D2، AC می باشد نشان می‌دهد. پیامی کاملاً متفاوت و پرهزینه برای Bob در بخش پائین شکل ۸-۸ نشان داده شده است. به عبارت دیگر هر دو پیام IOU 100.99 BOB و IOU 900.19 BOB دارای کد مجموع مقابله‌ی مشابه یکدیگر می‌باشند. با در اختیار داشتن داده‌ی اصلی یافتن مجموعه‌ی دیگری از داده‌ها با کد مجموع مقابله‌ی مشابه به سادگی امکان‌پذیر می‌باشد. بدیهی است به‌منظور اهداف امنیتی توابع درهم‌ریزی به مراتب قدرتمندتر از یک کد مجموع مقابله لازم می‌باشد.

Message	ASCII Representation				
I O U 1	49	4F	55	31	
0 0 . 9	30	30	2E	39	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Message	ASCII Representation				
I O U 9	49	4F	55	39	
0 0 . 1	30	30	2E	31	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

شکل ۸-۸ ♦ هر دو پیام اصلی و مجعول دارای یک کد مجموع مقابله می‌باشند!

الگوریتم درهم‌ریزی MD5 در مستند [RFC 1321] به صورت گسترده‌ای در حال حاضر مورد استفاده قرار می‌گیرد. این الگوریتم یک ارزش هاش 128 بیت را در یک فرآیند 4 مرحله‌ای شامل می‌شود که یک مرحله آن افزودن یک ارزش 1 و به دنبال آن تعدادی صفر در حد کفایت که طول پیام را برای شرایط ویژه تعریف نماید. یک مرحله نیز شامل کامل نمودن طول یک پیام 64 بیت و دو مرحله‌ی دیگر که به ترتیب مقداردهی اولیه یک رجیستر انباشت و اجرای یک حلقه که در آن توده‌های 16 کلمه‌ای در چهار دوره‌ی 4 تایی پردازش می‌شوند. الگوریتم مهم دیگر که از نوع هاش می‌باشد و در حال حاضر نیز به کار گرفته می‌شود **الگوریتم امن هاش (SHA-1)**[1] می‌باشد. این الگوریتم بر مبنای اصول مشابه طراحی شده در MD4 مستند [RFC1320] می‌باشد. الگوریتم MD5 جانشین الگوریتم MD4 است. این الگوریتم یک پیام 160 بیت را تولید می‌کند و هرچه طول خروجی بزرگ‌تر باشد SHA-1 دارای امنیت بیش‌تر خواهد بود.

۸-۶ کد اعتبارسنجی پیام

چالش مربوط به یکپارچگی پیام را می‌توان در قالب پیاده‌سازی آن چنین تصور نمود:

۱- پیام m از سوی Alice تعریف و تابع درهم‌ریزی آن یعنی H(m) با بهره‌گیری از الگوریتمی مانند (SHA-1) محاسبه می‌شود.

۲- پس از آن Alice تابع H(m) را به پیام m پیوند می‌دهد و در نتیجه یک پیام گسترش یافته‌ی (m,H(m)) را خلق و آن را به سوی Bob گسیل می‌کند.

۳- پس از دریافت پیام (m,h) از سوی Bob تابع H(m) محاسبه و اگر H(m)=h باشد آن‌گاه وی نتیجه‌گیری می‌کند که پیام دریافتی معتبر است.

این رویکرد خالی از عیب نمی‌باشد زیرا مهاجمی مانند T می‌تواند پیام مجعولی مانند m' را تعریف نماید که در آن اظهار می‌دارد وی Alice می‌باشد و بنابراین H(m') را محاسبه و پیامی به صورت

1- Secure hash algorithm-1

((m',H(m')) را به سوی Bob گسیل می‌کند. هنگامی که Bob پیام را دریافت می‌کند همه‌ی موارد برمبنای مرحله شماره‌ی 3 در بالا از اعتبار برخوردار می‌شود و بنابراین Bob پی به مجعول بودن پیام نمی‌برد. به‌منظور ارزیابی یکپارچگی پیام علاوه بر استفاده از توابع درهم‌ریزی رمز (CHF) هر دو موجودیت Alice و Bob نیازمند استفاده از یک رمز محرمانه‌ی s بین خود می‌باشند. این رمز محرمانه‌ی مشترک رشته‌ای از بیت می‌باشد که به آن **کلید اعتبارسنجی**[1] گفته می‌شود. با بهره‌گیری از این رمز مشترک ارزیابی یکپارچگی پیام را می‌توان در قالب مراحل ذیل به انجام رسانید.

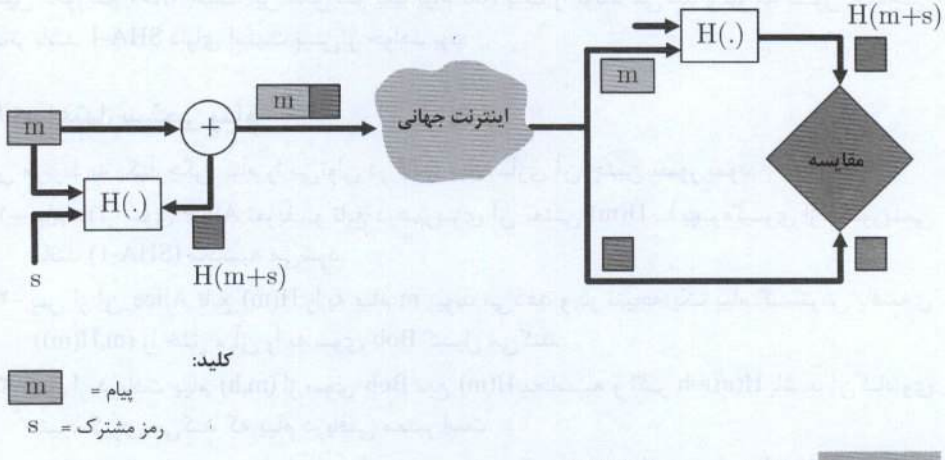
۱- پیام m از سوی Alice تعریف و به آن s افزوده می‌شود. نتیجه‌ی این عمل پیام m+s است که وی تابع هاش آن یعنی H(m+s) را با بهره‌گیری از الگوریتمی مانند SHA-1 محاسبه می‌کند. تابع محاسبه شده‌ی H(m+s) **کد اعتبارسنجی پیام (MAC)**[2] خوانده می‌شود.

۲- پس از آن Alice کُد اعتبارسنجی پیام MAC را به m اضافه می‌کند و در نتیجه پیام گسترش یافته‌ای مانند (m,H(m+s)) را خلق و آن را به سوی Bob ارسال می‌دارد.

۳- پس از دریافت پیام گسترش یافته‌ی (m,h) از سوی Bob، وی با آگاهی از رمز مشترک s اقدام به محاسبه‌ی MAC برای پیام H(m+s) می‌نماید در صورتی‌که H(m+s)=h باشد Bob اطمینان حاصل می‌کند که پیام دریافتی جعلی نمی‌باشد.

خلاصه‌ای از مراحل انجام شده در شکل ۸-۹ نشان داده شده است. بدیهی است خوانندگان بایستی بین MAC در این زیربخش و مشابه آن در پروتکل‌های لایه‌ی پیوند دچار اشتباه نشوند.

یکی از ویژگی‌های جالب MAC عدم نیاز آن به یک الگوریتم رمزگذاری می‌باشد. در بسیاری از کاربردها از جمله الگوریتم مسیریابی وضعیت پیوند (LSA) دو طرف ارتباط تنها نگران یکپارچگی پیام می‌باشند و محرمانه ماندن پیام برای آن‌ها چندان اهمیتی ندارد. با بهره‌گیری از MAC دو طرف ارتباط می‌توانند اعتبارسنجی پیام‌ها را انجام داده بدون آنکه مجبور به بهره‌گیری از الگوریتم‌های پیچیده‌ی رمزگذاری در فرآیند یکپارچگی پیام باشند.



شکل ۸-۹ ♦ کد اعتبارسنجی پیام (MAC)

1- Authentication key
2- Message authentication code

بدیهی است همان‌گونه که انتظار می‌رود تعدادی استاندارد گوناگون برای MAC در طول سال‌ها تعریف شده است. محبوب‌ترین این استانداردها در حال حاضر **HMAC** می‌باشد که از آن به‌همراه MD5 و یا SHA-1 استفاده می‌شود. استاندارد HMAC فرآیند پردازش داده و کلید اعتبارسنجی را از طریق دو مرتبه اجرای تابع هاش محاسبه می‌کند [RFC2014].

هنوز نکته‌ی مهم دیگری قابل تأمل است. چگونه کلید اعتبار سنجی مشترک بین طرفین تعامل توزیع می‌شود؟ به عنوان مثال در الگوریتم LSA به نوعی نیازمند توزیع کلید رمز اعتبارسنجی به هریک از مسیریاب‌ها در سیستم خودمختار هستیم. بدیهی است روترها می‌توانند همگی از کلید اعتبارسنجی مشابه بهره گیرند. یک مدیر شبکه در حقیقت می‌تواند به صورت فیزیکی با ملاقات هریک از مسیریاب‌ها این هدف را دستیابی نماید و یا آنکه برای هر روتر، کلید عمومی اختصاصی تعریف نماید و سپس کلید اعتبارسنجی را به هریک از روترها از طریق رمزگذاری با کلید عمومی آن مسیریاب انجام داده و سپس کلید رمز شده را از طریق شبکه به آن روتر گسیل نماید.

۸-۷ امضای دیجیتالی

تصور نمائید در طول هفته‌ی گذشته چندین مرتبه نام خود را در قالب امضا بر روی یک قطعه کاغذ مانند چک بانکی، رسید دریافت و یا پرداخت، اسناد حقوقی و غیره ثبت نموده‌اید. امضای شما بر روی هریک از این اسناد بیان کننده‌ی این واقعیت است که شما و نه فرد دیگری محتوای آن اسناد را تأئید و یا مورد موافقت قرار داده‌اید. در یک دنیای دیجیتالی اغلب موارد لازم است مالک و یا خالق یک سند، مشخص و تعریف شود و یا آنکه توافق آن فرد با محتوای یک سند آشکار شود. یک **امضای دیجیتالی (DS)**[1] یک روش رمزگذاری به‌منظور دستیابی به این اهداف در دنیای دیجیتالی است.

درست مانند امضاهای دست نویس امضای دیجیتالی بایستی به گونه‌ای انجام شود که قابل اعتبارسنجی و در عین حال غیرقابل جعل باشد. به بیان دیگر این امکان بایستی وجود داشته باشدکه یک سند امضا شده از سوی یک فرد قابل اثبات باشد وفقط آن فرد توان امضای سند را داشته باشد. دو موضوع اشاره شده به ترتیب به مفهوم اعتبارسنجی و غیرقابل جعل بودن امضای دیجیتالی است. اکنون طراحی یک امضای دیجیتالی و روش‌های آن توصیف می‌شود. آشکار است هنگامی که Bob پیامی را امضا می‌کند او بایستی اثری را برروی پیام قرار دهد که منحصر به وی می‌باشد. در این حالت Bob می‌تواند پیوست یک MAC را برای امضا مورد توجه قرار دهد که در آن MAC از طریق افزودن کلید او که منحصر به وی است به پیام مورد نظر پیوست و سپس گزینش هاش انجام پذیرد. از طرف دیگر برای Alice اعتبارسنجی امضا یکی از اهداف مهم به شمار می‌آید او بایستی نسخه‌ای از کلید که در این حالت منحصر به Bob نیست را در اختیار داشته باشد. بنابراین کدهای MAC قادر به انجام این وظیفه نیستند. یادآور می‌شود Bob هر دو کلید رمزگذاری عمومی و خصوصی را در اختیار دارد و این دو کلید در انحصار Bob می‌باشند بنابراین رمزگذاری با بهره‌گیری از کلید عمومی گزینش بسیار مناسبی برای تأمین اهداف امضای دیجیتالی به شمار می‌آید. در ادامه چگونگی پیاده‌سازی این مهم تحلیل می‌شود.

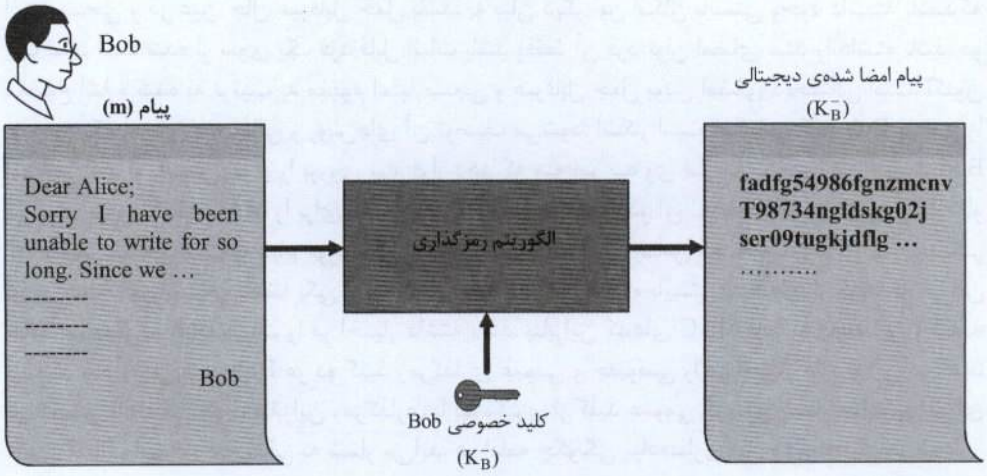
اگر فرض شود Bob مایل به امضای دیجیتالی یک سند m می‌باشد و منظور از سند در اینجا یک فایل یا یک پیامی است که Bob آن را امضا و سپس گسیل می‌کند آن‌گاه همان‌گونه در شکل ۸-۱۰ نشان داده شده است برای امضای این سند Bob از کلید خصوصی K_B^- خود بهره می‌گیرد تا تابع

1- Digital signature

$K_B^-(m)$ را محاسبه نماید. نخست ممکن است این موضوع به نظر غیرعادی جلوه نماید که Bob از کلید خصوصی خود بنابر مطالب زیربخش ۸-۲ به‌منظور رمزگشایی یک پیام رمز شده به همراه کلید عمومی استفاده می‌نمود، اکنون از آن جهت امضای یک سند استفاده نماید. اما یادآور می‌شود که موضوع رمزگذاری و رمزگشایی چیزی بیش‌تر از یک عملیات ریاضی نمی‌باشد، آن هم محاسبه‌ی توان e و یا d از یک ارزش در الگوریتم RSA و نیز یادآوری این نکته هم ضروری است هدفی که Bob دنبال می‌کند رمزگذاری و یا پیچیده نمودن محتوای پیام نمی‌باشد بلکه امضای آن سند به گونه‌ای است که توان‌مند از نقطه‌نظر اعتبارسنجی و غیرقابل جعل باشد. بنابراین امضای دیجیتالی Bob برای یک سند همان تابع $K_B^-(m)$ می‌باشد.

پرسش مهم آن است که آیا امضای دیجیتالی $K_B^-(m)$ نیازهای مربوط به اعتبارسنجی و عدم قابلیت جعل را تأمین می‌نماید؟ فرض کنید Alice پیام m و تابع $K_B^-(m)$ را در اختیار دارد و او مایل است در دادگاه اثبات نماید که Bob سند را امضا نموده است و او تنها فردی است که می‌تواند آن را امضای نموده باشد. در این حالت Alice کلید عمومی Bob یعنی K_B^+ را به امضای دیجیتالی وی یعنی تابع $K_B^-(m)$ وابسته به سند m اعمال می‌کند. به عبارت دیگر او ارزش $K_B^+(K_B^-(m))$ را محاسبه می‌کند و پس از انجام این مهم پیام یا سند m را تولید می‌کندکه دقیقاً قابل نگاشت با سند اصلی است. پس از آن Alice این نتیجه‌گیری را ارایه می‌دهد که فقط Bob قادر به امضای سند بوده است و دلایل ذیل را برای اثبات نظریه‌ی خود ارایه می‌دهد:

۱- هر فردی که پیام را امضا نموده است بایستی کلید خصوصی K_B^- را در اختیار داشته باشد که با بهره‌گیری از آن امضای دیجیتالی $K_B^-(m)$ را به‌گونه‌ای محاسبه نماید که نتیجه: $K_B^+(K_B^-(m)) = m$ باشد.



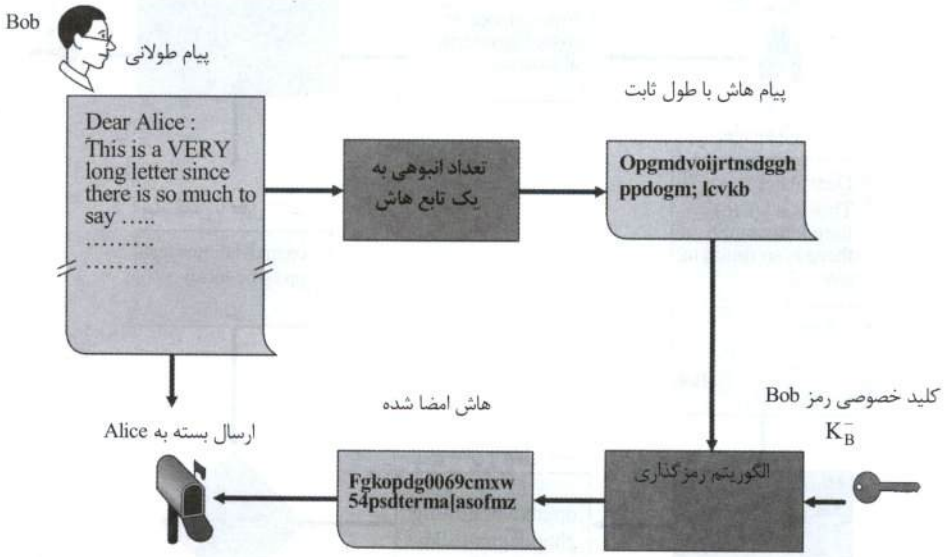
شکل ۸-۱۰ ♦ تعریف و خلق یک امضای دیجیتالی برای یک سند

۲- تنها فردی که آگاهی از کلید خصوصی K_B^- داشته است Bob بوده است زیرا از مباحث مربوط به RSA در بخش ۸-۲ می‌توان دریافت که آگاهی از کلید عمومی K_B^+ هیچ‌گونه کمکی را در

یادگیری کلید خصوصی K_B^- به‌همراه ندارد. در نتیجه تنها فردی که از کلید K_B^- آگاهی دارد همان شخصی است که زوج کلید (K_B^+, K_B^-) را نخستین مرتبه تعریف نموده است یعنی Bob. بدیهی است فرض بر آن است که کلید K_B^- را شخص Bob در اختیار هیچ فردی قرار نداده و یا آنکه کلید از وی به سرقت نرفته است.

توجه به این نکته ضروری است اگر در سند اصلی m به نوعی تغییر حاصل شود به گونه‌ای که بتوان جایگزین آن سندی مانند 'm را نمود آن‌گاه امضای دیجیتالی Bob برای m دیگر برای 'm معتبر نمی‌باشد، زیرا $K_B^+(K_B^-(m))$ برابر 'm نمی‌باشد و در نتیجه ملاحظه می‌شود امضای دیجیتالی موضوع تأمین یکپارچگی پیام را نیز تأمین می‌کند و این امکان را برای گیرنده فراهم می‌کند که عدم ایجاد تغییر در محتوای پیام را به همراه منبع پیام اعتبارسنجی نماید.

یک نگرانی از نقطه نظر امضای داده‌ها رمزگذاری آن است که از لحاظ محاسبه و پردازش عوامل رمزگذاری و رمزگشایی پرهزینه می‌باشد. با درنظر گرفتن سرآیندهای رمزگذاری و رمزگشایی امضای داده‌ها از طریق این عملیات می‌تواند دشوار و غیرممکن باشد. یک رویکرد کارآمد اعمال توابع درهم‌ریزی به درون امضای دیجیتالی است. با یادآوری این نکته از زیربخش ۸-۶ که یک الگوریتم هاش پیام m با طول متغیر تصادفی را گزینش و به محاسبه‌ی یک طول ثابت از پیام می‌پردازد که در قالب H(m) تعریف می‌شود. به عبارتی دیگر **اثر انگشت** آن را به دست می‌آورد، در نتیجه با بهره‌گیری از یک تابع درهم‌ریزی Bob پیام درهم آمیزی شده را به جای پیام اصلی امضا می‌کند و به عبارت دیگر تابع $K_B^-(H(m))$ را محاسبه می‌کند. از آنجایی که H(m) به‌طور کلی بسیار کوچک‌تر از پیام اصلی m است تلاش محاسباتی به‌منظور خلق امضای دیجیتالی به صورت چشمگیری کاهش می‌یابد.

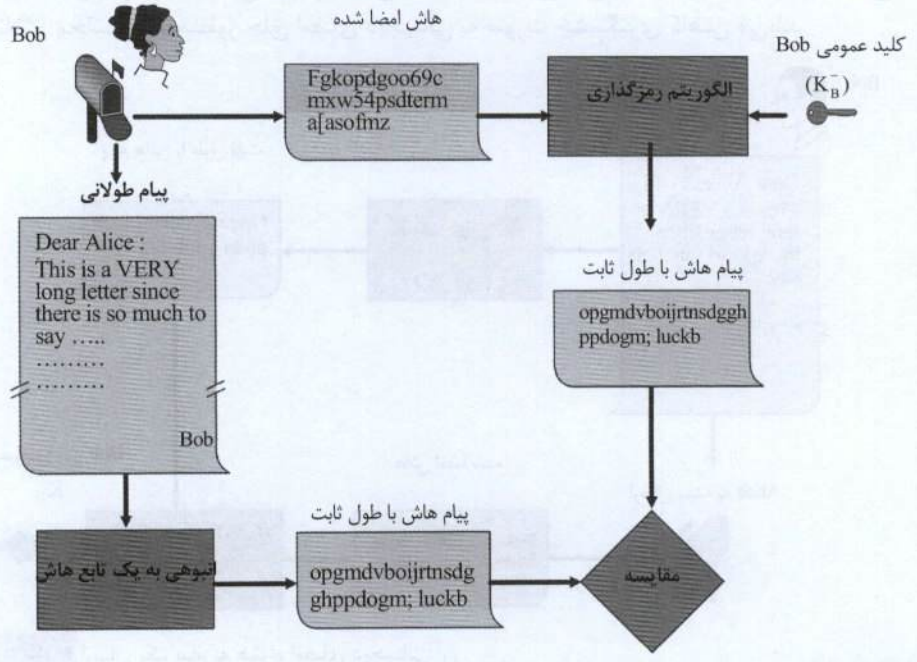


شکل ۸-۱۱ ♦ ارسال یک پیام به همراه امضای دیجیتالی آن

در چارچوب ارسال پیام از سوی Bob به Alice شکل ۸-۱۱ خلاصه‌ای از مراحل عملیاتی تولید یک امضای دیجیتالی را نشان می‌دهد. پس از آنکه Bob پیام طولانی و اصلی خود را درون یک تابع هاش

قرار داد نسبت به امضای دیجیتالی با بهره‌گیری از کلید خصوصی خود اقدام می‌کند. پیام اصلی در متن خام به همراه امضای دیجیتالی پیام به سوی Alice ارسال می‌شود. شکل ۸-۱۲ خلاصه‌ای از مراحل عملیاتی از امضا را نشان می‌دهد. با اعمال کلید عمومی فرستنده به پیام Alice نتیجه یک هاش را به دست می‌آورد. اگر دو مقدار هاش قابل نگاشت با یکدیگر باشند آن‌گاه Alice از یکپارچگی و اعتبار نویسنده‌ی پیام اطمینان حاصل می‌کند.

قبل از ادامه‌ی مطلب، مقایسه‌ای بین MAC و امضای دیجیتالی انجام می‌شود. اگرچه این دو در مواردی موازی و مشابه یکدیگرند اما تفاوت‌های بنیادی مهمی را نیز شامل می‌شوند. هر دو امضاهای دیجیتالی و MAC با یک پیام یا سند آغاز می‌شوند. به‌منظور خلق یک MAC از یک پیام، یک کلید اعتبارسنجی به پیام افزوده می‌شود و سپس هاش نتیجه شده گزینش می‌شود. بدیهی است کلیدهای رمز عمومی و متقارن هیچ کدام نقشی در خلق MAC بازی نمی‌کنند. از طرف دیگر برای تولید امضای دیجیتالی نخست تابع هاش پیام را گزینش و سپس پیام با کلید خصوصی از طریق بهره‌گیری از کلید عمومی رمزگذاری می‌شود. بنابراین یک امضای دیجیتالی از روش پیچیده‌تری در مقایسه با MAC برخوردار است زیرا نیازمند **زیرساخت کلید عمومی (PKI)**[1] به همراه گواهی معتبر بر مبنای آن‌چه در ذیل توصیف شده است می‌باشد. در زیربخش ۸-۱۰ ملاحظه خواهد شد PGP به عنوان یک سیستم امن ایمیل از امضاهای دیجیتالی به‌منظور یکپارچگی پیام بهره می‌گیرد. قبلاً ملاحظه گردید پروتکلی مانند OSPF از MAC به‌منظور یکپارچگی پیام بهره می‌گرفت. در بخش‌های ۸-۱۱ و ۸-۱۳ ملاحظه خواهد شد که از MAC هم‌چنین برای پروتکل‌های امنیتی لایه‌های انتقال و شبکه نیز استفاده می‌شود.



شکل ۸-۱۲ ♦ اعتبارسنجی یک پیام امضا شده‌ی دیجیتالی

1- Public key infrastructure

۸-۸ گواهی کلید عمومی

یکی از مهم‌ترین کاربردهای امضاهای دیجیتالی **گواهی کلید عمومی**[1] است که مفهوم آن عبارت است از گواهی مالکیت یک کلید عمومی به یک موجودیت ویژه. گواهی کلید عمومی در بسیاری از پروتکل‌های محبوب امن شبکه از جمله IP_{sec} و SSL کاربرد گسترده دارد. برای درک عمیق موضوع و نیاز به گواهی کلید عمومی، نوعی تجارت اینترنتی را در ذهن خود تصور نمائید: فرض کنید Alice در کسب و کار تحویل پیتزا از طریق دریافت سفارش برروی اینترنت مشغول می‌باشد. یکی از مشتریان وی یعنی Bob به دلیل علاقه‌مندی فراوان به پیتزا مرتباً سفارش‌های خود را از طریق ارسال یک پیام خام که شامل آدرس سکونت وی و نوع پیتزای درخواستی است به سوی Alice گسیل می‌کند. در این پیام Bob هم‌چنین یک امضای دیجیتالی که یک امضای هاش بر روی متن ساده‌ی پیام است را به‌منظور اثبات واقعی بودن منبع پیام برای Alice اضافه می‌نماید. به‌منظور اعتبارسنجی امضای دیجیتالی، Alice کلید عمومی Bob را از طریق یک سرور کلید عمومی و یا پیام ایمیل دریافت و امضای دیجیتالی را اعتبارسنجی می‌کند. در این رویکرد او اطمینان می‌یابد که شخص Bob پیام را ارسال نموده است و فرد دیگری خود را به جای Bob قرار نداده است.

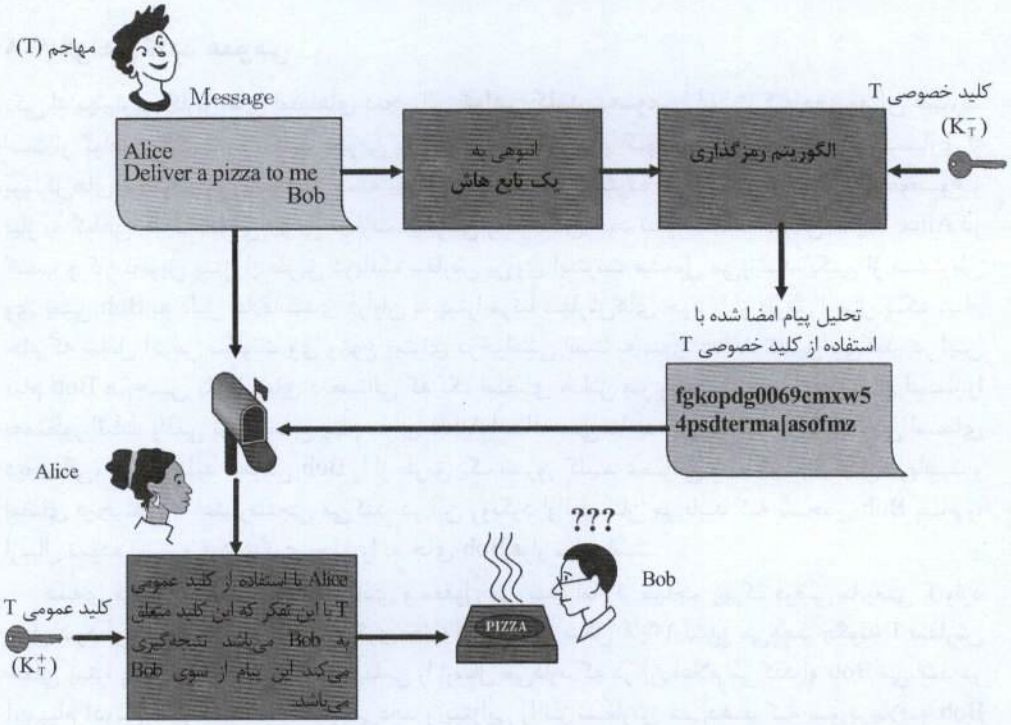
همه‌ی موارد اشاره شده به نظر عادی و معقول می‌رسند اما اگر مهاجم زیرک فرضی ما یعنی T وارد عمل شود آن‌گاه داستان به‌صورت دیگری جلوه‌گر می‌شود. شکل ۸-۱۳ نشان می‌دهد چگونه T سفارش جعلی پیتزا را به Alice می‌دهد. وی پیامی را ارسال می‌دارد که در آن اعلام می‌کند او Bob می‌باشد در آن پیام آدرس سکونت Bob را قرار می‌دهد و پیتزایی را نیز سفارش می‌دهد که مورد علاقه Bob نیست. در پیام ارسالی T هم‌چنین کلید عمومی خود را اضافه می‌کند و Alice نیز به شکل طبیعی فرض می‌کند این کلید عمومی Bob می‌باشد. علاوه بر آن T یک امضای دیجیتالی نیز به پیام می‌افزاید. پس از دریافت پیام، Alice کلید عمومی T را به امضای دیجیتالی با تصور آنکه کلید متعلق به Bob می‌باشد، اعمال می‌کند و به این نتیجه می‌رسد که پیام متن خام از جانب Bob ارسال شده است. از طرف دیگر Bob با تعجب بسیار با تحویل پیتزای پپرونی که مورد علاقه‌ی وی نمی‌باشد در محل سکونت خود روبه‌رو می‌شود!

از این مثال طنزگونه ملاحظه می‌شود به‌منظور سودمندی کلید رمز عمومی نیازمند نوعی اعتبارسنجی می‌باشیم تا اطمینان حاصل شود کلید عمومی واقعی یک موجودیت (فرد، مسیریاب، مرورگر و غیره) برای ارتباط در اختیار می‌باشد. به عنوان مثال هنگامی که Alice مایل به تعامل با Bob با بهره‌گیری از کلید عمومی رمز است، نیازمند اعتبارسنجی کلید عمومی می‌باشدکه به نظر می‌رسد Bob ارسال نموده است.

ارتباط یک کلید عمومی به یک موجودیت ویژه از طریق یک **مقام گواهی کننده (CA)**[2] انجام می‌گیرد که وظیفه‌ی وی تأئید هویت شناسه‌ها و صدور گواهی آن‌هاست. یک CA مسئولیت‌های ذیل را برعهده دارد:

1- Public key certification

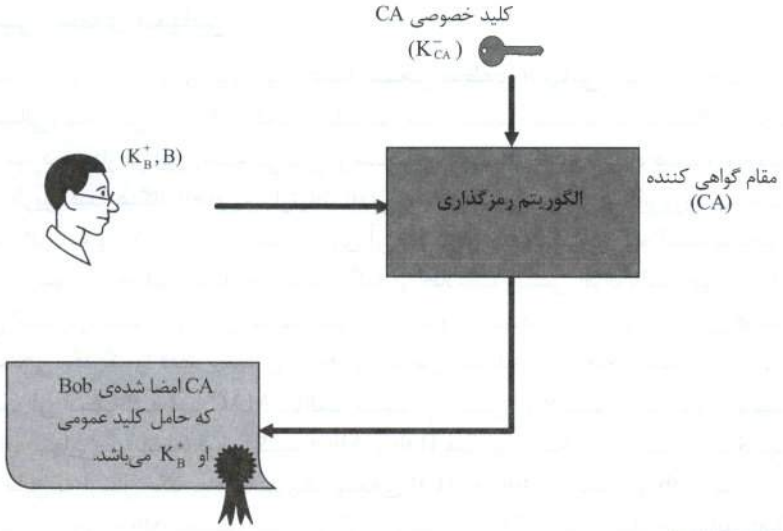
2- Certification authority



شکل ۸-۱۳ ♦ مهاجم T خود را به جای Bob با بهره‌گیری از کلید رمز عمومی معرفی می‌کند.

۱- یک CA گواهی کننده‌ی هویت یک موجودیت مانند فرد، روتر، مرورگر و غیره می‌باشد. هیچ رویکرد اجباری تعریف شده‌ای از لحاظ چگونگی انجام گواهی وجود ندارد. هنگامی که از CA استفاده می‌شود بایستی مطمئن از این حقیقت بود که CA تمامی عملیات اعتبارسنجی و بازرسی‌های لازم را انجام داده است. به عنوان مثال اگر T قادر به وارد شدن به حریم یک CA که موقتی و کوتاه عمر است، باشد و به سادگی اعلام نماید "I am alice" و سپس گواهی وابسته به شناسایی Alice را دریافت دارد آن‌گاه اعتماد به کلیدهای عمومی گواهی شده از سوی چنین CAهایی از میان می‌رود. از طرف دیگر یک کاربر ممکن است اعتماد بیش‌تری را نسبت به CAهایی داشته باشد که بخشی از برنامه‌های امنیتی دولت در کشورش می‌باشد. بدیهی است فرد ممکن است به این CAها نیز مشکوک باشد. بنابراین گواهی کلید عمومی از سوی یک CA تا آن‌جایی قابل اعتماد است که روش‌های اعتبارسنجی شناسه و CA از نقطه نظر کاربر قابل قبول باشد. ملاحظه می‌شود در شبکه‌ای کاملاً غیرقابل اعتماد سرگردان هستیم!

۲- هنگامی که CA شناسه‌ای را از یک موجودیت تأئید نماید آن CA یک **گواهی** خلق می‌کند که کلید عمومی آن موجودیت را به شناسه‌ی آن ارتباط می‌دهد. این گواهی حامل کلید عمومی واطلاعات شناسایی منحصری است که جنبه‌ای فراگیر در ارتباط با مالک آن کلید عمومی مانند نام شخص و یا آدرس IP آن دارد. بدیهی است این گواهی از سوی CA امضای دیجیتالی می‌شود. شکل ۸-۱۴ این مراحل را به نمایش گذارده است.



شکل ۸-۱۴ ♦ گواهی کلید عمومی Bob از سوی CA

اکنون به بررسی چگونگی مقابله‌ی گواهی‌ها با سفارش دهنده‌ی جعلی مثال پیتزا می‌پردازیم. هنگامی که Bob سفارش خود را ثبت می‌کند او هم‌چنین گواهی CA امضا شده را نیز ارسال می‌دارد. گواهی CA مربوط به کلید عمومی است و از سوی Alice مورد بهره‌برداری قرار می‌گیرد تا اعتبار گواهی Bob و سپس اخذ کلید عمومی او از سوی Alice انجام پذیرد. هر دو نهاد بین‌المللی اتحادیه‌ی بین‌المللی ارتباطات (ITU) و IETF استانداردهایی را برای گواهی‌های CA تعریف نموده‌اند. مستند ITUX.509 خدمات اعتبارسنجی و نیز قواعد مربوط به تعریف گواهی‌ها را تعریف می‌نماید در حالیکه مستند [RFC 1422] به توصیف مدیریت کلیدها برمبنای CA برای کاربرد در ایمیل‌های امن اینترنتی پرداخته است. این استاندارد کاملاً سازگار با X.509 می باشد اما به مراتب فراتر از آن رفته است. زیرا مراحل ایجاد و علامت‌گذاری را نیز برای معماری مدیریت یک کلید تعریف می‌نماید. جدول ۸-۴ برخی از دامنه‌های مهم موجود در یک گواهی را توصیف می‌کند.

جدول ۸-۴♦ دامنه‌های گزینش شده در یک X.509 و کلید عمومی RFC1422

نام میدان یا دامنه	شرح
نسخه	شماره‌ی نسخه‌ی مشخصات X.509
شماره سریال	شناسه‌ی انحصاری صادر شده از سوی CA برای گواهی
امضا	مشخصات الگوریتم به‌کار گرفته از سوی CA به‌منظور گواهی امضا
نام صادر کننده‌ی گواهی	شناسه‌ی صادر کننده‌ی گواهی در قالب نام قابل تمایز برمبنای مستند [RFC2253] نام‌های (DN)
مدت اعتبار	طول دوره‌ی زمانی اعتبار تاریخ شروع و پایان
عنوان موضوع	شناسه‌ی موجودیت که کلید عمومی وابسته به گواهی در قالب DN را مشخص می‌کند
موضوع کلید عمومی	موضوع کلید عمومی و نماد الگوریتم کلید عمومی به‌همراه پارامترهایی که در کلید استفاده شده است را مشخص می‌کند

۸-۹ اعتبارسنجی نقطه‌ی انتهایی

فرآیند اثبات هویت یک موجودیت برای فرد دیگر **اعتبارسنجی نقطه‌ی انتهایی** نامیده شده است. به عنوان موجودات انسانی، به روش‌های گوناگونی عملیات اعتبارسنجی نسبت به یکدیگر را انجام می‌دهیم. از جمله موارد آشکار در این زمینه می‌توان از شناسایی هویت از طریق چهره هنگام ملاقات، شناسایی هویت از طریق صدا هنگام انجام یک ارتباط تلفنی و یا شناسایی هویت هنگام ورود به خاک یک کشور از طریق گذرنامه و عکس پیوست شده برروی آن نام برد. اما هنگامی که اعتبارسنجی از طریق شبکه انجام می‌شود دو طرف ارتباط نمی‌توانند تکیه بر اطلاعات زیستی افراد مانند چهره و یا اثر صدا نمایند. بدیهی است در اغلب موارد این عناصر شبکه روترها و فرآیندهای سرور و مشتری هستند که بایستی اعتبارسنجی یکدیگر را فقط برمبنای پیام‌ها و داده‌های مبادله شده انجام دهند.

پرسش نخست آن است که آیا MAC مطالعه شده در بخش ۸-۶ قابل استفاده به‌منظور اعتبارسنجی نقطه‌ی انتهایی می‌باشد؟ فرض کنید Alice و Bob هر دو از یک رمز مشترک S بهره می‌گیرند و Alice مایل به ارسال یک پیام مانند یک قطعه‌ی TCP به Bob می‌باشد و Bob نیز مایل است از پیام دریافتی از سوی Alice مطمئن شود. رویکرد برمبنای MAC اینگونه عمل می‌نماید: Alice یک MAC را با بهره‌گیری از پیام و رمز مشترک تولید و MAC را به آن پیام پیوست می‌نماید و سپس نتیجه‌ی حاصل از این عملیات را که یک پیام توسعه یافته است به سوی Bob گسیل می‌دارد. هنگامی که Bob پیام گسترش یافته را دریافت می‌نماید او از MAC درون آن به‌منظور اعتبارسنجی هر دو مورد مبدأ و یکپارچگی پیام استفاده می‌کند. از آن‌جایی که فقط Alice و Bob از رمز مشترک بین خود آگاهی دارند اگر محاسبه‌ی MAC انجام شده از سوی Bob نتیجه‌ای را تأمین نماید که مشابه MAC حاضر در پیام گسترش یافته است آن‌گاه Bob مطمئن می‌شود که Alice پیام را ارسال داشته است و این پیام نیز در مسیر انتقال، جعل و یا دستکاری نشده است. در حقیقت Bob اطمینان 100% از منبع پیام ندارد زیرا این احتمال وجود دارد که وی از تهاجمی دچار گمراهی شود که به آن **تهاجم بازپخش**[1] گفته می‌شود. در شکل ۸-۱۵ نشان داده شده است، کافی است T به نوعی پیام گسترش یافته‌ی Alice را شنود و سپس ثبت نماید و آن را مجدداً در فرصت زمانی دیگری بازپخش نماید. پیام تکرار شده می‌تواند چنین عبارتی باشد:

"It is ok to transfer one million dollars from Bill's to T's account"

این موضوع سبب می‌شود مبلغ دو میلیون دلار انتقال یابد و یا آنکه مضمون پیام می‌تواند این گونه باشد:

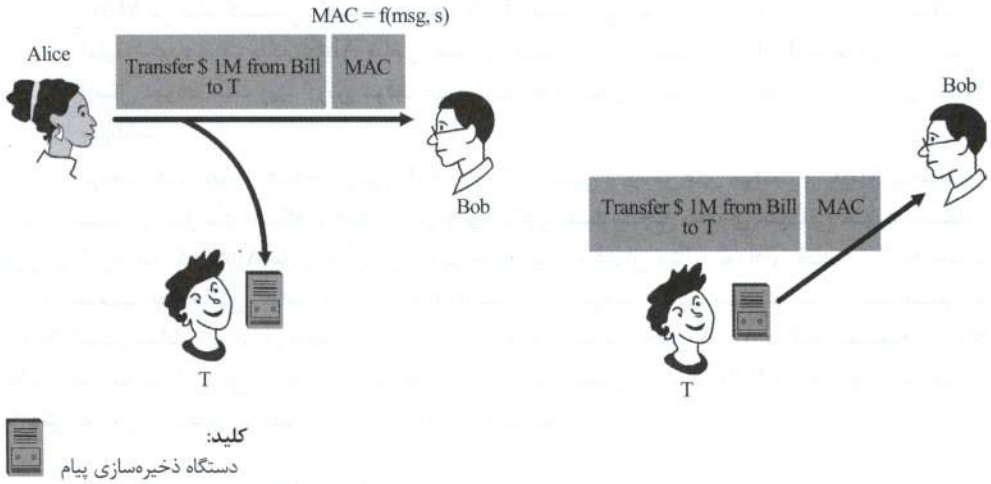
"The link from router Alice to router Charlie has gone done"

در صورت ارسال این پیام پس از آنکه پیوند تعمیر گردد باعث بروز خطا در پیکربندی جداول گسیل جلو خواهد شد.

سناریوی تخریب شکل ۸-۱۵ به این دلیل روی داده است که Bob قادر به تمایز بین پیام اصلی ارسالی از سوی Alice و بازپخش مجدد آن در فرصت دیگری نبوده است. به بیان دیگر Bob نمی‌توانسته تشخیص دهد که آیا Alice ارسال زنده‌ی پیام نموده است و یا آنکه پیام دریافتی نتیجه‌ی بازپخش یک پیام ثبت شده‌ی قبلی است. خواننده‌ی زیرک به یاد می‌آورد که عملیات دست‌دهی سه

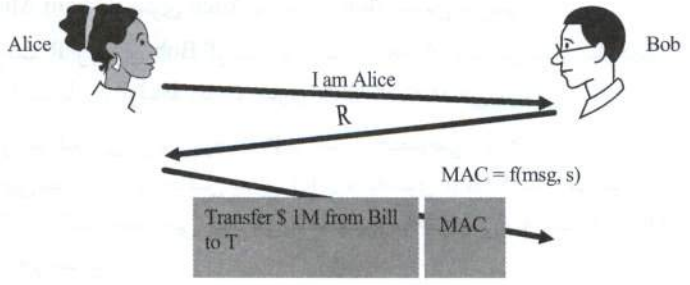
1- Playback attack

جانبه‌ی پروتکل TCP به‌منظور برطرف نمودن همین چالش تعریف شده است. زیرا سرور طرف ارتباط TCP مایل به قبول ارتباط نمی‌شود اگر قطعه‌ی SYN دریافتی نسخه‌ی قدیمی و به عبارت دیگر انتقال مجدد یک قطعه SYN از ارتباط قبلی می‌باشد. سرور TCP چگونه به این تشخیص می‌رسید که برنامه‌ی مشتری دارای ارتباط زنده است؟ با گزینش یک شماره ترتیب اولیه‌ای که مدتی طولانی از استفاده‌ی آن گذشته است. با ارسال آن شماره به مشتری و سپس انتظار پاسخ از طریق یک قطعه‌ی ACK که حامل آن شماره از سوی برنامه‌ی مشتری است این مهم را به سرانجام مطلوب می‌رسانید. از همین روش نیز می‌توان به‌منظور تأمین اهداف اعتبارسنجی بهره گرفت.



شکل ۸-۱۵♦ تهاجم از نوع بازپخش پیام

شماره‌ای که یک پروتکل فقط یک مرتبه در طول عمر زمانی خود مورد استفاده قرار می‌دهد **شماره ردیف موقت**[1] نامیده می‌شود. به عبارت دیگر هنگامی که یک پروتکل از یک شماره موقتاً بهره می‌گیرد آن پروتکل هرگز استفاده‌ی مجدد از آن شماره را نخواهد نمود. شکل ۸-۱۶ پروتکل بهره‌گیر از یک شماره‌ی موقت را نشان می‌دهد.



شکل ۸-۱۶♦ دفاع در برابر تهاجم بازپخش با بهره‌گیری از شماره‌ی موقت R

1- Nonces

بهره‌گیری از یک شماره‌ی موقت برمبنای شکل ۸-۱۶ به‌گونه‌ی ذیل است:

۱- با گزینش شماره‌ای موقت به نام R از سوی Bob و ارسال آن به Alice در قالب متن ساده یا شفاف، Alice به تعریف MAC با بهره‌گیری از پیام اصلی، رمز مشترک S و شماره موقت R می‌پردازد. برای تعریف MAC کافی است Alice دو بخش R و رمز مشترک S را به پیام پیوست و نتیجه را از طریق هاش اجرا نماید. با پیوست MAC به پیام اصلی Alice پیام گسترش یافته‌ای را خلق می‌کند که به سوی Bob گسیل می‌شود.

۲- از طریق پیام دریافتی Bob به محاسبه‌ی MAC، S و R می‌پردازد. اگر MAC حاصل برابر MAC در پیام گسترش یافته باشد آنگاه Bob مطمئن می‌شود که نه تنها Alice پیام ارسالی را تولید نموده است بلکه Alice پیام را هنگامی تولید کرده است که Bob شماره‌ی موقت را ارسال نموده است زیرا ارزش موقت مورد نیاز محاسبه‌ی درست و بدون خطا برای MAC می‌باشد.

در زیربخش‌های بعدی هنگام بررسی SSL، و IP_{sec} ترکیبی از ارزش‌های موقت و MAC به‌منظور تأمین امنیت پروتکل‌های شبکه با هدف یکپارچگی پیام و اعتبارسنجی نقطه‌ی انتهایی مورد استفاده قرار می‌گیرد. اما اگر Alice مایل به ارسال مجموعه‌ای پیام به عنوان مثال، تعدادی قطعه‌ی TCP باشد آن‌گاه وضعیت چه حالتی خواهد داشت؟ آیا Bob یک ارزش موقت جدید بایستی برای هر پیام به Alice گسیل نماید؟ در عمل فقط یک ارزش موقت لازم است. در بخش ۸-۱۱ هنگام مطالعه‌ی SSL یک ارزش موقت با ردیفی از اعداد ترکیب می‌شود که این امکان را برای Bob فراهم آورد تا اعتبار تازه‌گی همه‌ی پیام‌های دریافتی از سوی Alice را تأئید نماید.

• اعتبارسنجی با رمزگذاری کلید عمومی

استفاده از یک ارزش موقت و رمز مشترک، بنیادهای یک پروتکل اعتبارسنجی موفق به‌شمار می‌آید. پرسش طبیعی آن است که آیا می‌توان از یک ارزش موقت R و رمزگذاری کلید عمومی به‌عنوان راهکار چالش اعتبارسنجی بهره گرفت. بهره‌گیری از رویکرد کلید عمومی دشواری‌های موجود در هر سیستم رمز مشترک را پنهان می‌کند و به نگرانی یادگیری ارزش رمز مشترک از سوی دو طرف ارتباط پایان می‌دهد. پروتکل کاربردی رمزگذاری کلید عمومی در قالب گام‌های ذیل عمل می‌کند:

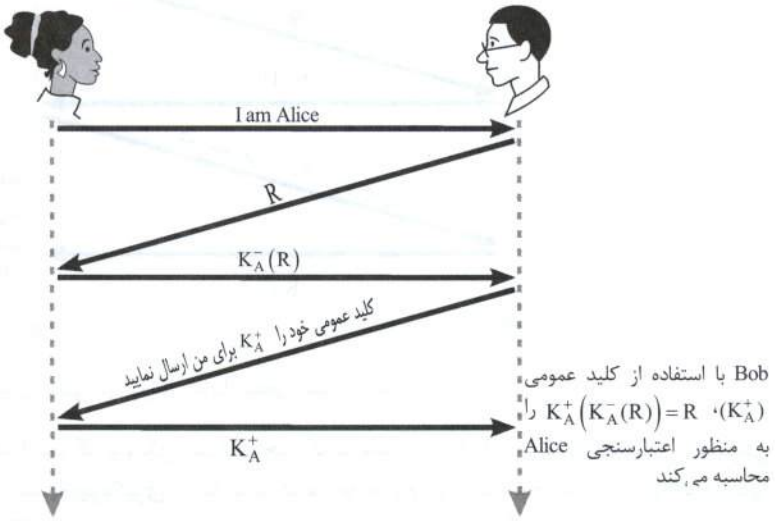
۱- پیام "I am Alice" از سوی Alice به سمت Bob گسیل می‌شود.

۲- ارزش موقت R از سوی Bob گزینش و به سوی Alice روانه می‌شود. بار دیگر از ارزش موقت به‌منظور اطمینان از ارتباط زنده و جاری Alice استفاده می‌شود.

۳- به‌منظور رمزگذاری ارزش موقت R از کلید خصوصی K_A^- متعلق به Alice از سوی وی استفاده می‌شود و ارزش حاصل یعنی $K_A^-(R)$ به‌سوی Bob گسیل می‌شود. از آن‌جائیکه فقط Alice آگاه از کلید خصوصی خود می‌باشد هیچ موجودیتی به غیر از Alice قادر به تولید $K_A^-(R)$ نخواهد بود.

۴- کلید عمومی K_A^+ متعلق به Alice از سوی Bob به پیام دریافتی اعمال می‌شود، به بیان دیگر Bob به محاسبه‌ی ارزش $K_A^+(K_A^-(R))$ می‌پردازد. بدیهی است از مباحث رمزگذاری کلید عمومی RSA در بخش قبل ارزش $K_A^+(K_A^-(R)) = R$ می‌شود و بنابراین Bob محاسبه‌ی R و اعتبارسنجی Alice را انجام می‌دهد.

عملیات مربوط به این پروتکل در شکل ۸-۱۷ نشان داده شده است. آیا این یک پروتکل امن است؟ از آن‌جا که از روش‌های کلید عمومی استفاده می‌شود این پروتکل نیازمند آن است که Bob کلید عمومی متعلق به Alice را دریافت نماید. این موضوع سناریوی جالبی را در شکل ۸-۱۸ نشان داده است که در آن مهاجم T فرصت جایگزینی خود به جای Alice و معرفی به Bob را پیدا می‌کند.



شکل ۸-۱۷ ♦ کارکرد درست پروتکل اعتبارسنجی کلید عمومی

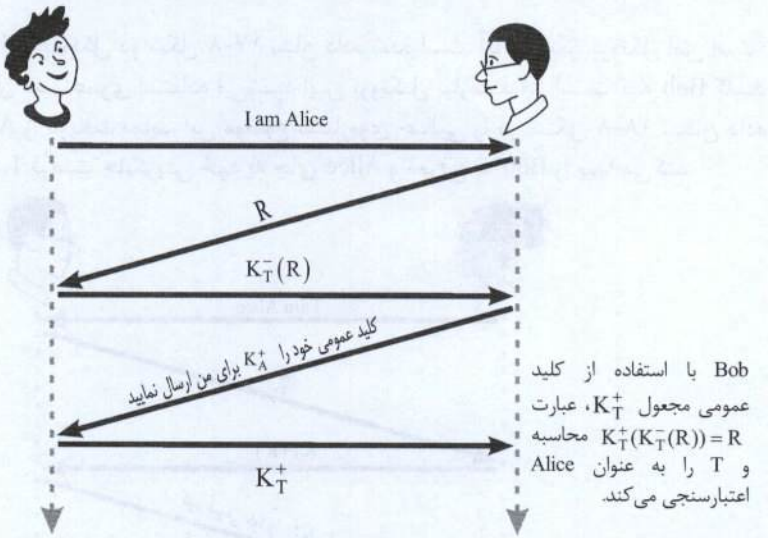
در ذیل گام‌های مربوط به تهاجم T تشریح می‌شود:

۱- پیام "I am Alice" از سوی T به Bob گسیل می‌شود.

۲- ارزش موقت R از سوی Bob گزینش و به سوی Alice روانه می‌شود اما پیام در بین مسیر از سوی T شنود می‌شود.

۳- با بهره‌گیری از کلید خصوصی K_T^- از سوی T ارزش موقت R رمزگذاری و نتیجه‌ی حاصل یعنی $K_T^-(R)$ به سوی Bob ارسال می‌شود. برای شخص Bob این ارزش $K_T^-(R)$ تعدادی بیت به شمار می‌آید و او از این موضوع آگاهی ندارد که این بیت‌ها به مفهوم $K_T^-(R)$ و یا $K_A^-(R)$ می‌باشند.

۴- در این مقطع Bob لازم است کلید عمومی Alice یعنی K_A^+ را به دست آورده و سپس آن را به ارزشی که به تازه‌گی دریافت نموده است اعمال نماید. بنابراین Bob پیامی را به سوی Alice گسیل و از وی می‌خواهد که کلید عمومی خود یعنی K_A^+ را برای وی ارسال نماید. ممکن است Bob کلید عمومی Alice را از طریق اینترنت بازیابی نماید. مهاجم T بار دیگر این پیام را شنود می‌کند و پاسخ Bob را با کلید عمومی خود یعنی K_T^+ می‌دهد. با محاسبه‌ی: $K_T^+(K_T^-(R)) = R$ از سوی Bob او باور می‌کند که T همان Alice است.



شکل ۸-۱۸ ♦ یک حفره‌ی امنیتی در پروتکل اعتبارسنجی کلید عمومی

از این سناریو آشکار است که پروتکل اعتبارسنجی کلید عمومی به اندازه‌ی توزیع کلیدهای عمومی ایمن می‌باشد. بدیهی است با بهره‌گیری از مکانیزم گواهی‌ها توزیع امن کلیدهای عمومی نیز امکان‌پذیر می‌شود. در سناریوی شکل ۸-۱۸ هر دو فرد Bob و Alice سرانجام همراه یکدیگر ممکن است قادر به کشف اختلال انجام شده باشند زیرا هنگامی که Bob ادعا می‌کند با Alice در تعامل بوده است موجودیت Alice آگاه است که هرگز با Bob در ارتباط نبوده است. نوع دیگری از تهاجم نیز وجود دارد که بسیار زیرکانه است و مانع از تشخیص و آشکارسازی می‌شود. در سناریوی شکل ۸-۱۹ هر دو فرد Alice و Bob در تعامل با یکدیگر می‌باشند در حالیکه T با بهره‌گیری از همان حفره یا ضعف امنیتی در پروتکل اعتبارسنجی خود را بین Alice و Bob قرار می‌دهد. به ویژه هنگامی که Bob با بهره‌گیری از کلید رمز دریافتی از سوی T آغاز به ارسال داده‌های رمزگذاری شده به Alice می‌نماید. مهاجم T می‌تواند متن ساده را از ارتباط بین Alice و Bob بازیابی نماید. همزمان T قادر به ارسال داده‌های Bob به Alice پس از رمزگذاری مجدد داده‌ها با بهره‌گیری از کلید عمومی واقعی Alice می‌باشد.

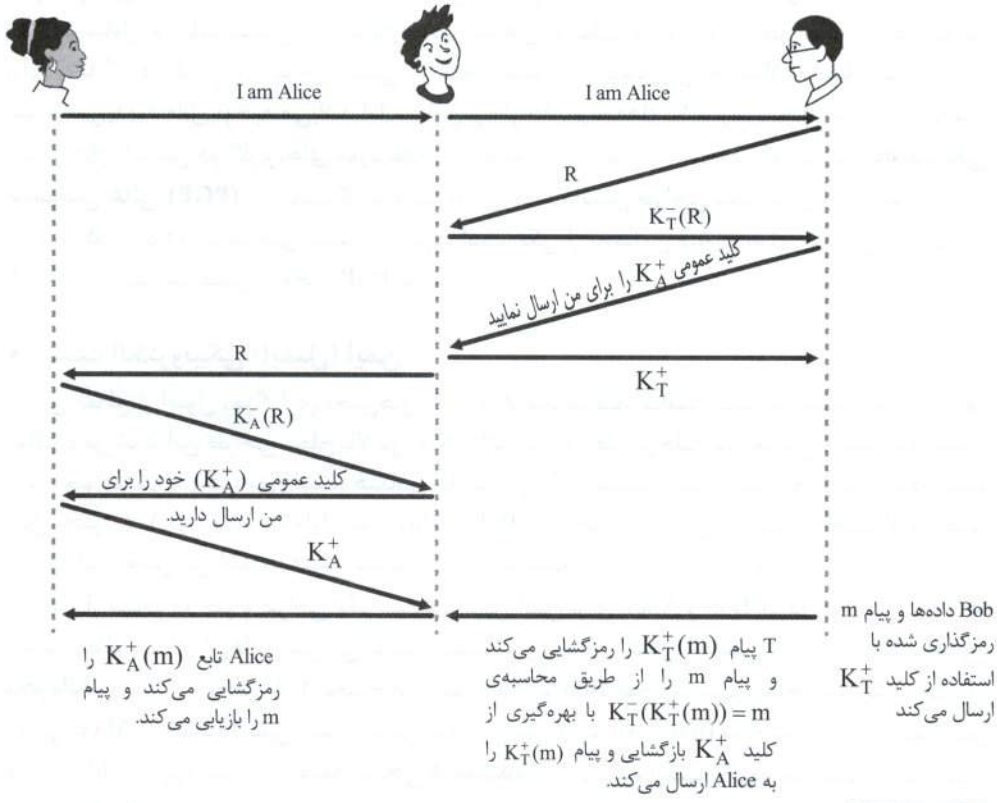
هر دو نفر Alice و Bob از نقطه نظر ارسال و دریافت داده‌ها با بهره‌گیری از کلید عمومی Alice راضی به نظر می‌رسند بدون آنکه از حضور T باخبر باشند. اگر Bob و Alice بعداً یکدیگر را ملاقات نمایند و تعامل بین خود را به بحث گذارند آن‌گاه Alice دقیقاً آن‌چه را Bob فرستاده است دریافت می‌دارد و بنابراین هیچ مورد مشکوکی جلب نظر نمی‌کند. این موضوع نمونه‌ای است از **تهاجم فرد واسطی (MIMA)** هنگام انجام یک تعامل اینترنتی.

۸-۱۰ تأمین امنیت پست الکترونیکی (ایمیل)

در بخش‌های قبل موضوعات بنیادی امنیت شبکه شامل رمزگذاری کلید عمومی و خصوصی، اعتبارسنجی نقطه‌ی انتهایی، توزیع کلید، یکپارچگی پیام و امضاهای دیجیتالی بررسی شدند. در ادامه‌ی فصل چگونگی تأمین امنیت از سوی این ابزار در اینترنت تحلیل می‌شوند.

نکته‌ی جالب توجه آن است که تأمین خدمات امنیتی در هریک از چهار لایه‌ی پشته‌ی پروتکلی

اینترنت امکان‌پذیر است. هنگامی که امنیت برای یک پروتکل لایه‌ی کاربرد تأمین می‌شود، کاربرد وابسته به این پروتکل از بیش از یک خدمات امنیتی بهره‌مند می‌شود. از جمله‌ی این خدمات می‌توان از محرمانگی، اعتبارسنجی و یکپارچگی پیام نام برد. هنگامی که امنیت برای لایه‌ی انتقال فراهم می‌شود همه‌ی کاربردهایی که از آن پروتکل بهره می‌گیرند خدمات امنیتی لایه‌ی انتقال را نیز در دسترس خود دارند. هنگامی که امنیت در لایه‌ی شبکه برمبنای میزبان به میزبان فراهم می‌شود همه‌ی قطعه‌های لایه‌ی انتقال و بنابراین همه‌ی داده‌های لایه‌ی کاربرد از خدمات امنیتی لایه‌ی شبکه بهره‌مند می‌شوند. سرانجام هنگامی که امنیت برمبنای یک پیوند تأمین می‌شود آن‌گاه داده‌ها در همه‌ی قاب‌های در حال انتشار از طریق آن پیوند بهره‌مند از خدمات امنیتی آن پیوند می‌شوند.



شکل ۸-۱۹ ♦ تهاجم فردمیانی (MIMA) در شبکه

در بخش‌های ۸-۱۰ تا ۸-۱۶ ابزارهای امنیتی استفاده شده در لایه‌های کاربرد، انتقال، شبکه و پیوند داده‌ها تحلیل می‌شوند. با توجه به چارچوب از فراز به فرود در این کتاب، ابتدا از فراز پشته‌ی پروتکلی آغاز و پیاده‌سازی مسایل امنیتی را در لایه‌ی کاربرد بررسی می‌کنیم. رویکردی که در این رابطه دنبال می‌کنیم بهره‌گیری از یک کاربرد ویژه یعنی ایمیل به‌عنوان موضوع مطالعه‌ی امنیت لایه‌ی کاربرد می‌باشد. سپس حرکت تحلیلی خود را به‌طرف پائین پشته‌ی پروتکلی انجام می‌دهیم. پروتکل SSL مخصوص تأمین امنیت لایه‌ی انتقال، IPsec با هدف تأمین امنیت لایه‌ی شبکه و امنیت پروتکل LAN بی‌سیم IEEE802.11 مواردی هستند که به‌ترتیب بررسی می‌شوند.

ممکن است این نکته باعث حیرت باشد که چرا عملکردهای امنیتی برای بیش از یک لایه در اینترنت فراهم می‌شود؟ آیا به‌سادگی کفایت نمی‌کند که تأمین امنیت در لایه‌ی شبکه انجام شود؟ دو پاسخ را می‌توان برای این پرسش‌ها ارایه نمود. نخست، اگرچه امنیت در لایه‌ی شبکه می‌تواند یک پوشش مناسب از طریق رمزگذاری همه‌ی داده‌های موجود در داده‌گرام‌ها باشد که همان قطعه‌های لایه‌ی انتقال به‌شمار می‌آیند و سپس با اعتبارسنجی همه‌ی آدرس‌های IP متعلق به منبع داده‌ها این امنیت کامل می‌گردد، اما نمی‌تواند امنیت در سطح کاربر را تأمین نماید. به عنوان مثال یک سامانه‌ی تجاری نمی‌تواند فقط اعتماد خود را بر امنیت لایه‌ی IP به‌منظور اعتبارسنجی مشتری که کالاهایی را از آن سامانه خریداری می‌نماید قرار دهد. در نتیجه نیاز برای عملکردهای امنیتی در لایه‌های بالاتر به همراه پوشش امنیتی لایه‌های پائین‌تر آشکار می‌شود. دوم، به طور کلی بهره‌گیری از خدمات جدید اینترنت شامل خدمات امنیتی در لایه‌های بالاتر پشته‌ی پروتکلی به مراتب ساده‌تر است. اگرچه انتظار برای به‌کارگیری گسترده‌ی عوامل امنیتی در لایه‌ی شبکه که تأمین آن احتمالاً در سال‌های آینده صورت می‌پذیرد قابل توجیه می‌باشد اما بسیاری از طراحان برنامه‌های کاربردی علاقه‌مند به تأمین عملکردهای امنیتی در کاربردهای مورد علاقه‌ی خویش می‌باشند. یک نمونه‌ی کلاسیک **محرمانگی خصوصی عالی (PGP)**[1] می‌باشد که با هدف تأمین امنیت ایمیل طراحی شده است. این نمونه تنها نیازمند کد کاربردی برنامه‌های مشتری و سرور است. یکی از نخستین فناوری‌های امنیتی با کاربرد گسترده در اینترنت همین نرم‌افزار PGP است.

- **پست الکترونیکی (ایمیل) ایمن**

در این مقطع از اصول رمزگذاری بخش‌های ۸-۲ تا ۸-۵ به‌منظور تأمین یک سیستم ایمن ایمیل استفاده می‌شود. این طراحی سطح بالا در رویکرد افزایشی در هر مرحله با معرفی خدمات امنیتی جدید صورت می‌پذیرد.بدیهی است، هنگامی که طراحی یک سیستم ایمیل ایمن انجام می‌گیرد آنگاه مثال بخش ۸-۱ در ارتباط با تعامل بین Alice و Bob و مهاجم T که مایل به شنود و احتمالاً مداخله در امور این تعامل می‌باشد به عنوان سنگ بنای این سیستم درنظر گرفته می‌شود.

قبل از شتاب به جلو و طراحی یک سیستم ایمیل ایمن برای Alice و Bob ابتدا نوع ویژگی‌های امنیتی مطلوب برای آن‌ها بررسی می‌شوند. نخستین و مهم‌ترین نگرانی امنیتی از دید آن‌ها موضوع **محرمانگی** است. در بخش ۸-۱ توضیح داده شد Alice و Bob هیچ یک مایل به خواندن پیام ایمیل ارسالی Alice از جانب T نمی‌باشند. ویژگی دوم آن است که Alice و Bob به احتمال زیاد از نقطه نظر امنیتی مایل به بهره‌گیری از **اعتبارسنجی فرستنده** در سیستم ایمیل خود می‌باشند، به ویژه هنگامی که Bob پیامی مانند:

"I don't like you anymore. I never want to see you again. Fromally yours. Alice"

را دریافت می‌دارد. او مایل است اطمینان حاصل نماید که این پیام از سوی Alice ارسال شده است و T به هیچ روی در ارسال این پیام غم‌انگیز دخالت ندارد. ویژگی دیگری که این دو دوست صمیمی مایل به بهره‌گیری از آن می‌باشند **یکپارچگی پیام** است که این اطمینان را به Bob می‌دهد که پیام ارسالی از سوی Alice در بین مسیر حرکت خود دستخوش جعل و یا دستکاری نشده است. سرانجام سیستم پست الکترونیکی بایستی **اعتبارسنجی گیرنده** را نیز تأمین نماید تا Alice از این موضوع

1- Pretty good privacy

اطمینان حاصل نماید که او یعنی Alice تنها فردی است که نامه را به Bob ارسال نموده و T نمی‌تواند خود را جایگزین Bob معرفی نماید.

ابتدا مهم‌ترین نگرانی یعنی محرمانگی پیام بررسی می‌شود. یک روش بسیار ساده که می‌تواند محرمانگی را برای Alice با هدف رمزگذاری پیام تامین نماید، فناوری کلید متقارن مانند DES و یا AES می‌باشد. در نقطه‌ی مقابل رمزگشایی پیام از سوی Bob را نیز امکان‌پذیر می‌سازد. همان‌گونه که در بخش ۸-۲ عنوان شده است اگر کلید متقارن به اندازه‌ی کافی بزرگ باشد و اگر این کلید فقط در اختیار Alice و Bob باشد آن‌گاه برای هر مهاجمی حتی T که زیرک است خواندن پیام بسیار دشوار می‌شود. اگرچه این رویکرد واضح است اما بنابر مطالب بخش ۸-۲ با دشواری توزیع کلید متقارن روبه‌رو می‌باشد که فقط Alice و Bob نسخه‌ای از آن را در اختیار دارند. طبیعی است رویکردی جایگزین درنظر گرفته شود یعنی رمزگذاری کلید عمومی با بهره‌گیری از الگوریتمی مانند RSA. در رویکرد کلید عمومی Bob کلید عمومی خود را از طریق یک سرور کلید عمومی و یا صفحه‌ی وب شخصی خود در اختیار همگان قرار می‌دهد. پیام ارسالی از سوی Alice با استفاده از کلید عمومی Bob رمزگذاری و او پیام رمز شده را به سوی آدرس ایمیل Bob گسیل می دارد. هنگامی که Bob پیام را دریافت می‌کند او آن را با استفاده از کلید خصوصی خود رمزگشایی می‌کند. اگر فرض شود Alice مطمئن از تعلق کلید عمومی به Bob می‌باشد این رویکرد ابزار بسیار مناسبی به‌منظور تأمین محرمانگی موردنظر است. تنها اشکال در آن است که کلید رمزگذاری عمومی نسبتاً ناکارآمد به‌ویژه برای پیام‌های طولانی می‌باشد.

به‌منظور برطرف نمودن ضعف ناکارآمدی می‌توان از یک کلید تعامل بهره گرفت. به ویژه آنکه Alice می‌تواند عملیات ذیل را با بهره‌گیری از آن انجام دهد:

۱- گزینش یک کلید متقارن تصادفی K_S.

۲- رمزگذاری پیام خود یعنی m با آن کلید متقارن.

۳- رمزگذاری کلید متقارن با بهره‌گیری از کلید عمومی K_B^+ متعلق به Bob.

۴- افزودن پیام رمز شده به کلید متقارن رمز شده و تشکیل یک بسته.

۵- ارسال بسته به آدرس ایمیل Bob.

مراحل پنج‌گانه‌ی بالا در شکل ۸-۲۰ نشان داده شده‌اند. در این شکل و تصاویر بعدی علامت $\oplus$ به مفهوم افزودن یا پیوست و علامت $\ominus$ به معنی کاستن یا تجزیه می‌باشد. هنگامی که Bob بسته‌ای را از طریق پست الکترونیکی خود دریافت می‌دارد او می‌تواند اقدام به انجام مراحل ذیل نماید:

۱- از کلید خصوصی K_B^- خود به‌منظور دستیابی کلید متقارن K_S بهره گیرد.

۲- از کلید متقارن K_S با هدف رمزگشایی پیام m استفاده نماید.

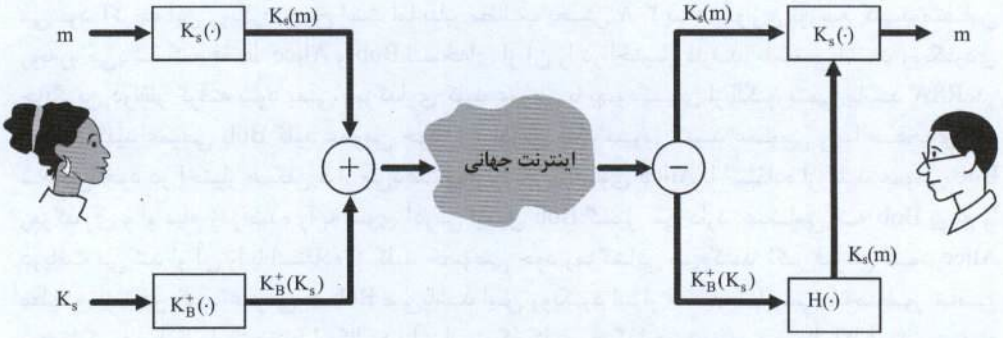
پس از طراحی یک سیستم ایمیل ایمن که محرمانگی را تأمین می‌کند، اکنون به طراحی سیستم دیگری می‌پردازیم که هر دو مورد اعتبارسنجی فرستنده و یکپارچگی پیام را تأمین می‌کند. برای یک لحظه این تصور را در ذهن خود مجسم نمائید که Alice و Bob دیگر نگران موضوع محرمانگی نیستند و مایل به اشتراک تعامل بین خود با هر فردی می‌باشند! و فقط نگران اعتبارسنجی فرستنده و یکپارچگی پیام می‌باشند. برای دستیابی به این هدف از امضاهای دیجیتالی و هضم پیام که در بخش ۸-۵ توصیف شده است، استفاده می‌شود. عملیاتی که Alice انحصاراً انجام می‌دهد به‌صورت ذیل است:

۱- اعمال یک تابع هاش H مانند MD5 به پیام m خویش تا از طریق آن عامل تجزیه‌ی پیام نتیجه شود.

۲- نتیجه‌ی تابع هاش را از طریق کلید خصوصی K_A^- خود امضا می‌کند تا امضای دیجیتالی موردنظر را تعریف نماید.

۳- افزودن یا پیوست پیام اصلی (رمز نشده) به امضای دیجیتالی تا یک بسته تولید شود.

۴- ارسال بسته به آدرس پست الکترونیکی .Bob



Alice پیام ایمیل (m) را ارسال می‌دارد. **Bob پیام ایمیل (m) را دریافت می‌دارد.**

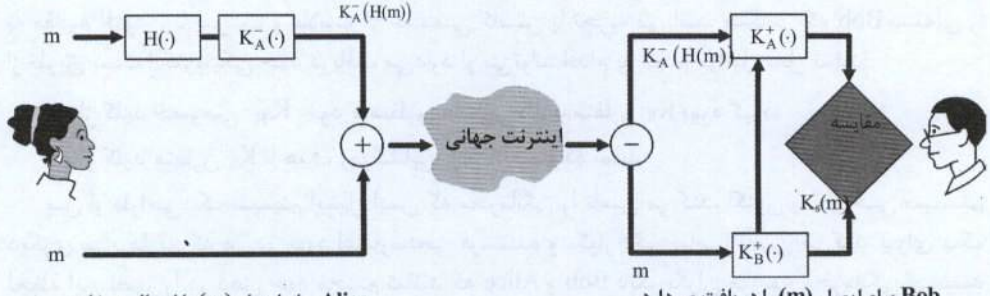
شکل ۸-۲۰ ♦ بهره‌گیری از کلید تعامل متقارن K_S از سوی Alice جهت ارسال ایمیل محرمانه به Bob

هنگامی که Bob بسته‌ای را دریافت می‌کند وی مبادرت به عملیات ذیل می‌کند:

۱- اعمال کلید عمومی K_A^+ متعلق به Alice به پیام امضا شده.

۲- مقایسه‌ی نتیجه‌ی این عملیات با هاش H از پیام خود.

این مراحل در شکل ۸-۲۱ نشان داده شده‌اند. بدیهی است برمبنای آن‌چه در بخش ۸-۵ عنوان گردیده است اگر دو نتیجه مشابه یکدیگر باشند آن‌گاه Bob با اطمینان می‌تواند پیام دریافتی را از سوی Alice و بدون هرگونه احتمال جعل قلمداد نماید.



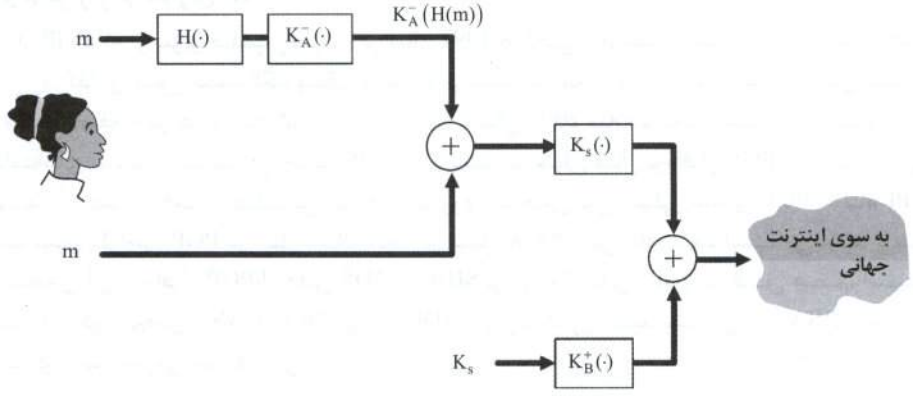
Alice پیام ایمیل (m) را ارسال می‌دارد. **Bob پیام ایمیل (m) را دریافت می‌دارد.**

شکل ۸-۲۱ ♦ بهره‌گیری از توابع هاش و امضاهای دیجیتالی با هدف تأمین اعتبارسنجی فرستنده و یکپارچگی پیام

اکنون حالتی را درنظر بگیرید که مایل به طراحی یک سیستم ایمیل می‌باشید که محرمانگی،

اعتبارسنجی فرستنده و یکپارچگی پیام را تأمین می‌کند. این مهم از طریق ترکیب مراحل نشان داده شده در دو شکل ۸-۲۰ و ۸-۲۱ امکان‌پذیر می‌باشد. نخست Alice اقدام به خلق یک بسته‌ی اولیه می‌کند که عیناً برمبنای شکل ۸-۲۱ است به این مفهوم که شامل پیام اصلی او به‌همراه یک امضای هاش دیجیتالی از پیام می‌باشد. او سپس با این بسته‌ی ابتدایی به‌عنوان یک پیام برخورد می‌کند و این پیام جدید را از طریق گام‌های فرستنده در شکل ۸-۲۰ گسیل می‌کند. حاصل نهایی این عمل تعریف یک بسته‌ی جدید است که به سوی Bob ارسال می‌شود. مراحل اعمال شده از سوی Alice در شکل ۸-۲۲ به نمایش درآمده‌اند. هنگامی که Bob بسته را دریافت می‌دارد او نخست مراحل نشان داده شده در شکل ۸-۲۰ در سمت خود را اعمال می‌کند و سپس به اعمال مراحل طرف شکل ۸-۲۱ می‌پردازد. بدیهی است دستاورد این طراحی شامل تأمین محرمانگی، اعتبارسنجی فرستنده و یکپارچگی پیام می‌شود. لازم به توضیح است در این روش Alice دو مرتبه از کلید عمومی رمزگذاری بهره می‌گیرد: (۱) هنگامی که از کلید خصوصی خود استفاده می‌کند. (۲) هنگامی که از کلید عمومی Bob بهره می‌برد. به‌گونه‌ی مشابه Bob نیز دو مرتبه از کلید عمومی رمزگذاری خود استفاده می‌کند: (۱) هنگام اعمال کلید خصوصی و (۲) هنگام بهره‌گیری از کلید عمومی متعلق به Alice.

چارچوب طراحی ایمیل ایمن در شکل۸-۲۲ نشان داده شده است. این احتمال وجود دارد که اغلب کاربران ایمیل از امکانات امنیتی فراهم آمده خشنود می‌باشند، اما هنوز یک موضوع مهم وجود دارد که بایستی مورد توجه قرار گیرد. در طراحی شکل ۸-۲۲ از Alice خواسته شده است که کلید عمومی Bob را دستیابی نماید و از Bob نیز خواسته شده است کلید عمومی Alice را اخذ نماید. توزیع این کلیدهای عمومی چندان هم ساده نمی‌باشد. به عنوان مثال مهاجم T ممکن است خود را به جای Bob معرفی نماید و کلید عمومی خود را در اختیار Alice قرار دهد در حالیکه اعلام می‌دارد کلید متعلق به Bob است. در نتیجه T قادر به دریافت پیام‌هایی می‌شود که در اصل هدف آن‌ها Bob بوده است. در بخش ۸-۵ آموختیم یک رویکرد محبوب برای توزیع محرمانه‌ی کلیدهای عمومی بهره‌گیری از عامل **گواهی کلیدهای عمومی** با بهره‌گیری از CA می‌باشد.



شکل ۸-۲۲ ♦ با بهره‌گیری از کلیدهای رمز متقارن، عمومی و یک تابع هاش و نیز یک امضای دیجیتالی از سوی Alice محرمانگی، اعتبارسنجی فرستنده و یکپارچگی پیام تأمین می‌شود.

• پیشینه‌ی PGP و فیل زیمرمن

فیلیپ زیمرمن (P.R. Zimmerman) شخصیتی است که عامل مؤثر طراحی PGP یا محرمانگی عالی بوده است که قبلاً به آن اشاره شد. برای همین منظور وی به مدت سه سال هدف بازجویی‌های کیفری در آمریکا بود، زیرا دولت آمریکا بر این باور بود که وی محدودیت‌های اعمال شده در زمینه‌ی صادرات نرم‌افزارهای رمزگذاری فدرال را نقض نموده است، زیرا PGP را در سراسر گیتی با انتشار آزاد افزار خود در سال 1991 گسترش و توزیع نموده است. پس از انتشار PGP به عنوان یک امکان مشترک در دسترس عموم، فرد دیگری آن را برروی اینترنت قرار داد و شهروندان غیرآمریکایی قادر به نسخه‌برداری (دانلود) از آن شدند. برنامه‌های رمزگذاری در ایالات متحده از سوی قوانین دولت فدرال به عنوان جنگ افزار طبقه‌بندی شده‌اند و بنابراین صادرات آن‌ها غیرمجاز است.

بدون برخورداری از بودجه‌ی کافی، نیروی فعال حقوق بگیر، عدم پشتیبانی از سوی شرکت‌های تجاری فنآوری اطلاعات و نیز دخالت دولت فدرال در هر حال PGP یکی از محبوب‌ترین نرم‌افزارهای رمزگذاری اینترنت در سراسر گیتی می‌باشد. شاید یکی از علل موفقیت چشمگیر و محبوبیت جهانی برای آن را بتوان دخالت دولت فدرال آمریکا در تحت تعقیب قرار دادن زیمرمن دانست.

در سال 1996 میلادی دولت فدرال شکایت خود بر علیه زیمرمن را بازپس گرفت و این موضوع سبب خشنودی فعالان در عرصه‌ی اینترنت گردید. موضوع زیمرمن تبدیل به داستان فرد بی‌گناهی شده است که برای کسب حقوق خود در برابر سوء استفاده در مقابل دولتی بزرگ ایستاده است و تسلیم دولت آمریکا در برابر وی به دلیل آنکه سبب تصویب قانون منع سانسور در اینترنت از سوی کنگره‌ی امریکا گردید و نیز به دلیل فشار FBI به‌منظور افزایش دخالت دولت در شنود اینترنتی خبری خوش محسوب گردید. پس از توقف پی‌گیری حقوقی دولت آمریکا زیمرمن شرکت PGP را تأسیس نمود که به قیمت نسبتاً قابل توجهی از سوی سازمان NA در سال 1997 خریداری گردید. زیمرمن در حال حاضر یکی از مشاوران برجسته‌ی مستقل در رابطه با موضوع رمزگذاری شبکه‌های کامپیوتری می‌باشد.

• نرم‌افزار رمزگذاری PGP

نرم‌افزار PGP که از سوی فیلیپ زیمرمن در سال 1991 طراحی و نوشته شد یکی از استانداردهای جهانی رمزگذاری متون پست الکترونیکی (ایمیل) می‌باشد. سامانه‌ی وب این نرم‌افزار حامل بیش از یک میلیون صفحه در هر ماه به کاربران 166 کشور تا سال 2007 میلادی بوده است. نسخه‌های PGP در دامنه‌های عمومی اینترنت در اختیار کاربران می‌باشد. به عنوان مثال نرم‌افزار PGP را می‌توان برروی کاربردهای محبوب یافت و مطالب بی‌شماری را بر روی صفحه‌ی بین المللی خانه‌ی PGP به نام PGPI مطالعه نمود. طراحی PGP در اصل مشابه آنچه در شکل ۸-۲۲ نشان داده شده است، می‌باشد و بنابر نوع نسخه‌ی آن، نرم‌افزار PGP از هاش MD5 یا SHA برای محاسبه‌ی پیام رمز قابل هضم، استفاده می‌کند. از الگوریتم‌های CAST، 3DES و یا IDEA برای رمزگذاری کلید متقارن و از RSA به‌منظور رمزگذاری کلید عمومی استفاده می‌شود.

هنگامی که PGP نصب می‌شود نرم‌افزار یک جفت کلید عمومی برای کاربر تعریف می‌کند. کلید عمومی را می‌توان بر روی سامانه‌ی وب کاربر پست نمود و یا آن را در یک سرور کلید عمومی قرار داد.

از کلید خصوصی با بهره‌گیری از یک رمز عبور می‌توان حفاظت نمود. نرم‌افزار PGP امکانات

اختیاری امضای دیجیتالی پیام، رمزگذاری پیام و یا هر دو را در اختیار کاربران قرار می‌دهد. شکل ۸-۲۳ یک پیام امضا شده‌ی PGP را نشان می‌دهد. این پیام پس از سرپیام MIME ظاهر می‌شود. داده‌ی رمزگذاری شده در پیام برابر $K_A^-(H(m))$ می‌باشد که همان پیام هضم با امضای دیجیتالی می‌باشد. قبلاً اشاره گردید برای آنکه Bob یکپارچگی پیام را اعتبارسنجی نماید وی نیازمند دستیابی کلید عمومی Alice می‌باشد.

```
----BEGIN PGP SIGNED MESSAGE----
Hash : SHA1
Bob:
Can I see you tonight?
Passionately yours, Alice
----BEGIN PGP SIGNED SIG-NATURE---
Version: PGP for Personal Privacy 5.0
Charest: noconv
yhHJRHhGJGhgg/12EpJ+1o8gE4v3mJhFEvZP9t6n7G6m5Gw2
----END PGP SIGNED SIG-NATURE---
```

شکل ۸-۲۳ ♦ یک پیام PGP امضا شده

شکل ۸-۲۴ پیام رمز PGP را نشان می‌دهد. این پیام نیز پس از سرپیام MIME ظاهر می‌شود. بدیهی است متن خام پیام در بین پیام ایمیل جایگذاری نشده است. هنگامی که یک فرستنده مانند Alice مایل است از امکانات محرمانگی و یکپارچگی بهره‌گیری نماید PGP شامل پیامی است مشابه شکل ۸-۲۴ در درون پیام شکل ۸-۲۳. نرم‌افزار PGP هم‌چنین روشی را برای گواهی کلید عمومی تأمین می‌کند اما این روش کاملاً متفاوت از CA می‌باشد. کلیدهای عمومی PGP از طریق **شبکه‌ای از معتمدها**[1] گواهی می‌شود. فردی مانند Alice هر جفت کلید و نام کاربری را در صورت اطمینان از تعلق آن‌ها به یکدیگر می‌تواند گواهی نماید. هم‌چنین PGP اجازه می‌دهد که Alice در صورت اعتماد کاربر دیگری از نقطه نظر اعتبارسنجی کلیدهای بیش‌تری را ارایه نماید. برخی از کاربران PGP کلیدهای همدیگر را به‌گونه‌ی متقابل امضا می‌کنند. در این حالت کاربران با حضور فیزیکی به تبادل کلیدهای عمومی می‌پردازند و سپس گواهی کلید یکدیگر را از طریق امضای آن‌ها با کلیدهای خصوصی خود انجام می‌دهند.

```
----BEGIN PGP MESSAGE----
Version: PGP for Personal Privacy 5.0
u2R4d+/jKmn8Bc5+hgDsqAewsDfrGdszX681iKm5F6Gc4sDfcXyt
Rfds10juHgbcfDssWe7/K=1KhnMiKLo0+1/BvcX4t==Ujk9PbcD4
Thdf2awQfgHbnmK1ok8iy6gTh1p
----END PGP MESSAGE
```

شکل ۸-۲۴ ♦ یک نمونه پیام رمز PGP

1- Web of trust

۸-۱۱ ارتباط‌های ایمن TCP: SSL

در بخش قبل ملاحظه شد چگونه روش‌های رمزگذاری محرمانگی، یکپارچگی داده و اعتبارسنجی نقطه‌ی انتهایی برای یک کاربرد ویژه مانند e-mail تأمین می‌گردد. در این بخش یک گام از فراز پشته‌ی پروتکلی به طرف پائین حرکت می‌کنیم و تأمین خدمات امنیتی را با بهره‌گیری از رمزگذاری در ارتباط‌های TCP شامل محرمانگی، یکپارچگی داده و اعتبارسنجی نقطه‌ی انتهایی جستجو می‌کنیم. این نسخه‌ی گسترش یافته‌ی TCP معروف به **لایه‌ی سوکت‌های ایمن (SSL)**[1] می‌باشد. نسخه‌ی سوم تغییر یافته‌ی SSL توصیف شده در مستند [RFC 2246] می باشد که از سوی IETF استاندارد شده است و مشهور به **امنیت لایه‌ی انتقال (TLS)**[2] می‌باشد.

رمزگذاری SSL در ابتدا از از سوی شرکت Netscape طراحی شد اما فلسفه‌ی فکری امنیت TCP به قبل از پیدایش این شرکت مربوط می‌شود. از زمان ورودش، SSL کاربردی فراگیر و گسترده را در شبکه‌ی اینترنت داشته است. پروتکل SSL از سوی همه‌ی مرورگرهای وب و نیز سرورهای وب پشتیبانی می‌شود و تقریباً از سوی همه‌ی سامانه‌های تجاری اینترنت از جمله آمازون، یاهو، MSN، eBay و غیره مورد استفاده قرار گرفته است. هر سال ده‌ها میلیارد دلار برای SSL هزینه می‌شود. در واقع هرگاه خریدی از طریق اینترنت انجام می‌گیرد در آن به نوعی از کارت‌های اعتباری بانکی استفاده می‌شود ارتباط بین مرورگر شما و سرور وابسته به عملیات خرید قطعاً از درون SSL عبور نموده است. هنگامی از کاربرد SSL در تعامل‌های اینترنتی روزانه خود آگاه می‌شوید که مشاهده نمائید آغاز URL مورد نظر شما با: **HTTPS** است تا **HTTP**.

به‌منظور درک نیاز برای SSL مراحل یک تعامل تجاری اینترنتی را در ذهن خود مجسم نمائید. هنگامی که Bob مشغول وبگردی است و گام به درون سامانه‌ی تجاری اینترنتی Alice می‌گذارد که تخصص آن فروش عطریات است. سامانه‌ی تجاری Alice پرسش‌نامه یا فرمی را به نمایش می‌گذارد که انتظار می‌رود Bob نوع عطر موردنظر و کمیت مطلوب خود را به‌همراه آدرس تحویل و شماره کارت اعتباری به‌منظور عملیات بانکی در آن وارد نماید. این اطلاعات از سوی Bob وارد و پس از کلیک برروی دکمه‌ی ارایه (Submit) به انتظار تحویل عطر خریداری شده از طریق پست می‌ماند. علاوه بر آن این احتمال وجود دارد که وی هزینه‌ای را برای سفارش خود هنگام دریافت صورت حساب بانکی متحمل گردد. همه‌ی این موارد عادی به‌نظر می‌آیند اما در صورت عدم وجود تدابیر امنیتی Bob ممکن است با موارد تعجب برانگیز متعددی به شرح ذیل روبه‌رو شود:

- در صورتی که محرمانگی و به عبارت دیگر رمزگذاری مورد استفاده قرار نگیرد یک مهاجم سفارش Bob را شنود و اطلاعات کارت اعتباری وی را در اختیار خود می‌گیرد. آن‌گاه مهاجم اشاره شده از جانب Bob برای نیازهای خود خرید انجام می‌دهد.
- اگر از امکانات یکپارچگی داده استفاده نشود یک مهاجم قادر به اعمال تغییر در سفارش Bob می‌شود و در نتیجه ممکن است به عنوان مثال ده برابر نیاز Bob اقدام به سفارش خرید عطر نماید.
- سرانجام اگر از اعتبارسنجی سرور بهره‌گیری نشده باشد یک سرور ممکن است علامت تجاری معروف و متعلق به Alice را به نمایش بگذارد در حالی که سامانه در واقع از سوی

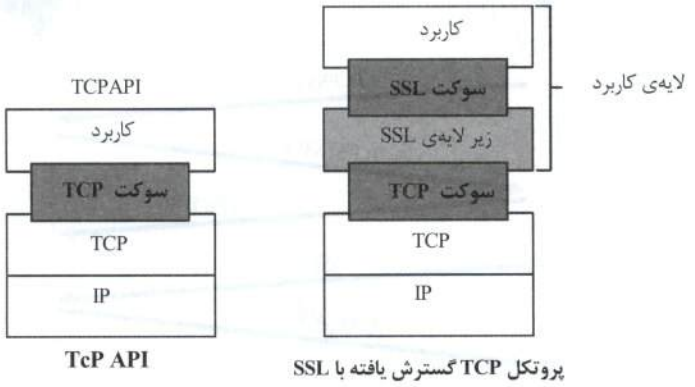
1- Secure sockets layer

2- Transport layer security

مهاجم T نگهداری و مدیریت می‌شود که خود را به جای شرکت Alice جایگذاری نموده است. پس از دریافت سفارش Bob مهاجم T ممکن است کلاهبرداری مالی از Bob نماید و یا آنکه سرقتی را با نام، آدرس و شماره کارت اعتباری Bob انجام دهد!

پروتکل SSL مانع از وقوع چنین رخدادهایی با بهینه‌سازی TCP از نقطه‌نظر محرمانگی، یکپارچگی داده‌ها و اعتبارسنجی سرور و نیز برنامه‌ی مشتری می‌شود.

پروتکل SSL در اغلب موارد به‌منظور تأمین امنیت تراکنش‌های در حال انجام بر روی HTTP مورد استفاده قرار می‌گیرد. از آن‌جائیکه SSL امنیت TCP را تأمین می‌کند از آن می‌توان به‌منظور استفاده‌ی هر کاربردی که بر روی TCP اجرا می شود و مشابه TCP-API می‌باشد، بهره گرفت. هنگامی که یک کاربرد مایل به بهره‌گیری از SSL می‌باشد. آن کاربرد شامل کلاس‌ها و کتابخانه‌های SSL می‌شود. در شکل ۸-۲۵ نشان داده شده است اگرچه SSL از لحاظ فنی در لایه‌ی کاربرد سکونت دارد اما از نقطه‌نظر یک طراح این پروتکل انتقال است که خدمات TCP را به‌همراه امنیت توسعه یافته‌ی آن تأمین می‌کند.



شکل ۸-۲۵ ♦ اگرچه SSL از لحاظ فنی در لایه‌ی کاربرد قرار می‌گیرد اما از نقطه‌نظر طراحان پروتکل لایه‌ی انتقال است.

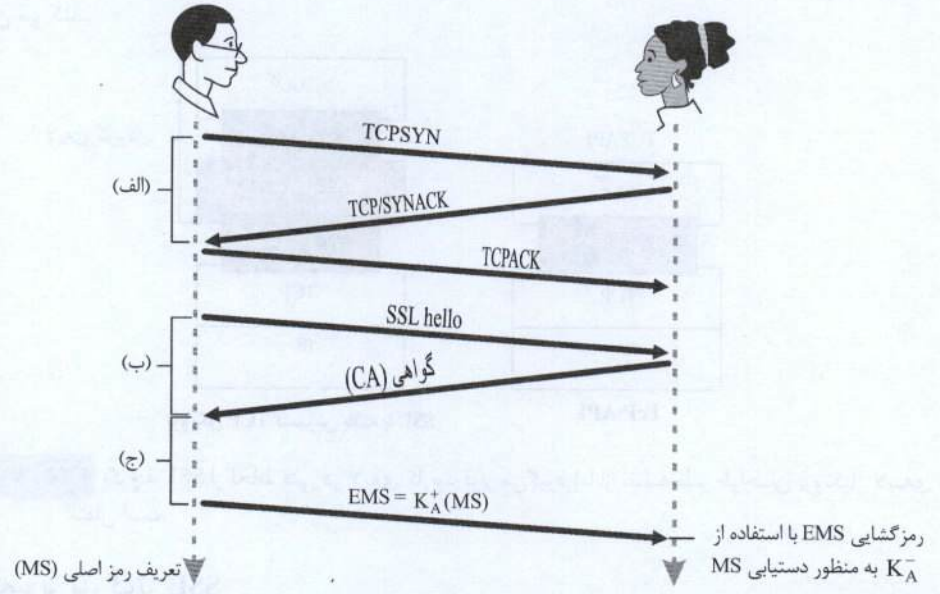
• تصویر بزرگ‌تر SSL

ابتدا به توصیف نسخه‌ی ساده شده‌ای از SSL می‌پردازم تا اجازه دهد تصویر بزرگ‌تری از چرا و چگونگی SSL در ذهن خود ترسیم نمائیم. نسخه‌ی ساده شده‌ی SSL را با نام «تقریباً SSL» تعریف می‌کنیم پس از تشریح «تقریباً SSL» در بخش بعدی به SSL واقعی می‌پردازیم. هر دو SSL و تقریباً SSL دارای سه مرحله‌ی **دست‌دهی، کشف کلید** و **انتقال داده** می‌باشند. اکنون به توصیف این سه مرحله برای یک تعامل ارتباطی بین یک برنامه‌ی مشتری مانند Bob و یک برنامه‌ی سرور مانند Alice می‌پردازیم که در آن Alice دارای یک جفت کلید عمومی و خصوصی و یک گواهی می‌باشد که شناسه‌ی Alice را به کلید عمومی او ارتباط می‌دهد.

۱- دست‌دهی

هنگام انجام دست‌دهی، Bob نیازمند: (الف) تعریف و ایجاد یک ارتباط TCP با Alice می‌باشد، (ب) اعتبارسنجی این نکته که Alice واقعاً او می‌باشد و (ج) ارسال یک کلید رمز اصلی به Alice که از

سوی هر دو نفر Alice و Bob به‌منظور تولید همه‌ی کلیدهای متقارن که مورد نیاز آن‌ها برای تعامل SSL است بهره‌برداری می شود. این سه مرحله در شکل ۸-۲۶ به نمایش گذارده شده‌اند. بدیهی است هنگامی که یک ارتباط TCP ایجاد می‌شود، Bob یک پیام "hello" را به سوی Alice ارسال می‌کند. پس از آن Alice با گواهی خود واکنش نشان می‌دهد که این گواهی حامل کلید عمومی او می‌باشد. همان‌گونه که در بخش ۸-۵ توضیح داده شد از آن‌جائیکه گواهی صادره از سوی CA گواهی امضا می‌شود Bob اطمینان کامل دارد کلید عمومی موجود در گواهی متعلق به Alice می‌باشد. پس از آن Bob یک **رمز اصلی (MS)**[1] که فقط برای این تعامل SSL کاربرد دارد را تولید می‌کند و آن‌گاه MS را با کلید عمومی Alice رمز می‌کند تا در نتیجه‌ی آن **رمز اصلی کدگذاری شده (EMS)**[2] را خلق و آن EMS را به سوی Alice گسیل نماید. پس از آن Alice نسبت به رمزگشایی EMS با کلید خصوصی خود اقدام می‌کند تا از طریق آن MS را دستیابی نماید. پس از این مرحله فقط Bob و Alice هر دو از MS این تعامل SSL آگاهی دارند.



شکل ۸-۲۶ ♦ دست‌دهی "تقریباً SSL" با آغاز یک ارتباط TCP

۲- کشف کلید

اصولاً MS که در این مقطع، مشترک بین Bob و Alice می باشد را می‌توان به عنوان کلید تعامل متقارن برای ارزیابی همه‌ی رمزگذاری‌ها و یکپارچگی پیام‌های بعدی که در دنبال نخستین تعامل ایجاد می‌شوند مورد استفاده قرار داد. اگرچه از لحاظ امنیت بهینه است که Alice و Bob از کلیدهای رمز گوناگون به‌منظور اهداف ارزیابی رمزگذاری و یکپارچگی استفاده نمایند. اما Alice و Bob هر دو از MS بهره می‌گیرند تا چهار کلید را تولید نمایند:

1- Master secret

2- Encrypter MS

- کلید E_B کلید رمزگذاری تعامل برای داده‌های ارسالی از Bob به Alice می باشد.
- کلید M_B عبارت است از کلید تعامل MAC برای داده‌های ارسالی از Bob به Alice.
- کلید E_A کلید رمز تعامل برای داده‌های ارسالی از سوی Alice به Bob می‌باشد.
- کلید M_A عبارت است از کلید تعامل MAC برای داده‌های ارسالی از Alice به Bob.

بدیهی است Alice و Bob هرکدام چهارکلید را از طریق MS تولید می‌کنند. این عمل از طریق برش MS در قالب چهار کلید انجام می‌گیرد. در SSL واقعی این موضوع اندکی پیچیده‌تر است. در انتهای مرحله‌ی کشف کلید Bob و Alice هر دو دارای چهار کلید می‌باشند. از دو کلید رمزگذاری به‌منظور کدگذاری رمز داده‌ها استفاده می‌شود و از دو کلید MAC با هدف اعتبارسنجی یکپارچگی داده‌ها بهره‌گیری می‌شود.

۳- انتقال داده

اکنون که Alice و Bob از چهار کلید تعامل مشابه E_B ، E_A ، M_B ، M_A استفاده می‌کنند آن‌ها می‌توانند اقدام به ارسال داده‌های ایمن به یکدیگر از طریق ارتباط TCP نمایند. از آن‌جائیکه TCP یک پروتکل جریان بایت می‌باشد یک رویکرد طبیعی برای SSL رمزگذاری داده‌های کاربردی در جریان می‌باشد و پس از آن ارسال داده‌های رمز شده به درون TCP جاری. اما برای انجام ارزیابی یکپارچگی MAC را در کدام نقطه بایستی قرار داد؟ بدیهی است ما مایل به انتظار تا انتهای تعامل TCP به‌منظور اعتبارسنجی یکپارچگی همه‌ی داده‌های Bob که از طریق تمامیت تعامل ارسال شده است! نمی‌باشیم. به همین منظور SSL جریان داده‌ها را به **رکوردها**[1] تقسیم می‌کند و سپس یک MAC را به هر رکورد برای ارزیابی یکپارچگی پیوست می‌کند. پس از انجام این مهم record+MAC را رمزگذاری می‌کند. برای خلق MAC ورودی‌های داده به همراه کلید M_B از طریق Bob به درون تابع هاش برمبنای مطالب بخش ۸-۵ تزریق می‌شود. پس از آن Bob با بهره‌گیری از کلید تعامل E_B بسته‌ی record+MAC را رمز نموده و آن را دراختیار TCP برای انتقال از طریق اینترنت قرار می‌دهد.

اگرچه این رویکرد مسیری طولانی را طی می‌کند هنوز از نقطه‌نظر یکپارچگی داده‌ها برای تمامی جریان پیام از امنیت کامل برخوردار نمی‌باشد. به‌ویژه اگر فرض شود T یک مهاجم فرد واسطی (MIMA) می‌باشد و قادر است قطعه‌های جریان TCP ارسالی بین Alice و Bob را اضافه، حذف و جایگزین نماید. مهاجم T به عنوان مثال می‌تواند دو قطعه‌ی ارسال شده به Bob را موقتاً متوقف و ترتیب توالی قطعه را عکس نماید. به بیان دیگر شماره‌های ردیف TCP که رمزگذاری نشده‌اند را تغییر داده و سپس دو قطعه‌ی TCP عکس شده از نقطه نظر ترتیب را به Alice ارسال نماید. با فرض آنکه هر قطعه‌ی TCP دقیقاً یک رکورد را بسته‌بندی می‌کند چگونگی پردازش قطعه‌ها از سوی Alice را می‌توان به‌صورت ذیل درنظر گرفت:

۱- جریان TCP در حال اجرا در محیط Alice بر این تفکر است که همه چیز وضعیت مطلوب خود را دارد و دو رکورد را به زیر لایه‌ی SSL گسیل می‌کند.

۲- پروتکل SSL در محیط Alice دو رکورد را رمزگشایی می‌کند.

۳- پروتکل SSL در محیط Alice با بهره‌گیری از MAC در هر رکورد اعتبارسنجی یکپارچگی داده را برای دو رکورد انجام می‌دهد.

1- Records

۴- پروتکل SSL پس از آن جریان بایت رمزگشایی شده از دو رکورد را به لایه‌ی کاربرد انتقال می‌دهد، اما جریان بایت کامل دریافتی از سوی Alice خارج از تنظیم می‌باشند زیرا نظم رکوردها معکوس شده است.

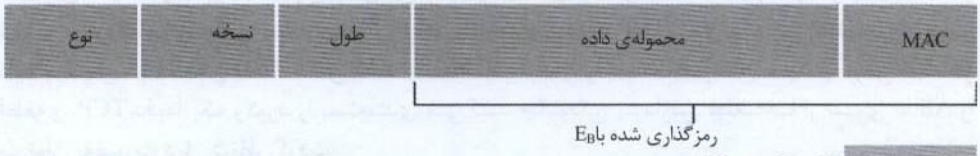
خوانندگان علاقه‌مند قویاً توصیه می‌شوند که سناریوی مشابهی را تحلیل نمایند که در آن مهاجم T قطعه‌ها را حذف و یا مجدداً بازپخش می‌کند.

راهکار این چالش می‌تواند بهره‌گیری از شماره‌های ردیف باشد. پروتکل SSL مراحل ذیل را در این ارتباط انجام می‌دهد:

- یک شمارشگر ردیف از سوی Bob تعریف و نگهداری می‌شود که در ابتدا ارزش آن صفر است و پس از ارسال هر رکورد SSL از سوی Bob مقدار آن افزایش می‌یابد. شماره ردیف هر رکورد در حقیقت از جانب Bob تعریف نمی‌شود بلکه هنگامی که او MAC را محاسبه می‌کند شماره ردیف را در محاسبه‌ی MAC اعمال می‌نماید. بنابراین MAC یک هاش از داده به همراه کلید MAC یعنی M_B و نیز شماره ردیف جاری می‌باشد. رهگیری شماره ردیف Bob از سوی Alice انجام می‌شود و به وی این امکان را می دهد که نسبت به یکپارچگی داده‌ها از یک رکورد با افزودن شماره ردیف مناسب در محاسبه‌ی MAC اعتبارسنجی لازم را انجام دهد. این بهره‌گیری از شماره ردیف SSL مانع از انجام تهاجم فرد واسطی از سوی T مانند بازپخش قطعه‌ها و یا تعویض ترتیب آن‌ها می‌شود.

۴- رکوردهای SSL

رکورد هر دو SSL و تقریباً SSL در شکل ۸-۲۷ نشان داده شده است. این رکورد شامل دامنه‌های **نوع، نسخه، طول، داده و MAC** می‌باشند. بدیهی است محتوای سه دامنه‌ی نخستین رمزگذاری نمی‌شود. دامنه‌ی نوع، تعیین کننده‌ی این واقعیت است که آیا رکورد یک پیام دست‌هی است و یا آنکه پیام شامل داده‌های کاربردی است. از آن هم‌چنین به‌منظور بستن یا قطع یک ارتباط SSL که در ادامه تشریح شده است، استفاده می‌شود. رکورد SSL در انتهای دریافت از دامنه‌ی طول به‌منظور جداسازی رکورد SSL از بین جریان بایت TCP ورودی استفاده می‌کند. دامنه‌ی نسخه نیز نیازی به توضیح ندارد و برای خواننده آشکار است.



شکل ۸-۲۷♦ قالب رکورد SSL

۸-۱۲ تصویر کامل‌ترین SSL

بخش قبل پروتکل تقریباً SSL را در پوشش قرار داد و نتیجه‌ی حاصل از آن درک بنیادی عملکرد چرا و چگونه‌ی SSL بود. اکنون که درک اصولی از SSL داریم پروتکل واقعی SSL را با عمق بیش‌تری تحلیل و بررسی می‌کنیم:

- **دست‌دهی SSL**

پروتکل SSL واقعی بهره‌گیری از یک الگوریتم کلید متقارن، کلید عمومی و یا MAC ویژه‌ای را به Alice و Bob تحمیل نمی‌کند، بلکه SSL این اجازه را به Alice و Bob می دهد که هر دو برروی الگوریتم‌های رمزگذاری در آغاز تعامل SSL موافقت نمایند. این توافق در طول انجام مرحله‌ی دست‌دهی انجام می‌شود. علاوه برآن در طول انجام این مرحله Alice و Bob پیام‌های موقت به یکدیگر نیز گسیل می‌دارند که از آن‌ها در خلق کلیدهای تعامل E_B ، E_A ، M_B ، M_A استفاده می‌شود. مراحل دست‌دهی SSL واقعی به‌صورت ذیل انجام می‌شود:

۱- برنامه‌ی مشتری فهرستی از الگوریتم‌های رمزگذاری که آن‌ها را پشتیبانی می‌کند همراه پیام‌های موقت مشتری گسیل می‌دارد.

۲- از میان یک فهرست، برنامه‌ی سرور الگوریتم متقارنی مانند AES، الگوریتم کلید متقارن مانند RSA با طول کلید مشخص و یک الگوریتم MAC را گزینش می‌کند. پس از آن فهرست انتخاب خود را به برنامه‌ی مشتری به‌همراه یک گواهی و یک پیام موقت سرور بازگشت می‌دهد.

۳- برنامه‌ی مشتری اعتبارسنجی گواهی را انجام و به دنبال آن کلید عمومی سرور را بازگشت می‌دهد. یک پیش رمز اصلی (PMS) را تولید می‌کند و PMS را با کلید عمومی سرور رمزگذاری و نتیجه را به سوی برنامه‌ی سرور گسیل می‌دارد.

۴- با بهره‌گیری از تابع کشف کلید مشابه که در استاندارد SSL مشخص شده است برنامه‌های مشتری و سرور مستقل از یکدیگر MS را از طریق PMS و پیام‌های موقت محاسبه می‌کنند. پس از برش PMS تعداد دو کلید رمزگذاری و دو کلید MAC تولید می‌شوند. علاوه بر آن هنگامی که رمز متقارن گزینش شده از CBC مانند 3DES یا AES بهره می‌گیرد آن‌گاه دو بردار اولیه (IV) برای هر سوی ارتباط را از طریق PMS دستیابی می‌کند. بنابراین همه‌ی پیام‌های ارسالی بین برنامه‌های مشتری و سرور به کمک MAC رمزگذاری و اعتبارسنجی می‌شوند.

۵- برنامه‌ی مشتری یک MAC از همه‌ی پیام‌های دست‌دهی را ارسال می‌دارد.

۶- برنامه‌ی سرور یک MAC از همه‌ی پیام‌های دست‌دهی را گسیل می‌دارد.

دو مرحله‌ی آخر از جعل پیام‌های دست‌دهی جلوگیری می‌کند. برای درک بهتر این موضوع به مرحله‌ی (۱) دقت نمائید، برنامه‌ی مشتری فهرستی از الگوریتم‌ها را پیشنهاد می‌کند، برخی از این الگوریتم‌ها قدرتمند و برخی دیگر ضعیف می‌باشند. این فهرست الگوریتم‌ها در قالب متن خام ارسال می‌شود زیرا الگوریتم‌های رمزگذاری و کلیدها هنوز در توافق با یکدیگر قرار ندارند. مهاجم فرضی T به عنوان یک تهاجم فرد واسطی (MIMA) می‌تواند الگوریتم‌های قوی‌تر را از فهرست حذف نماید و در نتیجه برنامه‌ی مشتری را مجبور به گزینش الگوریتم‌های ضعیف نماید. به‌منظور ممانعت از چنین اقدام مخربی، در مرحله‌ی ۵ برنامه‌ی مشتری یک MAC از پیوست تمامی پیام‌های دست‌دهی که ارسال و دریافت نموده است را گسیل می‌نماید. برنامه‌ی سرور به مقایسه‌ی این MAC و آن MAC که مربوط به پیام‌های دست‌دهی که آن نیز آن‌ها را ارسال و دریافت نموده است، می‌پردازد. در صورت وجود اختلاف برنامه‌ی سرور می‌تواند به ارتباط پایان دهد. به‌گونه‌ی مشابه برنامه‌ی سرور یک MAC از پیام‌ها دست‌دهی را که ملاحظه نموده است ارسال می‌دارد و به برنامه‌ی مشتری اجازه می‌دهد اختلاف‌ها را ارزیابی و آزمایش نماید.

ممکن است از این موضوع در شگفت باشید چرا در مراحل (۱) و (۲) پیام‌های موقت وجود دارند؟ آیا شمارگان ردیف به‌منظور جلوگیری از تهاجم از نوع بازپخش قطعه کافی نمی‌باشند؟ پاسخ آری است اما آن‌ها به تنهایی قادر به جلوگیری از نوع **تهاجم ارتباط بازپخش** نمی‌باشند. تهاجم ارتباط بازپخش ذیل را درنظر بگیرید: فرض کنید مهاجم T همه‌ی پیام‌های مبادله شده بین Alice و Bob را شنود نماید. روز بعد T خود را به جای Bob معرفی می‌کند و همان مجموعه پیام ارسالی روز قبل از سوی Bob را به Alice گسیل می کند. اگر Alice از پیام‌های موقت استفاده نکند پاسخ وی دقیقاً همان مجموعه پیامی است که وی روز قبل برای Bob ارسال نموده است. از آن‌جائیکه هریک از پیام‌های دریافتی از سوی Alice از نقطه نظر آزمایش یکپارچگی معتبر می‌باشند بنابراین وی مشکوک به حالت غیرعادی در شبکه نخواهد شد. اگر Alice یک سرور تجارت الکترونیکی است وی ممکن است براین تصور باشد که Bob درخواست سفارش دوم را برای همان قلم کالا دارد. از طرف دیگر با افزودن یک پیام موقت در پروتکل، Alice پیام‌های موقتی گوناگونی را برای هر تعامل TCP ارسال می‌کند که سبب تعریف کلیدهای رمز مختلف در دو روز متفاوت می‌شود. بنابراین هنگامی که Alice رکوردهای SSL بازپخش شده را از T دریافت می‌دارد این رکوردها از نقطه‌ی عبور از صافی یکپارچگی مردود می‌شوند و آلودگی تراکنش تجارت الکترونیکی موفق به دستیابی اهداف مخرب خود نخواهد شد. به اختصار SSL پیام‌های موقت را به‌منظور دفاع در برابر تهاجم ارتباط بازپخش مورد استفاده قرار می‌دهد و شمارگان ردیف را با هدف دفاع در برابر احتمال بازپخش بسته‌های منفرد در طول یک تعامل در جریان به‌کار می‌گیرد.

- **انسداد ارتباط**

در مقطعی از عملیات Bob و یا Alice ممکن است مایل به خاتمه‌ی تعامل SSL باشند. یک رویکرد ممکن است به Bob اجازه‌ی قطع تعامل SSL را از طریق تحریک ارتباط TCP به وسیله‌ی ارسال قطعه‌ی TCPFIN از سوی Bob به طرف Alice بدهد. اما چنین واکشی سبب **تهاجم لحظه‌ای کوتاه**[1] خواهد شد که T را قادر می‌سازد اقدام به یک تهاجم فرد واسطی برروی یک SSL جاری نماید و تعامل را با یک TCPFIN خاتمه دهد. اگر T چنین حرکتی را انجام دهد Alice بر این باور خواهد شد که وی تمام داده‌های Bob را دریافت داشته است در حالیکه او فقط بخشی از آن‌ها را دستیابی نموده است. راهکار این چالش بیان این نکته در دامنه‌ی **نوع** است که آیا رکورد به‌منظور خاتمه‌ی تعامل SSL می‌باشد یا خیر. بدیهی است نوع SSL در قالب متن به صورت آشکار یا خام ارسال می‌شود و از سوی گیرنده با بهره‌گیری از MAC متعلق به رکورد اعتبارسنجی می‌شود. با درنظر گرفتن چنین دامنه‌ای اگر Alice موفق به دریافت یک TCPFIN قبل از دریافت یک رکورد انسداد SSL گردد او آگاه می‌شود که فعالیت غیرعادی در حال وقوع می‌باشد. در این مقطع بحث معرفی SSL به انتها می‌رسد.

۸-۱۳ امنیت لایه‌ی شبکه (VPN , IPsec)

پروتکل امنیت IP که اختصاراً IPsec نامیده می شود امنیت لایه‌ی شبکه را تأمین می‌کند. پروتکل IPsec امنیت داده گرام‌ها را بین هر دو موجودیت در دو سوی لایه‌ی شبکه از جمله میزبان‌ها و

1- Truncation attack

مسیریاب‌ها تأمین می‌نماید. اغلب نهادهای دولتی و خصوصی، تجاری و یا فرهنگی، انتفاعی و غیرانتفاعی از پروتکل IPsec به‌منظور ایجاد **شبکه‌های خصوصی مجازی (VPN)** که از طریق اینترنت عمومی قابل پیاده‌سازی می‌باشند، استفاده می‌کنند.

قبل از ورود به جزئیات IPsec گامی به عقب نهاده و مفهوم تأمین محرمانگی در لایه‌ی شبکه را بررسی می‌کنیم. محرمانگی در لایه‌ی شبکه بین یک جفت موجودیت شبکه از جمله بین دو روتر، بین دو میزبان و یا بین یک روتر و یک میزبان امکان‌پذیر می‌باشد. یک موجودیت ارسال کننده رمزگذاری محموله‌ی همه‌ی داده‌گرام‌ها را انجام و آن‌ها را به موجودیت دریافت کننده گسیل می‌دارد. محموله‌ی رمز می‌تواند یک قطعه‌ی TCP، یک قطعه‌ی UDP و یا یک پیام ICMP و غیره باشد. در صورت وجود چنین امکاناتی در لایه‌ی شبکه همگی داده‌های ارسالی از یک موجودیت به دیگری شامل ایمیل‌ها، صفحه‌های وب، پیام‌های دست‌دهی TCP و پیام‌های مدیریت مانند ICMP و SNMP از نظر موجودیت ثالثی که ممکن است در حال شنود از شبکه باشد، پنهان می‌گردند. برای همین منظور امنیت لایه‌ی شبکه اصطلاحاً یک پوشش فراگیر نامیده می‌شود.

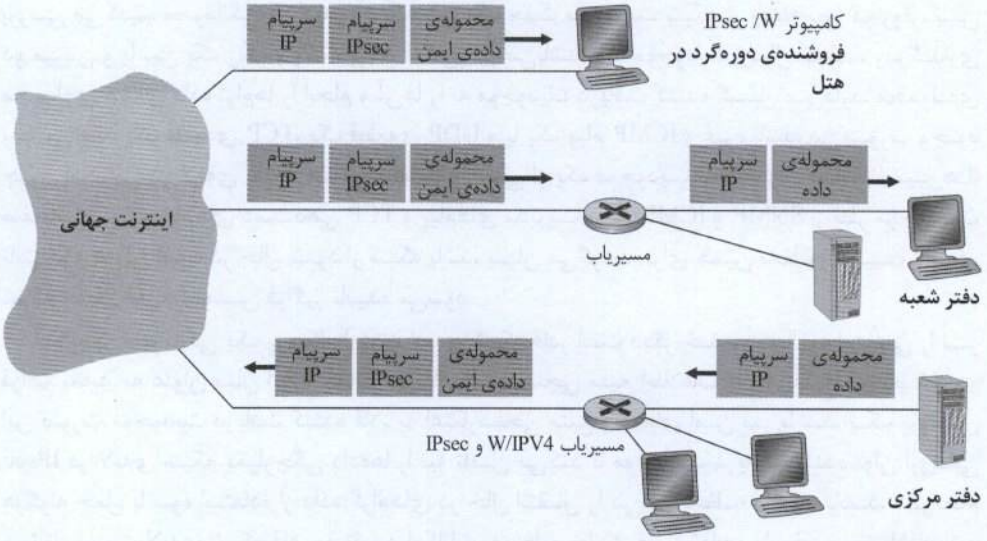
علاوه بر محرمانگی یک پروتکل امنیت لایه‌ی شبکه قادر است دیگر خدمات بالقوه امنیتی را نیز فراهم نماید. به عنوان مثال این امکانات می‌تواند اعتبارسنجی منبع اطلاعات را نیز تأمین نماید که در این صورت موجودیت دریافت کننده قادر به اعتبارسنجی منبع داده‌های امن می‌باشد. یک پروتکل IPsec در لایه‌ی شبکه یکپارچگی داده‌ها را نیز تأمین می‌کند تا موجودیت دریافت کننده توان ارزیابی هرگونه جعل یا سوء استفاده از داده گرام‌های در حال انتقال را در هر لحظه داشته باشد. سرانجام پروتکل امنیت لایه‌ی شبکه قادر به تأمین امکانات به‌منظور جلوگیری از تهاجم بازپخش می‌باشد که Bob را به عنوان مثال قادر به تشخیص هر نوع نسخه برداری از داده گرام‌ها به‌وسیله‌ی مهاجمان می‌نماید. نتیجه آن‌که پروتکل IPsec محرمانگی، اعتبارسنجی منبع، یکپارچگی داده و ممانعت از تهاجم بازپخش را تأمین می‌کند.

- **پروتکل IPsec و شبکه‌های مجازی خصوصی (VPNS)**

یک نهاد بزرگ در چندین منطقه‌ی جغرافیایی نیازمند شبکه‌ی IP اختصاصی خویش می‌باشد تا میزبان‌ها و سرورهای آن قادر به مبادله‌ی داده‌ها بین یکدیگر در چارچوب ایمن و محرمانه باشند. برای دستیابی به این هدف آن نهاد می‌تواند از یک شبکه‌ی فیزیکی مستقل شامل روترها، پیوندها و یک زیرساخت DNS برخوردار گردد که آن نهاد را کاملاً از شبکه‌ی اینترنت عمومی متمایز نماید. چنین شبکه‌ی غیرمتصل به اینترنت عمومی و اختصاصی را **شبکه‌ی خصوصی** می‌نامند. بدیهی است چنین شبکه‌ای بسیار پرهزینه می‌باشد زیرا یک نهاد مجبور به خرید، نصب، راه‌اندازی و نگهداری زیرساخت این شبکه می‌باشد.

به جای بهره‌گیری از یک شبکه‌ی خصوصی بسیاری از نهادها در حال حاضر به تعریف VPNS از طریق اینترنت عمومی می‌پردازند. با یک VPN ترافیک درونی آن نهاد به جای آنکه از طریق یک شبکه‌ی فیزیکی مستقل عبور نماید از امکانات اینترنت عمومی استفاده می‌کند. اما تأمین محرمانگی برای ترافیک درونی یک نهاد از طریق رمزگذاری آن ترافیک قبل از ورود به اینترنت صورت می‌پذیرد. یک نمونه‌ی ساده از یک VPN در شکل ۸-۲۸ نشان داده شده است. در این شکل نهاد موردنظر شامل یک دفتر مرکزی، یک شعبه و تعدادی فروشنده‌ی دوره‌گرد است که دستیابی به اینترنت را از طریق اتاق هتل خود انجام می‌دهند. در این شکل فقط یک فروشنده نشان داده شده است. در این VPN

هرگاه دو میزبان در محل دفتر مرکزی داده گرام‌های IP را به یکدیگر ارسال می‌دارند و یا هرگاه دو میزبان در دفتر شعبه نیازمند ارتباط می‌باشند از امکانات قدیمی IPV4 که بدون خدمات IPsec می‌باشد استفاده می‌کنند. هنگامی که میزبان‌های دو نهاد مایل به ارتباط از طریق یک مسیر اینترنت عمومی می‌باشند آن‌گاه ترافیک نخست رمزگذاری و سپس وارد اینترنت می‌شود.



شکل ۸-۲۸♦ شبکه مجازی خصوصی (VPN)

به‌منظور درک بهتر از چگونگی عملکرد VPN یک مثال ساده را در چارچوب شکل ۸-۲۸ دنبال می‌کنیم. هنگامی که از محل دفتر مرکزی شرکت یک داده گرام IP به فروشنده‌ای در یک هتل ارسال می شود روتر دریچه‌ی مسیریابی در دفتر مرکزی داده گرام IPV4 را به یک داده گرام IPsec تبدیل و سپس این داده گرام IPsec به درون اینترنت گسیل می‌شود. این داده گرام IPsec در حقیقت دارای یک سرپیام IPV4 سنتی است تا مسیریاب‌های موجود در اینترنت عمومی پردازش این داده گرام را همانند یک داده گرام IPV4 معمولی انجام دهند. برای آن‌ها این داده گرام کاملاً معمولی و بدون عیب است در حالیکه همان‌گونه که شکل ۸-۲۸ نشان می‌دهد محموله‌ی داده گرام IPsec شامل یک سرپیام IPsec می‌باشد که از آن به‌منظور پردازش لازم IPsec استفاده می‌شود. هم‌چنین محموله‌ی داده گرام IPsec رمزگذاری شده است. هنگامی که این داده گرام وارد رایانه‌ی همراه فروشنده در اتاق هتل وی می‌شود سیستم عامل (OS) آن رایانه محموله را رمزگشایی می‌کند و خدمات امنیتی موردنیاز مانند اعتبارسنجی یکپارچگی داده‌ها را برای آن تأمین می‌نماید. پس از آن محموله‌ی رمزگشایی شده به پروتکل لایه‌ی بالاتر مانند TCP یا UDP تحویل می‌شود. مطالب اشاره شده تصویر ساده‌ای از VPNs را نشان می‌دهد اکنون موضوع را عمیق‌تر دنبال می‌کنیم.

پروتکل‌های AH و ESP

پروتکل IPsec پیچیدگی بسیاری را در درون خود حمل می‌کند و در بیش از ده‌ها RFC توصیف شده است. دو مستند RFC مهم در ارتباط با IPsec مستندهای [RFC4301] که معماری کلی امنیت IP را تشریح می‌کند و [RFC2411] می‌باشند که چارچوب مجموعه پروتکل IPsec را توصیف می‌کند.

هدفی که در این بخش دنبال می‌شود تحلیل راهکارهای عملی IPsec می‌باشد و به تفسیر تجربی این مستندها نمی‌پردازد. در مجموعه‌ی پروتکلی IPsec دو پروتکل بنیادی را می توان ملاحظه نمود:

(۱) پروتکل **سرپیام اعتبارسنجی (AH)**[1] و (۲) پروتکل **بسته‌بندی امنیتی محموله (ESP)**[2].

هنگامی که یک منبع IPsec مانند یک میزبان یا مسیریاب داده گرام‌های ایمنی را به موجودیتی در مقصد ارسال می‌کند که آن نیز یک میزبان یا روتر است این مهم را با بهره‌گیری از پروتکل AH و یا ESP انجام می‌دهد. پروتکل AH اعتبارسنجی منبع و یکپارچگی داده را انجام می‌دهد بدون آنکه محرمانگی را تأمین نماید. پروتکل ESP اعتبارسنجی منبع، یکپارچگی داده و محرمانگی را تأمین می‌نماید. محرمانگی برای VPN در اغلب موارد حیاتی می‌باشد و برای دیگر کاربردهای IPsec نیز پروتکل ESP کاربرد به مراتب گسترده‌تری از پروتکل AH دارد. بنابراین در این بخش از فصل ۸ تمرکز بیش‌تر برروی پروتکل ESP می‌باشد.

- **وابستگی‌های امنیتی**

در پروتکل IPsec داده‌گرام‌ها مابین زوج موجودیت‌های شبکه مبادله می‌شوند. این زوج می‌توانند دو میزبان، دو مسیریاب و یا یک میزبان و یک روتر باشند. قبل از ارسال داده گرام‌های IPsec از منبع به مقصد، موجودیت‌های منبع و مقصد یک ارتباط منطقی لایه‌ی شبکه را تشکیل می‌دهند. این ارتباط منطقی **وابستگی امنیتی (SA)**[3] نامیده می‌شود. یک SA یک ارتباط منطقی یک طرفه (simplex) است که فقط از منبع به مقصد جاری می‌شود. اگر دو موجودیت مایل به ارسال داده گرام‌های امن مابین خود باشند آن‌گاه دو SA یا به بیان دیگر دو ارتباط منطقی لازم می‌باشد هر SA برای یک سوی ارتباط.

به عنوان مثال بار دیگر نهاد شکل ۸-۲۸ را مجسم نمائید. این نهاد شامل سه بخش دفتر مرکزی، شعبه و تعداد n فروشنده‌ی دوره‌گرد است. به‌منظور ساده شدن موضوع فرض کنید ترافیک IPsec دوطرفه به جای یک طرفه می‌باشد. این دو طرفه بودن ارتباط بین دفتر مرکزی و شعبه و نیز دفتر مرکزی و فروشندگان است. در این VPN چه تعداد SA وجود دارد؟ برای پاسخ به این پرسش به این نکته بایستی توجه داشت که تعداد دو SA بین مسیریاب دریچه‌ی دفتر مرکزی و روتر دریچه‌ی شعبه هرکدام برای یک سوی ارتباط وجود دارند. برای رایانه‌ی همراه هر فروشنده نیز دو SA بین مسیریاب دریچه‌ی دفتر مرکزی و هر رایانه‌ی همراه تعریف شده است. بنابراین مجموع SA های موردنیاز معادل (2+2n) می‌باشد. بدیهی است تمامی ترافیک ارسالی به اینترنت عمومی از سوی مسیریاب‌های دریچه و یا رایانه‌های همراه از لحاظ امنیتی حالت IPsec ندارند. به عنوان مثال یک میزبان در دفتر مرکزی ممکن است مایل به دستیابی یک وب سرور مانند آمازون و یا گوگل مستقر در اینترنت عمومی باشد. در این حالت مسیریاب دریچه و رایانه‌ی همراه فروشنده دو نوع داده گرام IPV4 و IPsec را به درون اینترنت عمومی انتشار می‌دهند.

1- Authentication header

2- Encapsulation security payload

3- Security association



شکل ۸-۲۹♦ وابستگی امنیتی (SA) از مسیریاب R1 به R2

اکنون درون یک SA تحلیل می‌شود. به‌منظور درک آسان‌تر موضوع این مهم در چارچوب یک SA از مسیریاب R1 به روتر R2 در شکل ۸-۲۹ تحلیل می‌شود. مسیریاب R1 اقدام به نگهداری اطلاعات وضعیت SA به شرح ذیل می‌نماید:

- تعریف یک شناسه‌ی 32 بیت برای SA که به آن **شاخص پارامتری امنیت (SPI)**[1] گفته می‌شود.
- واسط اصلی SA در این حالت (شکل ۸-۲۹) آدرس IP:200.168.1.100 و واسط مقصد SA در این حالت آدرس IP:193.68.2.23 را اختیار کرده است.
- نوع رمزگذاری مورد استفاده مانند 3DES با CBC
- کلید رمز
- نوع اعتبارسنجی یکپارچگی مانند HMAC با MD5
- کلید اعتبارسنجی

هنگامی که مسیریاب R1 نیازمند تعریف یک داده گرام IPsec به‌منظور ارسال از طریق این SA می‌باشد، این روتر اطلاعات وضعیت را دستیابی می‌کند تا از چگونگی انجام اعتبارسنجی و رمزگذاری داده گرام‌ها آگاه شود. به‌گونه‌ی مشابه مسیریاب R2 همان اطلاعات وضعیت را برای این SA نگهداری می‌کند و از این اطلاعات به‌منظور اعتبارسنجی و رمزگشایی هر داده گرام IPsec که به SA وارد می‌شوند استفاده می‌کند. یک موجودیت IPsec مانند میزبان یا مسیریاب نگهداری وضعیت، اطلاعات تعداد زیادی SA را برعهده می‌گیرد. به عنوان مثال در VPN شکل ۸-۲۸ با تعداد n فروشنده، مسیریاب دریچه‌ی دفتر مرکزی اطلاعات وضعیت (2+2n) تعداد SA را نگهداری می‌نماید. یک موجودیت IPsec اطلاعات وضعیت برای همه‌ی IPهای خود را در یک بانک اطلاعاتی به نام **پایگاه وابستگی امنیتی (SAD)**[2] ذخیره می‌کند. این پایگاه در هسته‌ی سیستم عامل تعریف می‌شود.

۸-۱۴ داده گرام IPsec

پس از تشریح SA و عملکرد آن، در این مقطع به توصیف داده گرام‌های واقعی IPsec می‌پردازیم.

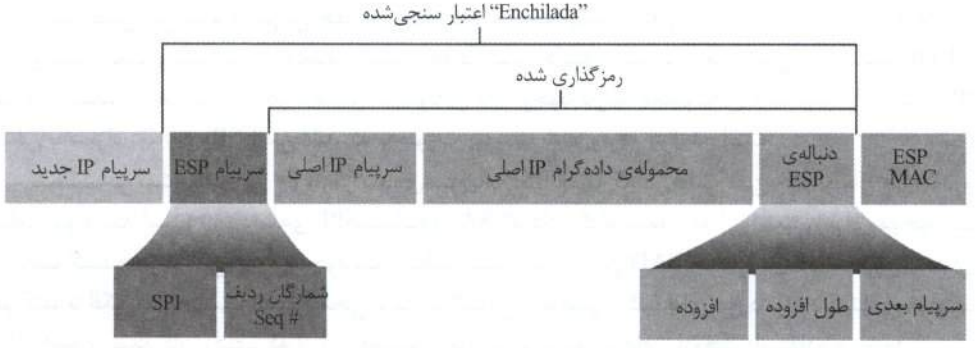
1- Security parameter index

2- Security association database

پروتکل IPsec دارای دو قالب بسته‌ی گوناگون می‌باشد یکی از این دو **حالت تونل**[1] و دیگری **حالت انتقال**[2] می‌باشد. روش تونل به‌منظور کاربرد VPN مناسب‌تر و بنابراین به‌صورت گسترده‌ای مورد استفاده قرار می‌گیرد. به همین دلیل در این زیربخش بیش‌ترین تمرکز بر روش تونل IPsec می‌باشد.

قالب یک بسته‌ی IPsec در شکل ۸-۳۰ نشان داده شده است. در چارچوب شکل ۸-۲۹ دامنه‌های بسته‌ی IPsec تحلیل می‌شوند. فرض کنید مسیریاب R1 یک داده گرام معمولی IPV4 را از میزبان 172.16.1.17 در شبکه‌ی دفتر مرکزی دریافت می‌دارد که مقصد آن میزبان 172.16.2.48 می‌باشد و در شبکه‌ی دفتر شعبه استقرار دارد. مسیریاب R1 با بهره‌گیری از مراحل ذیل داده گرام IPV4 اصلی را تبدیل به یک داده گرام IPsec می‌نماید:

- به انتهای داده گرام IPV4 اصلی که شامل دامنه‌های سر پیام اصلی نیز می‌باشد یک دامنه‌ی دنباله‌ی ESP پیوست می‌کند.
- نتیجه‌ی حاصل را با بهره‌گیری از الگوریتم و کلید مشخص شده در SA رمز می‌کند.
- به ابتدای (پیشانی) این بسته‌ی رمز شده یک دامنه با نام سرپیام ESP پیوست می‌کند. بسته‌ی نتیجه شده اصطلاحاً "enchilada" به مفهوم کیک ذرت مکزیکی نامیده می‌شود.
- یک MAC اعتبارسنجی را برروی تمامیت enchilada با بهره‌گیری از الگوریتم و کلید مشخص شده در SA تعریف می‌کند.
- اطلاعات MAC را به انتهای enchilada پیوست می‌کند تا محموله‌ی داده را تشکیل دهد.
- تعریف یک سرپیام جدید IP به‌همراه همه‌ی دامنه‌های سرپیام کلاسیک IPV4 که معمولاً دارای طولی معادل 20 بایت است و پیشاپیش محموله‌ی داده پیوست می‌شود.



شکل ۸-۳۰ ♦ قالب داده گرام **IPsec**

بدیهی است داده گرام IPsec نتیجه‌ی داده گرام IPV4 است که به دنبال سرپیام‌های آن محموله قرار می‌گرفت. داده گرام‌های IPsec این تفاوت را دارند که محموله شامل یک سرپیام ESP، داده گرام IP اصلی، یک دنباله‌ی ESP و یک اعتبارسنجی ESP است که داده گرام اصلی و دنباله‌ی ESP در آن رمز شده‌اند. در مثال شکل ۸-۲۹ داده گرام اصلی IP دارای آدرس IP مبدأ به‌صورت: 172.16.1.17 و

1- tunnel mode
2- transport mode

172.16.2.48 برای مقصد می‌باشد. از آن‌جائیکه داده گرام IPsec شامل داده گرام IP اصلی است، این آدرس‌های به‌صورت رمزبخشی از محموله‌ی بسته‌ی IPsec به‌شمار می‌آیند. اکنون پرسش مطرح آن است که آدرس‌های IP مبدأ و مقصد در سرپیام IP جدید چه مفهومی دارند به عبارت دیگر در انتهای سمت چپ سرپیام داده گرام IP چه اطلاعاتی نهفته است؟ شاید حدس زده باشید آن‌ها هدف خود را واسطه‌های مسیریاب‌های مبدأ و مقصد در دو انتهای تونل‌ها قرار داده‌اند. در شکل ۸-۲۹ این دو آدرس عبارتند از:

200.168.1.100 برای واسط مبدأ 193.68.2.23 برای واسط مقصد. بدیهی است شماره‌ی پروتکل در این سرپیام IPV4 جدید به‌سوی TCP، UDP و یا SMTP هدف‌گیری نشده‌اند بلکه با ارزش 50 که مفهوم آن یک داده گرام IPsec با بهره‌گیری از پروتکل ESP است مقداردهی شده است.

پس از آنکه R1 داده گرام IPsec را به درون اینترنت عمومی تزریق می‌کند داده گرام قبل از رسیدن به روتر مقصد R2 از درون تعداد زیادی مسیریاب دیگر عبور می‌نماید. هریک از روترها داده گرام IPsec را مانند یک داده گرام معمولی پردازش می‌کنند آن‌ها کاملاً از این حقیقت که این داده گرام حاصل داده‌ی رمز شده‌ی IPsec است ناآگاه می‌باشند. برای این روترهای اینترنت عمومی مقصدنهایی داده گرام مسیریاب R2 می‌باشد زیرا آن‌چه آن‌ها ملاحظه می‌کنند آدرس IP مقصد در سرپیام خارجی R2 می‌باشد.

با توجه به بررسی مثال ساختاری یک داده گرام IPsec، اکنون به بررسی عمیق‌تر درون داده گرام enchilada یا همان **کیک ذرت*** می‌پردازیم. در شکل ۸-۳۰ ملاحظه شد دنباله‌ی ESP از سه دامنه‌ی : **افزوده‌ها** (padding)، **طول افزوده‌ها و سرپیام بعدی** تشکیل شده است. یادآوری این نکته ضروری است که توده رمزها نیازمند آن هستند که پیام رمز شده ضریب کاملی از طول توده باشد. افزودنی‌ها در این حالت شامل بایت‌های بدون مفهوم هستند که با افزودن به داده گرام اصلی به‌همراه طول افزوده‌ها و سرپیام بعدی **پیامی** را تعریف می‌کنند که یک ارزش کامل از توده‌ها می‌باشد. دامنه‌ی طول افزوده‌ها، حامل این پیام در درون خود می‌باشد که به موجودیت دریافت کننده چه مقدار افزودنی تزریق شده است و بنابراین در مقصد بایستی حذف شوند. دامنه‌ی سرپیام بعدی نوع داده مانند UDP و غیره را شناسایی می‌کند که در دامنه‌ی محموله‌ی داده وجود دارند. محموله‌ی داده نیز داده گرام IP اصلی به‌همراه دنباله‌ی ESP می‌باشد که به‌صورت پیوست شده و رمز درآمده است.

پیوست در پیشاپیش این جزء رمز شده‌ی سرپیام ESP است که در قالب متن خام ارسال می‌شود و شامل دو دامنه است: (۱) دامنه‌ی SPI شناسه‌ی SA که داده گرام متعلق به آن است را برای موجودیت دریافت کننده آشکار می‌سازد. موجودیت دریافت کننده پس از آن SAD خود را با SPI شاخص‌گذاری می‌کند تا الگوریتم مناسب اعتبارسنجی و یا رمزگشایی را به‌همراه کلیدهای مورد نیاز دستیابی نماید. (۲) دامنه‌ی شمارگان ردیف که از آن به‌منظور دفاع در برابر تهاجم‌های بازپخش استفاده می‌شود.

موجودیت فرستنده هم‌چنین یک MAC اعتبارسنجی را نیز پیوست می‌کند. قبلاً اشاره گردید ارسال کننده یک MAC را برای تمامیت enchilada شامل سرپیام ESP، داده گرام IP اصلی و دنباله‌ی ESP محاسبه می‌کند که در آن داده گرام و دنباله رمز می‌شوند. یادآوری می‌شود برای محاسبه‌ی MAC، فرستنده یک کلید رمز MAC را به enchilada پیوست می‌کند و سپس طول ثابت هاش را از نتیجه‌ی به‌دست آمده محاسبه می‌کند.

---

* داده گرام IPsec را enchilada یا کیک ذرت مکزیکی نامیده‌اند زیرا این کیک دارای روکش آرد ذرت است که درون آن به‌صورت نهان انواع ادویه‌ها و مواد خوراکی گوشتی و سبزیجات قرار دارد.

هنگامی که R2 داده گرام IPsec را دریافت می‌کند، پی می‌برد که خود آدرس IP مقصد داده گرام می‌باشد بنابراین داده گرام را پردازش می‌کند. از آن‌جائیکه دامنه‌ی پروتکل در انتهای چپ سرپیام IP دارای ارزش 50 می‌باشد R2 مشاهده می‌کند که بایستی پردازش IPsec-ESP را به داده گرام اعمال نماید. ابتدا با ورود به enchilada مسیریاب R2 از SPI بهره می‌گیرد تا دریابد داده گرام به کدام SA تعلق دارد. دوم، به محاسبه‌ی MAC مربوط به enchilada می‌پردازد و آن‌گاه سازگاری MAC با ارزش درون دامنه‌ی ESP را ارزیابی می‌کند در صورت تأئید مطمئن می‌شود که کیک ذرت از جانب R1 گسیل شده است بدون آنکه در محتوای آن جعلی صورت گرفته باشد. سوم، دامنه‌ی شمارگان ردیف را ارزیابی می‌کند تا اطمینان از عدم بازپخش داده گرام حاصل نماید. چهارم، رمزگشایی بخش رمز شده‌ی داده گرام را انجام و برای این مهم از الگوریتم و کلید وابسته به SA بهره می‌گیرد. پنجم، افزوده‌های داده گرام را حذف و متن اصلی را که همان داده گرام IP می‌باشد بازیابی می‌کند. سرانجام به عنوان حرکت ششم، داده گرام اصلی را به درون شبکه‌ی شعبه‌ی شرکت تزریق و آن را به‌سوی مقصد نهایی هدایت می‌کند. ملاحظه می‌شود تمامی این فرآیندها مانند آماده‌سازی enchilada اصلی بسیار پیچیده و دشوار می‌باشند. به همین دلیل نام کیک ذرت (enchilada) برای این فرآیند انتخاب شده است.

بدیهی است نکته مهم دیگری نیز وجود دارد که نیازمند مطالعه‌ی دقیق می باشد. این مهم به این گونه است که: هنگامی که R1یک داده گرام محافظت نشده را از یک میزبان شبکه‌ی دفتر مرکزی دریافت می‌دارد و آن داده گرام به‌سوی آدرس IP در خارج شبکه‌ی دفتر مرکزی گسیل شده است، چگونه R1 آگاه می‌شود که آیا بایستی داده گرام را تبدیل به یک پیام IPsec نماید؟ اگر پردازش از سوی IPsec بایستی انجام پذیرد چگونه R1 از SA مورد نیاز در بین تعداد انبوهی از SAها جهت ساختن IPsec آگاه می شود؟ پاسخ این پرسش‌ها چنین است: به‌همراه SAD موجودیت IPsec ساختمان داده‌ی دیگری را نگهداری می‌کند که اصطلاحاً **پایگاه سیاست امنیتی (SPD)**[1] نامیده می‌شود. این پایگاه SPD نوع داده گرام‌ها را به‌صورت تابعی از یک آدرس IP مبدأ و مقصد و نوع پروتکلی که بایستی مورد پردازش IPsec واقع شوند مشخص می‌نماید و برای آن گروه از داده گرام‌هایی که نیازمند پردازش IPsec می‌باشند SA مورد استفاده را نیز معین می‌کند. به بیان دیگر اطلاعات درون یک SPD مشخص می‌کند با یک داده گرام ورودی چه عملی بایستی صورت پذیرد و اطلاعات درون SAD چگونگی عملیات را تعریف می‌کند.

اکنون این پرسش مطرح می‌شود که IPsec چه نوع خدماتی را تأمین می‌نماید؟ خدمات از این دست را از نقطه نظر یک مهاجم مانند T ارزیابی می‌کنیم که یک مهاجم، فرد واسطی است و در نقطه‌ای مابین R1 و R2 در شکل ۸-۲۹ قرار گرفته است. در سراسر این بخش فرض می‌شود T از کلیدهای اعتبارسنجی و رمز مورد استفاده‌ی SA آگاهی ندارد. چه کنش‌هایی را T می‌تواند انجام دهد و چه حرکت‌هایی را نمی‌تواند؟ نخست، او نمی‌تواند داده‌ی اصلی را ملاحظه نماید. در حقیقت نه تنها داده در داده گرام اصلی از T پنهان می‌باشد بلکه شماره‌ی پروتکل، آدرس IP مبدأ و آدرس IP مقصد نیز برای او نهان است. آن‌چه از داده گرام ارسالی از طریق SA برای مهاجم T آشکار است این بخش از اطلاعات است که داده گرام از میزبان آدرس 172.16.1.0/24 سرچشمه گرفته است و به سوی میزبانی با آدرس 172.16.2.0/24 گسیل شده است. او از این موضوع که آیا این داده گرام حامل TCP، UDP و یا داده‌ی ICMP می‌باشد و یا چیز دیگری؟ آگاهی ندارد. او از این موضوع نیز که داده گرام حامل

1- Security policy database

HTTP، SMTP و یا نوع دیگری از داده‌های کاربردی است ناآگاه می‌باشد. ملاحظه می‌شود این محرمانگی بسیار فراتر از SSL می‌باشد. دوم، فرض کنید T تلاش درجعل یا دستکاری یک داده گرام در SA از طریق معکوس نمودن برخی از بیت‌ها را دارد. هنگامی که چنین داده گرام جعل شده‌ای وارد R2 می‌شود از نقطه نظر آزمایش یکپارچگی مردود می‌شود زیرا از MAC برای منظور اعتبارسنجی یکپارچگی استفاده شده است. بنابراین بار دیگر مانعی در برابر تهاجم T وجود دارد. سوم، فرض کنید T تلاش در گمراه نمودن از لحاظ جایگذاری خود به جای یکی از میزبان‌ها را دارد. از آن‌جائیکه R1 یک داده گرام IPsec با آدرس‌های به‌ترتیب مبدأ و مقصد: 200.168.1.100 و 193.68.2.23 را تعریف می‌کند بازهم تهاجم T بی‌اثر خواهد شد زیرا این داده گرام نیز از لحاظ ارزیابی یکپارچگی در مسیریاب R2 مردود می‌شود. سرانجام به دلیل آنکه IPsec شامل شمارگان ردیف است، مهاجم T قادر به تهاجم موفق بازپخش پیام نیز نخواهد شد. نتیجه آنکه همان‌گونه که در ابتدای این بخش ادعا شد پروتکل IPsec به‌منظور پردازش بسته‌ها بین هر زوج از تجهیزات در لایه‌ی شبکه محرمانگی، اعتبارسنجی مبدأ، یکپارچگی داده و جلوگیری از تهاجم بازپخش را تأمین می‌نماید.

۸-۱۵ مدیریت کلید در IPsec: (IKE)

هنگامی که یک VPN تعداد اندکی نقطه‌ی انتهایی مانند فقط دو مسیریاب شکل ۸-۲۹ را در اختیار دارد، مدیر شبکه به صورت دستی اطلاعات SA شامل الگوریتم‌های رمزگذاری و اعتبارسنجی و کلید و نیز SPIها را در درون SAD نقاط انتهایی تعریف می‌کند. بدیهی است اینگونه اعمال کلیدهای رمز برای یک VPN بزرگ غیرممکن می‌باشد زیرا ممکن است شامل صدها و یا هزاران مسیریاب IPsec و میزبان‌ها باشد. کاربردهای انبوه و از لحاظ توزیع جغرافیایی گسترده، نیازمند روش‌های خودکار برای تعریف SAها می‌باشند. پروتکل IPsec این مهم را از طریق پروتکل **تبادل کلید اینترنت (IKE)** که در مستند [RFC4306] تعریف شده است، انجام می‌دهد.

پروتکل IKE دارای وجوه تشابه یا دست‌دهی SSL نیز می باشد. هر موجودیت IPsec دارای یک گواهی است که شامل کلید عمومی موجودیت است. همانند SSL پروتکل IKE از دو کلید گواهی موجودیت بهره می‌گیرد. یکی از این کلیدها اعتبارسنجی را انجام می‌دهد و دیگری الگوریتم‌های رمزگذاری را تعریف می‌کند و نیز اطلاعات کلید را به‌صورت محرمانه به‌منظور تعریف کلیدهای تعامل در پروتکل IPsec هریک از SAها مورد استفاده قرار می‌دهد. برخلاف SSL پروتکل IKE از دو مرحله در چارچوب دو مسیریاب R1 و R2 در شکل ۸-۲۹ بهره می‌گیرد. مرحله‌ی نخست شامل دو مبادله از جفت پیام‌های بین R1 و R2 می‌باشد:

۱- در طول انجام نخستین مبادله‌ی پیام‌ها دو طرف ارتباط از الگوریتم دیفی – هلمن به‌منظور تعریف یک **IKE-SA** دوطرفه بین روترها استفاده می‌کنند. این موجودیت IKE-SA دوطرفه کاملاً متفاوت با IPsec-SA می‌باشد که در دو بخش ۸-۱۳ و ۸-۱۴ بررسی شده‌اند. موجودیت IKE-SA یک کانال اعتبارسنجی و رمزگذاری شده را بین دو روتر تأمین می‌کند. در طول انجام این مهم، نخست تبادل جفت پیام انجام می‌شود و کلیدهای رمزگذاری و اعتبارسنجی نیز برای IKE-SA تعریف می‌شوند. علاوه بر این موارد یک رمز اصلی به‌منظور محاسبه‌ی کلیدهای IPsec-SA که در مرحله‌ی دوم انجام می‌شود مورد بهره‌برداری قرار می‌گیرد. بدیهی است در طول مرحله‌ی نخست کلیدهای عمومی و خصوصی RSA مورد استفاده قرار نمی‌گیرند. به‌ویژه آنکه هیچ‌یک از روترهای R1 و R2 شناسه‌ی خود را به دلیل بهره‌گیری از امضای دیجیتالی یک پیام با کلید خصوصی خود آشکار نمی‌سازد.

۲- در طول انجام مرحله‌ی دوم تبادل پیام‌ها، هر دو طرف شناسه‌ی خود را برای دیگری از طریق امضای پیام‌های خویش آشکار می‌سازند. بدیهی است این شناسه‌ها برای یک شنود کننده‌ی خنثی آشکار نمی‌شود زیرا پیام‌ها از طریق یک کانال محافظت شده‌ی IKE-SA ارسال می‌شود. در طول انجام این مرحله دوطرف، الگوریتم‌های رمزگذاری IPsec و اعتبارسنجی را که بایستی از سوی IPsec-SA مورد استفاده قرار گیرد، معرفی می‌کنند.

در مرحله‌ی دوم IKE، دو طرف یک SA را برای هر سمت مبادله تعریف می‌کند. در پایان مرحله‌ی دوم کلیدهای تعامل رمزگذاری و اعتبارسنجی از سوی دو طرف برای دو SA تعریف می‌شوند. پس از آن دو طرف قادر به بهره‌گیری از SAها به‌منظور ارسال داده گرام‌های ایمن برمبنای دو بخش ۸-۱۳ و ۸-۱۴ می‌شوند. فلسفه‌ی بنیادی اختیار دو مرحله در IKE هزینه‌ی محاسبه‌ی آن می‌باشد. از آن‌جائیکه مرحله‌ی دوم شامل هیچ‌گونه کلید عمومی رمزگذاری نمی‌شود، IKE قادر به تولید تعداد انبوهی از SAها بین دو موجودیت IPsec با هزینه‌ی نسبتاً اندکی می‌باشد.

۸-۱۶ تأمین امنیت شبکه‌های محلی بی‌سیم (WLAN)

امنیت یکی از مهم‌ترین نگرانی‌های شبکه‌های بی‌سیم می‌باشد زیرا امواج رادیویی در آن‌ها حاصل قاب‌هایی هستند که از نقطه نظر انتشار قابل دریافت به‌وسیله‌ی هر ایستگاه پایانه‌ی بی‌سیم و میزبان‌ها می‌باشند. در این بخش به معرفی مقدماتی موضوع امنیت در شبکه‌های بی‌سیم می‌پردازیم. موضوع امنیت در پروتکل 802.11 توجه زیادی را از نقطه‌نظر فنی و نیز رسانه‌ای به خود معطوف نموده است. به نظر می‌رسد توافق فراگیری از این نظر وجود دارد که مشخصات 802.11 شامل حفره‌های امنیتی جدی است که نفوذ به آن‌ها بسیار ساده و امکان‌پذیر می‌باشد. نرم‌افزارهای عمومی و آزاد را می‌توان از طریق اینترنت عمومی دانلود نمود که قادر به شناسایی حفره‌های امنیتی شبکه‌های 802.11 می‌باشند و در نتیجه این امکان را برای مهاجمان فراهم می‌کنند که به آسانی با بهره‌گیری از آن‌ها کاربران فاقد عوامل امنیتی را مورد تهاجم و سوء استفاده قرار دهند. در ادامه‌ی مطالب این بخش الگوهای امنیتی استاندارد شده در پروتکل 802.11 که مجموعه‌ی آن‌ها را **معادل سیمی حریم خصوصی (WEP)**[1] نامیده‌اند تحلیل می‌کنیم. همان‌گونه که این نام نشان می‌دهد WEP سطحی از امنیت را فراهم می‌آورد که مشابه آن را می‌توان در شبکه‌های کامپیوتری کابلی ملاحظه نمود. پس از آن تعدادی از حفره‌های امنیتی در WEP بررسی می‌شوند و آن‌گاه استاندارد 802.11i که به صورت بنیادی نسخه‌ی به مراتب امنیتی‌تر از 802.11 می‌باشد و در سال 2004 میلادی نهایی و گزینش شد تحلیل خواهد شد.

• معادل سیمی حریم خصوصی (WEP)

پروتکل IEEE802.11 توصیف شده در مستند [IEEE802.112012] اعتبارسنجی و رمزگذاری داده‌ها را بین یک میزبان و یک نقطه‌ی دستیابی بی‌سیم (WAP) با بهره‌گیری از یک رویکرد کلید مشترک متقارن تأمین می‌کند. پروتکل WEP یک الگوریتم مدیریت کلید را تعریف نمی‌کند و بنابراین فرض شده است که میزبان و نقطه‌ی دستیابی بدون سیم به نوعی بر روی استفاده از کلید از طریق روش خارج از باند توافق نموده‌اند. اعتبارسنجی لازم نیز در قالب مراحل ذیل انجام می‌گیرد:

۱- یک میزبان بی‌سیم تقاضای اعتبارسنجی را از یک نقطه‌ی دستیابی (AP) می‌کند.

1- Wired equivalent privacy

۲- نقطه‌ی دستیابی به تقاضای اعتبارسنجی با یک ارزش موقت 128 بایت پاسخ می‌دهد.
۳- میزبان بی‌سیم پیام موقت را با بهره‌گیری از کلید متقارنی که با AP به‌صورت اشتراکی استفاده می‌کند، رمزگذاری می‌کند.
۴- نقطه‌ی دستیابی رمزگشایی پیام موقت رمز میزبان را انجام می‌دهد.

در صورتی که پیام موقت رمزگشایی شده مشابه ارزش پیام موقت ارسالی از سوی میزبان باشد آن‌گاه اعتبارسنجی میزبان را موفق ارزیابی می‌کند.

الگوریتم رمز داده‌ی WEP در شکل ۸-۳۱ به تصویر درآمده است. یک کلید متقارن رمز K_S به طول 40 بیت با فرض آنکه از سوی میزبان و AP شناخته شده است، معرفی می‌شود. علاوه بر آن یک بردار اولیه‌ی (IV) به طول 24 بیت نیز به کلید K_S پیوست می‌شود تا یک کلید 64 بیت را تعریف نماید که از آن به‌منظور بازگشایی یک قاب استفاده می‌شود. بردار IV از یک قاب به قاب دیگر تغییر می‌کند و بنابراین هر قاب با یک کلید 64 بیت گوناگون رمزگذاری می‌شود. رمزگذاری به‌صورت مراحل ذیل انجام می‌شود:

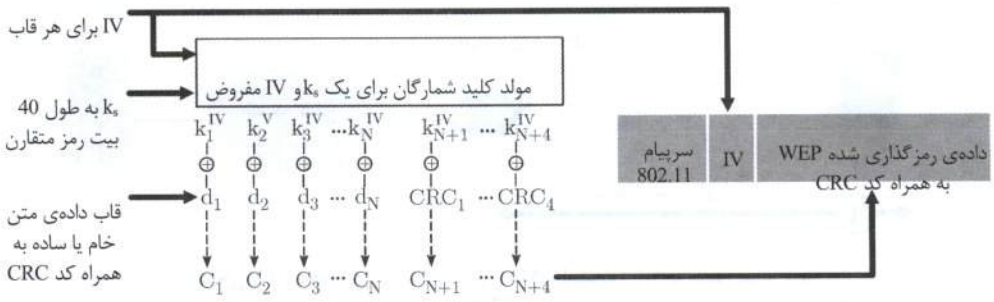
یک ارزش CRC به طول 4 بایت برمبنای فصل ۵ برای محموله‌ی داده محاسبه می‌شود. محموله و چهار بایت CRC سپس با استفاده از جریان رمز RC4 رمزگذاری می‌شوند. بررسی جزئیات رمز RC4 خارج از چارچوب این اثر می‌باشد. برای هدف ما کافی است بدانیم الگوریتم RC4 هنگامی که با یک ارزش کلید که در این حالت کلید 64 بیت به‌صورت (K_S, IV) می‌باشد، روبه‌رو می‌شود جریانی از ارزش کلیدها مانند: $K_1^{IV}, K_2^{IV}, K_3^{IV}, \ldots$ را تولید می‌کند که از آن‌ها برای رمز داده‌ها و ارزش CRC درون یک فریم استفاده می‌شود. از نقطه نظر عملی این عملیات را می‌توان در قالب انجام یک بایت در هر لحظه تصور نمود. انجام رمزگذاری از طریق XOR بایت شماره‌ی i از داده‌ی d_i با کلید K_i^{IV} در جریان ارزش کلیدها که از سوی زوج (K_S^{IV}) تولید شده است صورت می‌پذیرد. نتیجه‌ی حاصل از این عملیات تولید بایت شماره‌ی i از متن رمز C_i براساس رابطه‌ی (۸-۱۲) در ذیل می‌باشد.

$$C_i = d_i \oplus K_i^{IV} \qquad (۱۲-۸)$$

ارزش بردار IV از فریمی به قاب دیگر تغییر می‌کند و در **متن خام یا ساده‌ی پیام** در سرپیام هر WEP رمز شده‌ی قاب 802.11 مانند آنچه در شکل ۸-۳۱ نشان داده شده است جای می‌گیرد. گیرنده کلید رمز متقارن 40 بیت را که به طور اشتراکی با فرستنده مورد استفاده قرار می‌دهد دریافت می‌کند و بردار IV را به آن پیوست می‌کند. کلید 64 بیت حاصل را که مشابه کلید فرستنده برای رمزگذاری است دریافت و از آن به‌منظور رمزگشایی فریم به صورت رابطه‌ی (۸-۱۳) استفاده می‌کند:

$$d_i = C_i \oplus K_i^{IV} \qquad (۱۳-۸)$$

استفاده‌ی مناسب از الگوریتم RC4 نیازمند آن است که از ارزش کلید 64 بیت مشابهی هرگز بیش از یک مرتبه استفاده نشود. با توجه به تغییر کلید WEP از یک فریم به قاب دیگر برای یک کلید K_S مفروض که تقریباً هرگز تغییر نمی‌کند. این موضوع به مفهوم آن است که تعداد 2^{24} کلید منحصر به فرد وجود دارد. اگر این کلیدها به‌گونه‌ی تصادفی گزینش شوند می‌توان نشان داد که احتمال گزینش ارزش بردار IV مشابه و در نتیجه استفاده از کلید 64 بیت مشابه بیش از 99 درصد پس از فقط تعداد 12000 قاب می‌باشد. در یک قاب به طول یک کیلو بایت و سرعت انتقال 11Mbps تنها چند ثانیه لازم است قبل از آنکه تعداد 12000 فریم انتقال یابد. علاوه بر آن از آنجائیکه بردار IV به‌صورت متن خام در قاب ارسال می‌شود یک شنود کننده هرگاه تکرار ارزش IV روی دهد از آن آگاه می‌شود.



شکل ۸-۳۱♦ پروتکل 802.11 WEP

به‌منظور مشاهده‌ی یکی از چندین چالش روی داده هنگامی که یک کلید تکراری استفاده شده است، تهاجم متن ساده‌ی گزینش شده‌ی ذیل را از سوی T علیه Alice درنظر بگیرید.

فرض کنید T که احتمالا از یک IP جعلی استفاده می‌کند تقاضایی مانند یک HTTP یا FTP را به‌سوی Alice ارسال می‌کند تا Alice نیز یک فایل با محتوای شناخته شده‌ای مانند:

$d_1, d_2, d_3, d_4, \ldots$ را به‌سوی وی انتشار دهد. هم‌چنین T داده‌های رمز:

$C_1, C_2, C_3, C_4, \ldots$ را نیز ملاحظه می‌کند زیرا از رابطه‌ی (۸-۱۳) آگاه است. اگر حاصل XOR پیام C_i را با هر طرف از این تساوی انجام دهد آن‌گاه رابطه‌ی (۸-۱۴) در ذیل نتیجه می‌شود:

$$d_i \oplus C_i = K_i^{IV} \qquad (۸-۱۴)$$

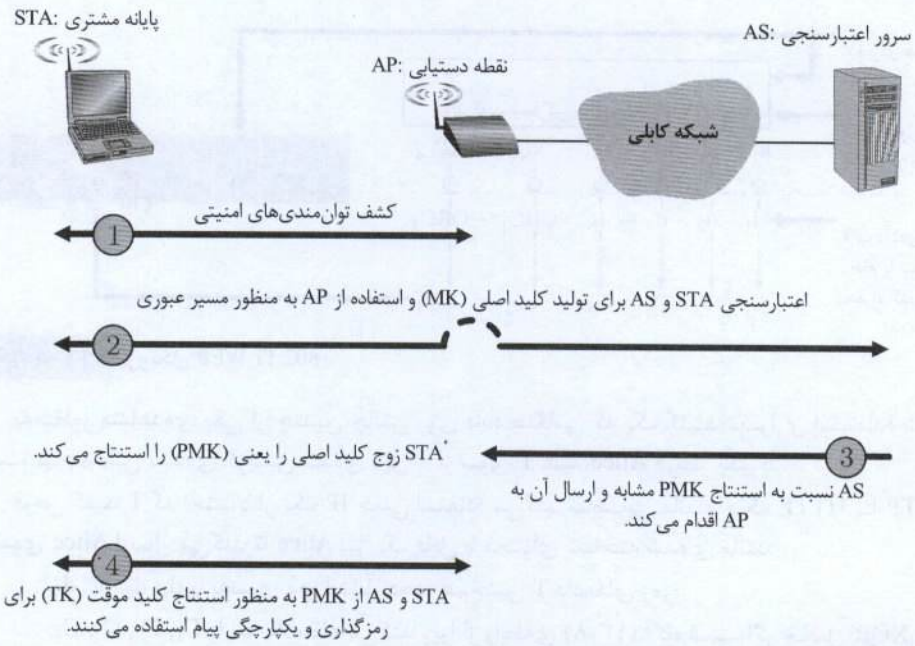
با این رابطه، T می‌تواند از ارزش‌های شناخته شده‌ی d_i و C_i برای محاسبه‌ی K_i^{IV} بهره گیرد. مرتبه‌ی بعد اگر T همان ارزش مشابه از IV را هنگام استفاده ملاحظه نماید او از ردیف کلید: $K_1^{IV}, K_2^{IV}, K_3^{IV}, \ldots$ آگاهی دارد و بنابراین قادر است پیام رمز را بازگشایی نماید.

چندین نگرانی امنیتی دیگر نیز در WEP وجود دارد که خواننده‌ی مشتاق در این زمینه به گزارش‌های مربوط به نقاط ضعف WEP در سامانه‌های اینترنت عمومی هدایت می‌شود.

۸-۱۷ پروتکل IEEE802.11i

بی‌درنگ پس از انتشار IEEE802.11 در سال 1999 میلادی تلاش در جهت توسعه و تکمیل نسخه‌ای جدید از پروتکل 802.11 سرانجام در سال 2004 میلادی نهایی و به تصویب رسید. ملاحظه خواهد شد در حالیکه WEP روش رمزگذاری نسبتاً ضعیفی را فراهم می‌نمود که فقط یک روش آن به‌منظور انجام اعتبارسنجی و بدون توزیع کلید می‌شد، پروتکل IEEE802.11i روش رمزگذاری به مراتب قوی‌تر و مجموعه‌ای پیشرفته از روش‌های اعتبارسنجی و یک توزیع کلید را تأمین می‌نماید. در ادامه چارچوب کلی پروتکل 802.11i ارایه می‌شود.

شکل ۸-۳۲ چارچوب کلی پروتکل 802.11i را نشان می‌دهد. علاوه بر برنامه‌ی مشتری بی‌سیم و AP پروتکل 802.11i یک سرور اعتبارسنجی که از طریق آن AP قادر به ارتباط می‌باشد را نیز نشان می‌دهد. با جداسازی سرور اعتبارسنجی از AP این امکان فراهم می‌شود که یک سرور اعتبارسنج در خدمت تعداد قابل توجهی از APها قرار گیرد و تصمیم‌گیری متمرکز در مورد اعتبارسنجی و دستیابی درون یک سرور امکان‌پذیر گردد. در نتیجه هزینه‌های AP و پیچیدگی کاهش یابد.



شکل ۸-۳۲♦ چهار مرحله‌ی پروتکل **IEEE802.11i**

پروتکل IEEE802.11i دارای چهار مرحله‌ی عملیاتی است:

۱- **کشف**. در مرحله‌ی کشف نقطه‌ی دستیابی AP حضور خود و نیز نوع اعتبارسنجی و رمزگذاری که قادر به تأمین آن برای گره‌ی مشتری بی‌سیم است را آگهی می‌کند. برنامه‌ی مشتری آن‌گاه تقاضای نوعی ویژه از اعتبارسنجی و رمزگذاری مطلوب خود را می‌نماید. اگرچه برنامه‌ی مشتری و AP در حال تبادل پیام می باشند اما مشتری هنوز اعتبارسنجی نشده است و کلید رمز را نیز در اختیار ندارد. بنابراین مراحل بیش‌تری لازم است تا مشتری قادر به ارتباط با یک میزبان فرضی از طریق کانال بی‌سیم باشد.

۲- **اعتبارسنجی دو جانبه و تولید کلید رمز اصلی (MK)**. اعتبارسنجی بین مشتری بی‌سیم و سرور اعتبارسنج روی می‌دهد. در این مرحله AP در قالب یک تکرار کننده عمل می‌کند و پیام‌ها را بین مشتری و سرور اعتبارسنج گسیل می‌کند. پروتکل مربوطه با نام **پروتکل اعتبارسنج پیشرفته (EAP)**[1] معروف می‌باشد که در مستند [RFC2284] به تعریف قالب‌بندی پیام انتها به انتها می‌پردازد که در یک تعامل ساده‌ی تقاضا و پاسخ بین مشتری و سرور اعتبارسنج به‌کار برده می‌شود. براساس آن‌چه شکل ۸-۳۳ نشان می‌دهد پیام‌های EAP با بهره‌گیری از پروتکل **EAPOL**[2] بسته‌بندی و از طریق پیوند بی‌سیم 802.11 ارسال می‌گردد. این پیام‌های EAP سپس در AP از بسته‌بندی خارج شده و آن‌گاه مجدداً با بهره‌گیری از پروتکل **RADIUS** (شعاع) برای انتقال از طریق پروتکل UDP/IP به سرور اعتبارسنج انتقال

1- Extensible authentication protocol
2- EAP over LAN [IEEE802.X]

می‌یابد. اگرچه سرور RADIUS و پروتکل [RFC2865] مورد نیاز پروتکل 802.11i نمی باشند اما به نوعی عناصر استاندارد آن به شمار می‌آیند. پروتکل اخیراً استاندارد شده‌ی **DIAMETER** (قطر) در مستند [RFC3588] احتمالاً در آینده‌ی نزدیکی جایگزین پروتکل RADIUS خواهد شد.

با پروتکل WEP سرور اعتبارسنج قادر به گزینش یکی از تعداد روش‌های انجام اعتبارسنجی می‌باشد. اگرچه 802.11i اجباری را در بهره‌گیری از یک روش ویژه‌ی اعتبارسنجی تحمیل نمی‌کند اما از روش اعتبارسنجی **EAP-TLS** توصیف شده در مستند [RFC 2716] در اغلب مواقع استفاده می‌شود. پروتکل EAP-TLS از روش کلید عمومی شامل پیام رمز موقت و هضم پیام، مشابه با آن‌چه در بخش ۸-۵ مطالعه شده بهره می‌گیرد تا به مشتری و سرور اعتبارسنج اجازه دهد اعتبارسنجی یکدیگر را انجام دهند و به تعریف یک کلید اصلی (MK) که شناخته شده برای هر دو می‌باشد، اقدام نمایند.

۳- **تولید زوج کلید اصلی (PMK)**. کلید MK یک رمز مشترک شناخته شده فقط برای مشتری و سرور اعتبارسنج است که هریک از آن‌ها به‌منظور تولید یک کلید دوم از آن بهره می‌گیرد و به آن زوج کلید اصلی **(PMK)**[1] گفته می‌شود. سرور اعتبارسنج سپس PMK را به AP گسیل می‌نماید. این همه همان هدفی است که به دنبال آن بودیم. هر دو مشتری AP دارای یک کلید مشترک می‌باشند و به صورت دو جانبه یکدیگر را اعتبارسنجی نموده‌اند.

۴- **تولید کلید موقت (TK)**. با کلید PMK مشتری بی‌سیم و AP قادرند کلید جدید اضافه‌ای را تولید نمایند که از آن به‌منظور ارتباط استفاده می‌شود. یکی از موارد جالب توجه **کلید موقت (TK)**[2] می‌باشد که از آن برای انجام رمزگذاری سطح پیوند از داده‌ها برروی پیوند بی‌سیم و سپس به یک میزبان فرضی استفاده می‌شود. پروتکل 802.11i چندین نوع رمزگذاری از جمله یک رمزگذاری بر مبنای AES و یک نوع رمزگذاری WEP تقویت شده را پشتیبانی می‌کند.

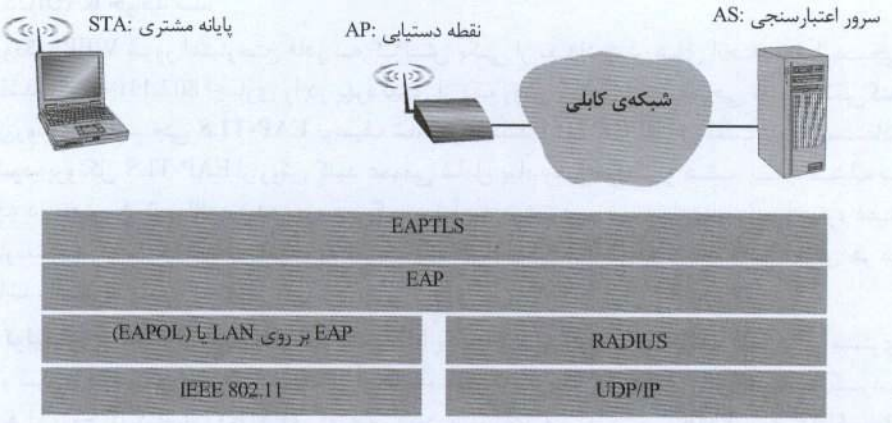
۸-۱۸ عملیات امنیتی: دیواره‌های آتش و سیستم‌های آشکارساز تهاجم (IDS)

سرتاسر این فصل ملاحظه شده است که اینترنت با توجه به تهدیدهای گسترده مکان کاملاً ناامنی از لحاظ تبادل اطلاعات می‌باشد. با توجه به این شرایط نامطلوب امنیتی در شبکه‌ی یک نهاد، وضعیت مدیر آن شبکه را در ذهن مجسم نمائید. از نقطه نظر یک مدیر شبکه دنیا به دو گروه کاملاً متمایز تقسیم شده است. انسان‌های خوب که تعلق به شبکه‌ی نهاد دارند و کاربران آن می‌باشند و بنابراین بایستی قادر به دستیابی منابع درون شبکه‌ی نهاد با رویکرد نسبتاً ساده و بدون محدودیت باشند و انسان‌های نامطلوب که شامل هر موجودیتی می‌باشد و بنابراین دستیابی به منابع شبکه برای آن‌ها بایستی به دقت تحت نظر باشد. در بسیاری از نهادها از قدیمی‌ترین تا جدیدترین آن‌ها تنها یک نقطه‌ی ورود و خروج وجود دارد که هر دو گروه خوب و بد به آن وارد و یا از آن خارج می شوند و مورد بازرسی امنیتی قرار می‌گیرند. در گذشته و در یک کاخ تاریخی این عمل از طریق یک دروازه‌ی ورودی از انتهای یک پل که باز و بسته می‌شد انجام می‌گرفته است و در یک برج شرکت تجاری مدرن و بزرگ امروزی از طریق بازرسی‌های امنیتی. در یک شبکه‌ی کامپیوتری هنگامی که ترافیک وارد و یا از شبکه

1- Pairwise master key

2- Temporal key

خارج می‌شود آن ترافیک از نقطه نظر امنیتی بازرسی، ثبت، حذف و یا گسیل می‌شود. این عملیات از طریق تجهیزاتی به نام **دیواره‌های آتش** یا **فایروال‌ها، سیستم‌های آشکارساز تهاجم (IDS)**[1] و یا **سیستم‌های ممانعت از تهاجم (IPS)** انجام می‌پذیرد.



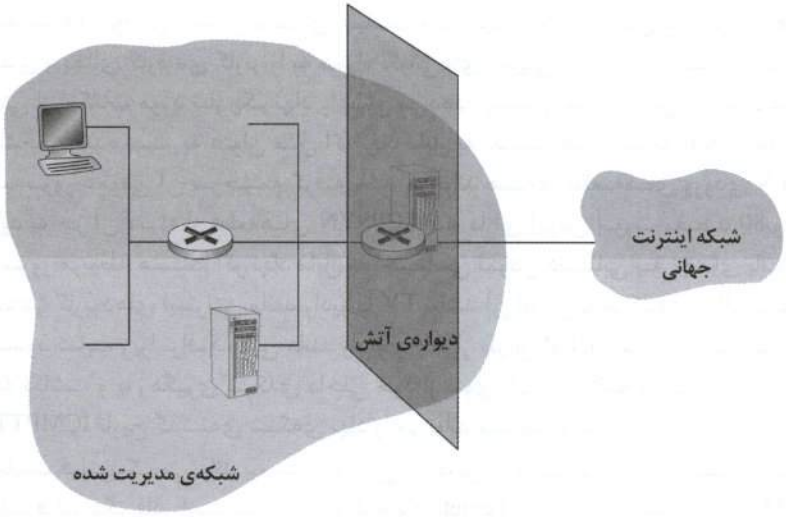
شکل ۸-۳۳ پروتکل انتها به انتهای EAP پیام‌های EAP با بهره‌گیری از EAPOL از طریق پیوند بی‌سیم بین مشتری و AP و نیز SADIUS از طریق UDP/IP بین AP و سرور اعتبارسنج بسته‌بندی می‌شوند.

۱- دیواره‌های آتش (فایروال)

یک دیواره‌ی آتش ترکیبی از سخت افزار و نرم‌افزار است که شبکه‌ی داخلی یک نهاد را از نقطه نظر کلی از شبکه‌ی اینترنت ایزوله می‌نماید و اجازه می دهد برخی از بسته‌های داده از آن عبور نمایند و برخی دیگر مسدود شوند. یک فایروال به مدیر شبکه این امکان را می‌دهد که دستیابی بین محیط خارج از شبکه و منابع تحت مدیریت او را از نقطه نظر مدیریت جریان ترافیک از سوی این منابع ویا به طرف آن‌ها کنترل و مدیریت نماید. یک دیواره‌ی آتش دارای سه هدف مشخص است:

- **تمامی ترافیک از خارج به داخل و برعکس آن از دیواره‌ی آتش عبور می‌کند.** شکل ۸-۳۴ یک دیواره‌ی آتش را نشان می‌دهد که در گوشه‌ای از مرز بین شبکه‌ی مدیریت شده و اینترنت قرار گرفته است. نهادهای بسیار بزرگ ممکن است از دیواره‌های آتش چندین سطحی و یا به عبارت دیگر توزیع شده بهره گیرند اما اسکان یک فایروال تنها در یک نقطه‌ی دستیابی شبکه همان‌گونه که شکل ۸-۳۴ نشان می‌دهد مدیریت و اعمال سیاست‌های امنیتی را ساده‌تر می‌نماید.
- **اجازه‌ی عبور برای فقط ترافیک مجاز برمبنای اعمال سیاست امنیت داخلی.** با توجه به همه‌ی ترافیک ورودی و خروجی شبکه‌ی کامپیوتری یک نهاد از درون دیواره‌ی آتش، فایروال دستیابی به ترافیک مجاز را نیز می‌تواند محدود نماید.
- **ایمن‌سازی دیواره‌ی آتش از نفوذهای مخرب.** فایروال به تنهایی یک وسیله‌ی متصل به شبکه است و در صورتی‌که به شکل مطلوب طراحی و نصب نشده باشد به آسانی قابل نفوذ می‌باشد. در این حالت فایروال فقط یک حس کاذب امنیتی را تأمین می‌نماید که عدم وجود آن به مراتب بهتر از حضورش می‌باشد.

1- intrusion detection systems



شکل ۸-۳۴♦ اسکان دیواره‌ی آتش بین شبکه‌ی مدیریت شده و دنیای خارج آن

دو شرکت Cissco و Check Point از جمله‌ی مهم‌ترین تولیدکنندگان فایروال در جهان می‌باشند. بدیهی است شما هم به آسانی می‌توانید با بهره‌گیری از جداول IP (iptable) از درون یک جعبه‌ی سیستم عامل Linux یک دیواره‌ی آتش تعریف نمائید که بسته‌ها را فیلتر نماید. نرم‌افزار iptable معمولاً بخشی از Linux می‌باشد اما از طریق اینترنت نیز می‌توان آن را دانلود نمود. دیواره‌های آتش در قالب سه گروه طبقه‌بندی می‌شوند، این سه عبارتند از:

- **فیلترهای سنتی بسته (TPF)**
- **فیلترهای حالت کامل (SF)**
- **دریچه‌های کاربرد (AG)**
- **فیلترهای سنتی بسته**

همان‌گونه که در شکل ۸-۳۴ نشان داده شده است یک نهاد معمولاً دارای یک مسیریاب دریچه است که شبکه‌ی داخلی آن نهاد را به ISP آن و در نتیجه شبکه‌ی بزرگ‌تر اینترنت متصل می‌کند. تمامی ترافیک ورودی و خروجی به شبکه‌ی داخلی از این روتر عبور می‌کند و در این مسیریاب است که کنترل و یا **فیلترینگ بسته** صورت می‌پذیرد. یک فیلتر بسته هر داده گرام را به تنهایی آزمایش می‌کند تا مطمئن شود آیا داده گرام اجازه‌ی عبور دارد و یا آنکه بایستی برمبنای قواعد ویژه‌ی تعریف شده از سوی مدیر شبکه حذف گردد. تصمیم‌گیری‌های فیلترینگ برمبنای موارد ذیل می‌باشد.

- آدرس IP مبدأ یا مقصد.
- نوع پروتکل در دامنه‌ی داده گرام IP مانند TCP، UDP، ICMP، OSPF و غیره.
- پورت مبدأ و مقصد TCP یا UDP.
- بیت‌های نشانه‌ی (پرچم) TCP مانند SYN، ACK و غیره.
- نوع پیام ICMP.
- قوانین متفاوت به‌منظور ورود و خروج داده گرام‌های شبکه.
- قوانین متفاوت برای واسطهای گوناگون مسیریاب.

یک مدیر شبکه دیواره‌ی آتش را برمبنای سیاست و نیاز نهاد پیکربندی می‌کند. این نیازها ممکن است کارآمدی و پهنای کاربردی کاربر را به همراه نگرانی‌های امنیتی در یک نهاد شامل شود. جدول ۸-۵ فهرستی از امکانات مورد نیاز یک نهاد را نشان می‌دهد. چگونگی برخورد این سیاست‌ها با یک بسته نیز مشخص شده است. به عنوان مثال اگر نهاد مایل به پذیرش هر ارتباط TCP ورودی نباشد، آن‌چه از وب سرور عمومی آن سرچشمه گرفته باشد می‌تواند همه‌ی قطعه‌های ورودی TCPSYN را مسدود نماید به جز آن دسته از قطعه‌های TCPSYN که دارای آدرس پورت مقصد 80 و آدرس IP مقصد وب سرور مربوطه هستند. اگر نهاد مایل به اختصاصی نمودن دستیابی به پهنای باند از سوی کاربران خود در کاربردهای اینترنتی مانند رادیو یا TV نباشد آن‌گاه می‌تواند کلیه‌ی ترافیک غیرحساس UDP را مسدود نماید، زیرا ترافیک‌هایی مانند رادیو و TV از طریق UDP ارسال و پخش می‌شوند. اگر نهاد مایل به نگاشت و یا رهگیری شبکه‌ی داخلی خود از سوی کاربر بیگانه‌ای نباشد آن‌گاه همگی ترافیک ICMPTTL تاریخ گذشته‌ی شبکه‌ی نهاد را می‌تواند مسدود نماید.

یک سیاست فیلترینگ می‌تواند برمبنای ترکیبی از آدرس‌ها و شماره پورت‌ها باشد. به عنوان مثال یک مسیریاب فیلترینگ قادر است همه‌ی داده گرام‌های Telnet را که دارای شماره پورت 23 می‌باشند عبور دهد به جز آن‌هایی که از یک فهرست از آدرس‌های IP ویژه وارد و یا خارج می‌شوند. این سیاست اجازه می‌دهد ارتباط‌های تِلنت به سوی و یا از طرف میزبان‌ها برمبنای یک فهرست مجاز انجام پذیرد. بدیهی است تکیه بر آدرس‌های خارجی پوشش امنیتی لازم را در مقابل داده‌گرام‌هایی که آدرس منبع آن‌ها جعل شده است، فراهم نمی‌آورد.

جدول ۸-۵ ♦ سیاست‌ها و قواعد فیلترینگ برای شبکه‌ی یک نهاد با آدرس 130.27/16 و وب سرور با آدرس: IP : 130.207.244.203

سیاست	تعریف فایروال
ممنوعیت دستیابی وب از خارج	حذف همه‌ی بسته‌های خروجی به هر آدرس IP و پورت 80
ممنوعیت ارتباط‌های TCP ورودی به جز آن‌هایی که به‌سوی وب سرور عمومی نهاد می‌باشند.	حذف همه‌ی بسته‌های ورودی TCPSYN به هر IP به جز آدرس IP: 130.207.244.203 و پورت 80
ممانعت دستیابی رادیو و TV به‌منظور جلوگیری از اتلاف پهنای باند در دسترس شبکه	حذف همه‌ی بسته‌های UDP ورودی به جز بسته‌های DNS.
ممانعت شبکه در برابر کاربرد یک تهاجم DoS	حذف همه‌ی بسته‌های پینگ ICMP به یک آدرس فراپخش مانند: 130.207.255.255
ممانعت از رهگیری شبکه	حذف همه‌ی بسته‌های اعتبار گذشته ICMPTTL خروجی.

فیلترینگ هم‌چنین می‌تواند برمبنای ارزش 1 و یا 0 بیت TCPACK باشد. از این امکان سودمند یک نهاد هنگامی بهره می‌گیرد که مایل است میزبان‌های داخل آن به سرورهای بیرونی متصل شوند اما مایل است از اتصال میزبان‌های خارجی به سرورهای درونی خود ممانعت به عمل آورد. با یادآوری مطالب بخش ۳-۱۱ از فصل سوم که گفته شد نخستین قطعه در هر ارتباط TCP دارای یک بیت ACK مقداردهی شده به 0 می باشد در حالی‌که بقیه‌ی قطعه‌ها در آن ارتباط دارای بیت ACK با

ارزش ACK=1 می‌باشند. بنابراین اگر یک نهاد مایل به جلوگیری میزبان‌های خارجی از ارتباط به سرورهای داخلی باشد به آسانی همه‌ی قطعه‌های ورودی را از طریق مقداردهی ACK با ارزش 0 فیلتر می‌کند. این سیاست همه‌ی ارتباط‌های TCP سرچشمه گرفته از خارج را فیلتر می‌کند اما ارتباط‌های با مبدأ داخلی را امکان‌پذیر می‌سازد.

قواعد دیواره‌های آتش در روترها با بهره‌گیری از فهرست کنترل دستیابی پیاده‌سازی می‌شود که در آن هر واسط مسیریاب دارای فهرست خود می‌باشد. مثالی از یک فهرست کنترل دستیابی برای یک نهاد با آدرس: 222.22/16 در جدول ۸-۶ نشان داده شده است. این فهرست کنترل دستیابی برای واسطی است که مسیریاب را به ISP خارجی نهاد متصل می‌کند. قواعد اعمال شده به هر داده گرام عبوری از واسط به گونه‌ی از فراز به فرود می‌باشد. دو قانون نخست به همراه هم به کاربران داخلی اجازه می‌دهند کاوشگری در وب را انجام دهند: قانون اول به هر بسته‌ی TCP با آدرس پورت مقصد 80 اجازه می‌دهد که از شبکه‌ی نهاد خارج شود و قانون دوم اجازه می‌دهد هر بسته‌ی TCP با پورت مبدأ 80 و بیت ACK=1 به شبکه‌ی نهاد وارد شود. بدیهی است اگر یک منبع بیگانه تلاش نماید یک ارتباط TCP را با یک میزبان داخلی برقرار نماید آن ارتباط مسدود می‌شود حتی اگر پورت مبدأ و یا مقصد 80 باشد. دو قانون دوم جدول نیز به همراه یکدیگر اجازه می‌دهند بسته‌های DNS به شبکه‌ی یک نهاد وارد و یا از آن خارج گردند. به طور خلاصه روش فهرست کنترل دستیابی محدود کننده‌ی تمامی ترافیک است اما ترافیک وب آغاز شده از درون نهاد و یا DNS را مسدود می‌کند. مستند [CERT FILTERING 2012] فهرستی از پورت‌ها و پروتکل‌های فیلترینگ بسته را به‌منظور شناسایی و ممانعت از تعدادی از حفره‌های امنیتی شناخته شده در کاربردهای موجود شبکه معرفی می‌کند.

جدول ۸-۶: یک فهرست کنترل دستیابی برای یک واسط مسیریاب

عملیات	آدرس مبدأ	آدرس مقصد	پروتکل	پورت مبدأ	پورت مقصد	بیت پرچم
مجاز	222.22/16	خارج از 222.22/16	TCP	>1023	80	هر ارزش
مجاز	خارج از 222.22/16	222.22/16	TCP	80	>1023	ACK
مجاز	222.22/16	خارج از 222.22/16	UDP	>1023	53	-
مجاز	خارج از 222.22/16	222.22/16	UDP	53	>1023	-
غیرمجاز	همه	همه	همه	همه	همه	همه

- **فیلترینگ حالت کامل (SF)**

در روش فیلترینگ سنتی بسته، تصمیم‌گیری‌های فیلترینگ به‌گونه‌ی مستقل برای هر بسته انجام می‌شود. فیلترهای حالت کامل (SF) ارتباط‌های TCP را رهگیری می‌کنند و از این آگاهی به‌منظور اتخاذ تدابیر تصمیم‌گیری بهره می‌گیرند. به‌منظور درک بهتر فیلترهای SF مجدداً فهرست کنترل دستیابی جدول ۸-۶ تحلیل می‌شود. علیرغم محدودگرایی فهرست کنترل دستیابی جدول ۸-۶ در هر

حال این اجازه به هر بسته‌ی ورودی از خارج با ACK=1 و شماره پورت 80 داده می‌شود که از فیلتر عبور نماید. چنین بسته‌هایی می‌توانند مورد بهره برداری مهاجمان در تلاش به‌منظور فروپاشی سیستم‌های داخلی با بسته‌های معیوب و جعلی قرار گیرند که نتیجه‌ی آن انجام تهاجم‌های DoS و یا پیکربندی شبکه‌ی داخلی می‌باشد. راهکار غیرمنطقی برای این گونه چالش‌ها مسدود نمودن بسته‌های TCPACK می‌باشد اما چنین رویکردی سبب جلوگیری کاربران درونی نهاد از گردشگری در وب خواهد شد.

بهره‌گیری از فیلترهای SF این چالش‌ها را از طریق رهگیری همه‌ی ارتباط‌های TCP در جریان در یک جدول ارتباط برطرف می‌سازد. این رویکرد امکان‌پذیر است زیرا دیواره‌های آتش قادر به مشاهده‌ی یک ارتباط جدید از طریق یک دست‌دهی سه جانبه به صورت SYN، SYNACK و ACK می‌باشند و از خاتمه‌ی یک ارتباط نیز با مشاهده‌ی بسته‌ی FIN آگاه می‌شوند. یک فایروال هم‌چنین می‌تواند در صورت عدم مشاهده‌ی فعالیتی بر روی یک ارتباط TCP به عنوان مثال برای مدت 60 ثانیه آن ارتباط را خاتمه یافته تلقی نماید. نمونه‌ای از یک جدول ارتباط برای یک دیواره‌ی آتش در جدول ۸-۷ نشان داده شده است. این جدول ارتباط، مشخص می‌کند که در حال حاضر سه ارتباط TCP در جریان است که همگی آن‌ها از درون یک نهاد برآمده‌اند. علاوه بر این موضوع SF شامل یک ستون جدید به نام **ارزیابی ارتباط** در فهرست کنترل دستیابی خود می‌باشد که در جدول ۸-۸ نشان داده شده است. بدیهی است جدول ۸-۸ مشابه فهرست کنترل دستیابی جدول ۸-۶ می‌باشد به جز آن که ارتباط بایستی برای حداقل دوقانون مورد ارزیابی قرار گیرد.

به‌منظور درک بهتر چگونگی عملکرد جدول ارتباط و فهرست توسعه یافته‌ی کنترل دستیابی به تحلیل یک مثال می‌پردازیم. فرض کنید یک مهاجم تلاش می‌کند یک بسته‌ی مجعول و معیوب را از طریق ارسال یک داده گرام TCP با پورت مبدأ 80 و ACK=1 به درون شبکه‌ی نهاد گسیل دارد. علاوه بر آن فرض کنید این بسته حامل شماره پورت مبدأ 12543 و آدرس IP مبدأ: 150.23.23.155 می‌باشد. هنگامی که این بسته وارد فایروال می‌شود مورد ارزیابی فهرست کنترل دستیابی دیواره‌ی آتش در جدول ۸-۸ قرار می‌گیرد. این جدول مشخص می‌کند که جدول ارتباط نیز بایستی قبل از اجازه‌ی ورود این بسته به درون شبکه‌ی نهاد مورد ارزیابی قرار گیرد. فایروال جدول ارتباط را ارزیابی می‌کند و ملاحظه می‌کند این بسته بخشی از یک ارتباط TCP در جریان نمی‌باشد و بنابراین بسته را مسدود می‌کند. به عنوان مثال دوم فرض کنید یک کاربر داخلی مایل به جستجوی یک سامانه‌ی وب خارج از شبکه نهاد است. از آن‌جا که کاربر نخست یک قطعه‌ی TCPSYN ارسال می‌کند، ارتباط TCP این کاربر در جدول ارتباط ثبت می‌شود. هنگامی که وب سرور بسته‌ها را بازگشت می‌دهد و ACK=1 می‌شود، فایروال جدول را ارزیابی و مشاهده می‌کند که ارتباط وابسته به آن در جریان است. بنابراین دیواره‌ی آتش در حالی اجازه‌ی عبور بسته‌ها را می‌دهد که مداخله‌ای را در فعالیت وب‌گردی کاربر داخلی اعمال نمی‌کند.

جدول ۸-۷♦ جدول ارتباط فیلتر SF

آدرس مبدأ	آدرس مقصد	پورت مبدأ	پورت مقصد
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

جدول ۸-۸ ♦ فهرست کنترل دستیابی یک فیلتر SF

عملیات	آدرس مبدأ	آدرس مقصد	پروتکل	پورت مبدأ	پورت مقصد	بیت نشانه	ارزیابی ارتباط
مجاز	222.22/16	خارج از 222.22/16	TCP	>1023	80	هر ارزش	
مجاز	خارج از 222.22/16	222.22/16	TCP	80	>1023	ACK	X
مجاز	222.22/16	خارج از 222.22/16	UDP	>1023	53	-	
مجاز	خارج از 222.22/16	222.22/16	UDP	53	>1023	-	X
غیرمجاز	همه	همه	همه	همه	همه	همه	

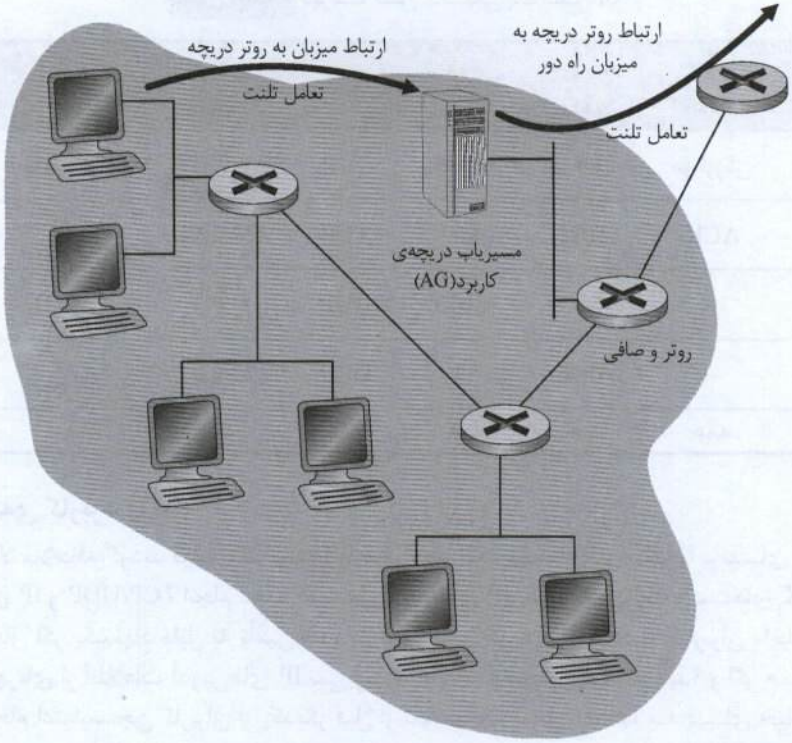
- **دریچه‌ی کاربرد (AG)**

در مثال بالا ملاحظه گردید فیلترینگ بسته اجازه می‌دهد یک نهاد، فیلترینگ را برمبنای محتوای سرپیام‌های IP و TCP/UDP انجام دهد، که شامل ادرس‌های IP، شمارگان پورت و بیت‌های ACK نیز می‌شود. حال اگر یک نهاد مایل به تأمین خدمات تلنت برای گروهی محدود از کاربران داخلی خود باشد که بهره‌ای از اطلاعات آدرس‌های IP نمی‌برند آن‌گاه چه وضعیتی پدید می‌آید؟ و اگر همان نهاد مایل به انجام اعتبارسنجی کاربران از یکدیگر قبل از تعریف یک تعامل Telnet به دنیای خارج باشد آن‌گاه وضعیت چگونه است؟ این عملکردها خارج از توان فیلترهای سنتی و SF می‌باشند. بدیهی است اطلاعات در مورد شناسه‌ی کاربران داخلی، داده‌ی از نوع لایه‌ی کاربرد است و در سرپیام‌های IP، TCP و یا UDP قرار ندارد.

به‌منظور تأمین سطح دقیق‌تری از امنیت، دیواره‌های آتش بایستی فیلترهای بسته را با مسیریاب‌های دریچه کاربرد ترکیب نمایند. این‌گونه مسیریاب‌ها فراتر از سرپیام‌های IP/TCP/UDP عمل می‌کنند و تصمیم‌گیری امنیتی را بر مبنای داده‌های کاربردی انجام می‌دهند. یک **دریچه‌ی کاربردی (AG)**[1] یک سرور ویژه‌ی کاربرد است که همه‌ی داده‌های کاربردی چه درون محدوده‌ی باند و یا خارج آن بایستی از آن عبور نمایند. دریچه‌های کاربرد چندگانه می‌توانند برروی یک میزبان پیاده‌سازی شوند اما هر دریچه به‌گونه‌ی یک سرور مجزا با پردازش‌های اختصاصی خود می‌باشد.

به‌منظور درک بهتر دریچه‌های کاربرد به طراحی یک فایروال می‌پردازیم که اجازه می‌دهد فقط مجموعه‌ای از کاربران داخلی باتلنت بیرونی ارتباط داشته باشند و در مقابل مانع همه‌ی کاربران خارج از شبکه‌ی خود به انجام امور Telnet داخلی می‌شود. چنین سیاست امنیتی از طریق ترکیبی از یک فیلتر بسته در یک روتر و یک دریچه‌ی کاربرد تلنت پیاده‌سازی می‌شود. شکل ۸-۳۵ این مهم را نشان داده است.

1- Application gateway



شکل ۸-۳۵♦ دیواره‌ی آتش شامل یک دریچه‌ی کاربرد و یک فیلتر

فیلتر مسیریاب به‌گونه‌ای پیکربندی شده است که همه‌ی ارتباط‌های تلنت به جز آن گروهی که از آدرس IP دریچه‌ی کاربرد (AG) سرچشمه گرفته‌اند، را مسدود می‌کند. پیکربندی این‌گونه از یک فیلتر همه‌ی ارتباط‌های Telnet خارج از باند را مجبور به عبور از AG می‌کند. اکنون یک کاربرد درونی که مایل است از امکان تلنت دنیای خارج از خود بهره گیرد را در ذهن تصور نمائید. کاربر نخست بایستی یک تعامل تلنت با AG برقرار نماید. یک کاربرد در حال اجرا در AG که مشغول شنود تعامل‌های تلنت ورودی می‌باشد کاربر را مجبور می‌سازد که یک شناسه (ID) و رمز عبور ارایه دهد. هنگامی که کاربر این اطلاعات را تأمین نمود مسیریاب AG این اطلاعات را ارزیابی می‌کند تا مطمئن شود کاربر مجاز به انجام Telnet به دنیای خارج از خود می باشد. در غیر این صورت ارتباط تلنت از کاربر داخلی به AG از سوی مسیریاب قطع می‌شود. در صورتی که کاربر مجوز لازم را در اختیار دارد مسیریاب AG اقدام به عملیات ذیل می‌نماید:

۱- از کاربر می‌خواهد که نام میزبان خارجی که کاربر مایل به ارتباط با آن می‌باشد را تأمین نماید.

۲- یک تعامل Telnet بین AG و میزبان خارجی برقرار می‌سازد.

۳- همه‌ی داده‌های ورودی از کاربر را به میزبان خارجی گسیل می‌کند ونیز همه‌ی داده‌های ورودی از میزبان خارجی را در اختیار کاربر مورد نظر قرار می‌دهد.

بنابراین AG یک تلنت نه تنها به انجام اعتبارسنجی کاربر بلکه به عنوان یک سرور و یک مشتری Telnet اطلاعات را بین کاربر و سرور تلنت خارجی مبادله می‌کند. بدیهی است فیلتر اجازه‌ی انجام

مرحله‌ی (۲) را میسر می‌سازد زیرا AG آغازگر ارتباط تلنت به دنیای خارج است. شبکه‌های داخلی در اغلب موارد دارای چندین AG می باشند. به عنوان مثال مسیریاب‌های دریچه برای Telnet، HTTP، FTP و e-mail. در حقیقت سرور ایمیل یک نهاد و وب نهان آن یک AG می‌باشند.

بدیهی است مسیریاب‌های AG خالی از عیب هم نیستند. نخست برای هر کاربردی AG ویژه‌ی آن کاربرد مورد نیاز است. دوم جریمه‌ی عملکرد و کارایی ضعف دوم است زیرا همه‌ی داده‌ها از طریق AG پخش مجدد می‌شوند. این موضوع نگران کننده است به ویژه در آن هنگام که چندین کاربر یا کاربرد از AG مشابه استفاده می‌کنند. سرانجام نرم‌افزار مشتری بایستی آگاه از چگونگی ارتباط با AG باشد. زیرا هنگامی که کاربرد تقاضایی را مطرح می‌کند بایستی از چگونگی اعلام آن به AG و سرور خارجی که به آن متصل می‌شود، آگاه باشد.

۸-۱۹ پیشینه‌ی تاریخی محرمانگی و حریم خصوصی

فرض کنید شما مایل به بازدید از یک سامانه‌ی وب غیرمتعارف می‌باشید و مایل به افشای اطلاعات ذیل نمی‌باشید:

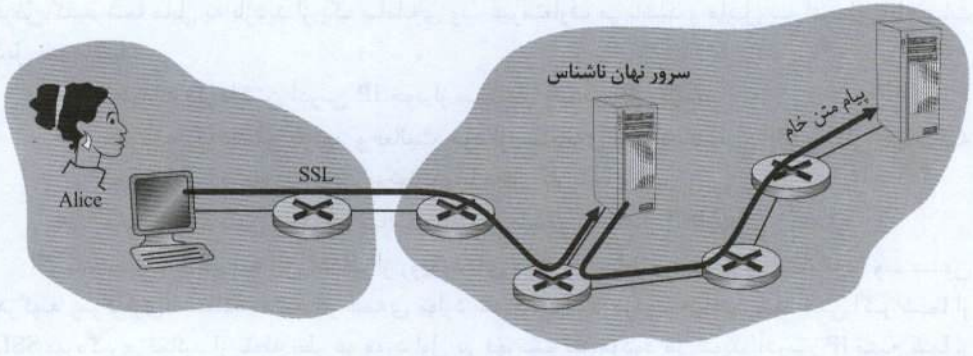
- محرمانه نگهداشتن آدرس IP خود از سامانه‌ی وب در حال بازدید.
- محرمانه نگهداشتن بازدید و فعالیت خود از سامانه‌ی اشاره شده از ISP محلی خود که ممکن است متعلق به محل سکونت و یا محل کار شما باشد.
- مخفی نگهداشتن داده‌های مبادله شده بین شما و سامانه از ISP محلی.

اگر به‌منظور دستیابی به این اهداف از رویکردهای سنتی ارتباط مستقیم با سامانه‌ی وب بدون هرگونه رمزگذاری استفاده نمائید در همه‌ی موارد سه گانه‌ی بالا مردود خواهید شد. حتی اگر شما از SSL بهره‌گیری نمائید از نقطه نظر دو مورد اول در فهرست نیز مردود می‌شوید. آدرس IP منبع شما از طریق هر داده گرامی که ارسال می‌نمائید در اختیار سامانه‌ی وب گذارده می‌شود و آدرس IP مقصد هریک از داده گرام‌های ارسالی از جانب شما نیز از سوی ISP محلی شما قابل شنود می‌باشد.

به‌منظور دستیابی محرمانگی و حریم خصوصی، شما می‌توانید از ترکیبی از سرورهای نهان مورد اعتماد و SSL برمبنای آن‌چه در شکل ۸-۳۶ نشان داده شده است استفاده نمائید. با این رویکرد نخست یک ارتباط SSL به سوی سرور نهان مورد اعتماد برقرار می‌سازید و آن‌گاه یک تقاضای HTTP را برای یک صفحه در سامانه‌ی مورد نظر به این ارتباط SSL تزریق می‌نمائید. هنگامی که سرور نهان (proxy) تقاضای HTTP رمز شده‌ی SSL را دریافت نماید آن را رمزگشایی می‌کند و متن خام تقاضای HTTP را به سامانه‌ی وب گسیل می دارد. سامانه‌ی وب سپس به سرور نهان پاسخ می‌دهد و آن نیز نتیجه را از طریق SSL به سوی شما ارسال می‌دارد. از آن‌جا که سامانه‌ی وب فقط آدرس IP سرور نهان را ملاحظه می‌کند و از آدرس مشتری آگاهی ندارد بنابراین دستیابی شما به سامانه‌ی وب حالت محرمانه و ناشناس را خواهد داشت. از طرفی به دلیل آنکه تمامی ترافیک بین شما و سرور نهان رمز شده است ISP محلی قادر به دست‌اندازی به حریم خصوصی شما از طریق ثبت سامانه‌هایی که ملاقات نموده‌اید نمی‌باشد و نیز داده‌های مبادله شده را نیز نمی‌تواند ثبت نماید. بسیاری از شرکت‌ها مانند proxify.com چنین خدمات نهان‌سازی را در اختیار کاربران قرار می‌دهند.

بدیهی است در این راهکار سرور نهان شما از همه چیز آگاه است. از جمله آدرس IP شما و نیز آدرس IP سامانه‌ای که مورد بازدید قرار داده‌اید و همه‌ی ترافیک مبادله شده بین شما و سامانه‌ی وب

را به صورت متن خام یا ساده یا به عبارت دیگر شفاف، مشاهده می‌کند. چنین راهکاری تا آن‌جایی که سرور نهان مورد اطمینان می‌باشد، مطلوب به نظر می‌رسد. یک رویکرد به مراتب قوی‌تر که از سوی پروتکل TOR برای خدمات محرمانگی و حریم خصوصی استفاده می‌شود مسیردهی ترافیک از درون مجموعه‌ای از سرورهای نهان غیروابسته بنابر مستند [TOR 2012] می باشد. به‌ویژه آنکه TOR اجازه می‌دهد به صورت انفرادی و مستقل از یکدیگر عوامل نهان‌سازی به مجموعه‌ی سرور نهان افزوده شود. هنگامی که یک کاربر به یک سرور متصل می‌شود و از TOR نیز بهره می‌گیرد، نرم‌افزار TOR به گونه‌ی تصادفی زنجیره‌ای از سه پروکسی را از انبار خود گزینش و مسیردهی تمامی ترافیک بین مشتری و سرور را از طریق این زنجیره انجام می‌دهد. در این حالت با فرض آنکه سرورهای نهان برای یکدیگر آشکار نمی‌باشند هیچ کس از انجام ارتباط بین آدرس IP شما و سامانه‌ی وب هدف آگاه نخواهد بود. علاوه بر آن اگرچه متن شفاف پیام بین آخرین سرور نهان و سرور وب ارسال می‌شود اما آخرین proxy از آدرس IP فرستنده و گیرنده‌ی متن خام آگاهی ندارد.



شکل ۸-۳۶♦ تأمین محرمانگی و حریم خصوصی از طریق یک proxy

۸-۲۰ سیستم‌های آشکارساز تهاجم (IDS)

در بخش‌های گذشته ملاحظه شد یک فیلتر بسته (سنتی و SF) بازرسی دامنه‌های سرپیام IP، TCP، UDP و ICMP را هنگام تصمیم‌گیری در مورد اینکه کدام بسته را بایستی از درون دیواره‌های آتش عبور دهد، انجام می‌دهد. برای آشکارسازی انواع بسیاری از تهاجم‌ها نیازمند انجام **بازرسی عمیق بسته‌ها** می‌باشیم که فراتر از دامنه‌های سرپیام می‌باشد و به درون داده‌های کاربردهایی نفوذ می‌کند که بسته‌ها حامل آن می‌باشند. در بخش ۸-۱۸ ملاحظه شد AGها اغلب به بازرسی عمیق بسته‌ها می پردازند. اما یک AG فقط این عمل را برای یک کاربرد ویژه انجام می‌دهد.

آشکار است وسیله‌ی راهبردی دیگری نیز وجود دارد که نه تنها سرپیام همه‌ی بسته‌های عبوری از درون خود را مانند یک فیلتر بسته بازرسی می‌کند بلکه بازرسی عمیق بسته‌ها را برخلاف یک فیلتر بسته نیز انجام می‌دهد. هنگامی که چنین وسیله‌ای به یک بسته‌ی مشکوک و یا تعدادی از آن‌ها برخورد می‌کند، می‌تواند مانع ورود آن بسته‌ها به شبکه‌ی سازمان شود. در صورتی که فعالیت بسته مشکوک به نظر آید آن وسیله می‌تواند اجازه‌ی عبور بسته را صادر نماید اما هشدارهایی را به یک مدیر شبکه ارسال نماید. مدیر شبکه پس از آن می‌تواند نگاه عمیق‌تری به همه‌ی ترافیک عبوری انداخته و تصمیم مناسب را دراین زمینه اتخاذ نماید. یک وسیله که هشدارها را در اثر روبه رو شدن با یک

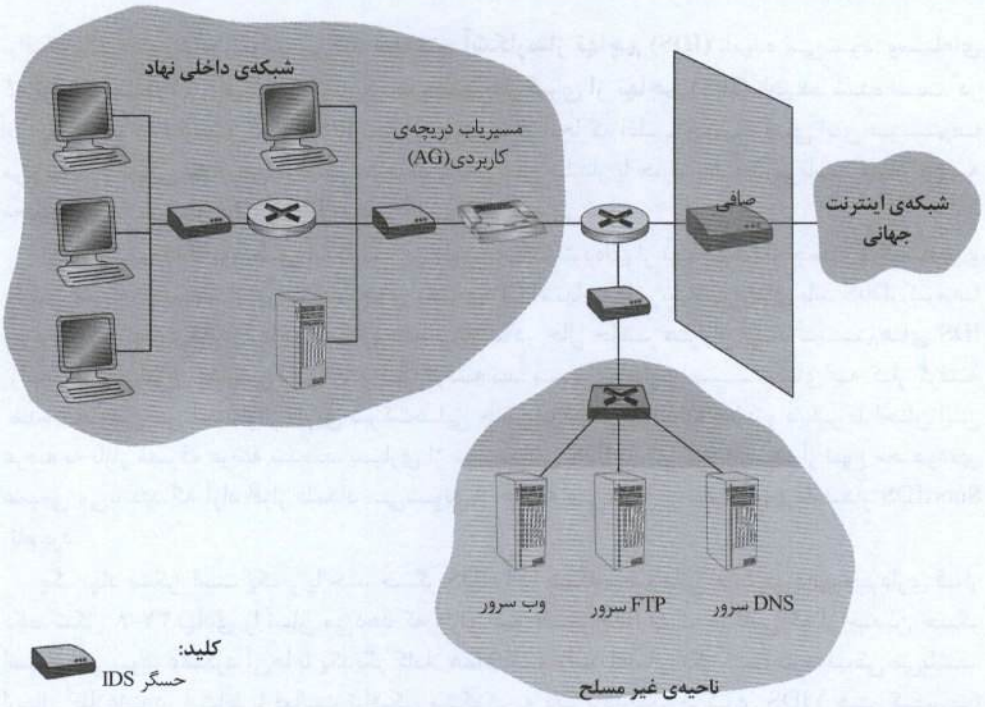
ترافیک بالقوه تهدیدآمیز تولید می‌کند **سیستم آشکارساز تهاجم (IDS)** نامیده می‌شود. وسیله‌ای که ترافیک مشکوک را فیلتر می‌کند یک **سیستم جلوگیری از تهاجم (IPS)** نامیده شده است. در این بخش هر دو سیستم IDS و IPS تحلیل می‌شوند. از آنجا که اغلب جوانب فنی این سیستم‌ها مربوط به چگونگی تشخیص ترافیک مشکوک و نه ارسال هشدار یا حذف بسته‌ها می‌باشد بنابراین به مجموعه‌ی دو سیستم IDS و IPS به طور کلی IDS گفته می‌شود.

یک سیستم IDS به‌منظور آشکارسازی دامنه‌ی گسترده‌ای از تهاجم‌ها از جمله پیکربندی و نگاشت شبکه، جستجوی پورت، جستجوی پشته‌ی TCP، تهاجم‌های سیلاب پهنای باند DoS، کرم‌ها و ویروس‌ها، تهدید OS و تهاجم به کاربردها می‌باشد. در حال حاضر هزاران نهاد سیستم‌های IDS را به‌منظور مسایل امنیتی خود به کار گرفته‌اند. بسیاری از این سیستم‌های به کار گرفته شده خصوصی می‌باشند و از طریق شرکت‌هایی چون Cissco و Checkpoint و دیگر طراحان این عرصه به بازار مصرف عرضه شده‌اند. بسیاری از سیستم‌های IDS به کار گرفته شده از نوع محدوده‌ی عمومی می‌باشند که آزاد افزار قلمداد می‌شوند. از جمله می‌توان از سیستم پرطرفدار SnortIDS نام برد.

یک نهاد ممکن است یک و یا چند حسگر IDS را در شبکه‌ی سازمانی خود مورد بهره‌برداری قرار دهد. شکل ۸-۳۷ نهادی را نشان می‌دهد که دارای سه حسگر IDS است. هنگامی که از چندین حسگر استفاده می‌شود، عملکرد آن‌ها با یکدیگر کاملاً هماهنگ و مانند اعضای یک کنسرت موسیقی می‌باشد. ارسال اطلاعات در ارتباط با فعالیت ترافیکی مشکوک به یک پردازشگر مرکزی IDS (رهبر کنسرت) انجام می‌گیرد. این پردازشگر اطلاعات را جمع‌آوری و یکپارچه می کند و سپس هشدارهای لازم را به مدیران شبکه در صورت لزوم ارسال می‌کند. در شکل ۸-۳۷ این نهاد ویژه شبکه‌ی خود را به دو ناحیه تقسیم نموده است. (۱) ناحیه‌ی با امنیت بالا که از طریق یک فیلتر بسته و یک مسیریاب دریچه‌ی کاربرد (AG) حفاظت می‌شود و نظارت بر آن نیز از سوی حسگرهای IDS انجام می‌شود و (۲) ناحیه‌ی با امنیت پائین که اصطلاحاً به آن **ناحیه‌ی غیرنظامی (DMZ)**[1] گفته می‌شود. این ناحیه فقط از طریق فیلتر بسته حفاظت می‌شود اما نظارت بر آن از سوی حسگرهای IDS می باشد. بدیهی است DMZ شامل سرورهای نهاد نیز می‌باشد زیرا به‌منظور برقراری ارتباط با دنیای خارج به آن‌ها نیاز می‌باشد. این سرورها عبارتند از وب سرور عمومی و سرور DNS خودمختار آن.

ممکن است تعجب کنید چرا از چندین حسگر IDS استفاده می‌شود؟ چرا یک حسگر IDS فقط در پشت فیلتر بسته و یا حتی در درون آن مانند شکل ۸-۳۷ قرار نمی‌گیرد؟ در ادامه ملاحظه می‌شود که یک IDS نه تنها نیازمند بازرسی عمیق بسته است بلکه بایستی به مقایسه‌ی هر بسته‌ی عبوری با ده‌ها هزار **امضای دیجیتالی** بپردازد. این حجم اطلاعات نیازمند مقدار قابل توجهی پردازش می‌باشد به ویژه اگر نهاد به دریافت گیگابیت بر ثانیه اطلاعات از اینترنت نیاز داشته باشد. از طریق قرار دادن حسگرهای IDS در انتهای جریان دریافت اطلاعات هر حسگر فقط بخشی کوچک از ترافیک اطلاعات نهاد را مشاهده می‌کند و بنابراین به راحتی می‌تواند وقایع شبکه را دنبال نماید. حسگرهای IDS با کارایی بالا و سیستم های IPS در حال حاضر به آسانی در دسترس شبکه‌ها می‌باشند و بسیاری از نهادها می‌توانند نیازهای خود را فقط از طریق یک حسگر که در نزدیکی مسیریاب دستیابی قرار می‌گیرد، تأمین نمایند.

1- Demilitarized zone



شکل ۸-۳۷♦ شبکه‌ی یک نهاد با بهره‌گیری از یک فیلتر بسته، یک AG و حسگرهای IDS

سیستم‌های IDS به شکل گسترده‌ی خود در قالب دو گروه طبقه‌بندی می‌شوند: **سیستم‌های برمبنای امضا (SBS)**[1] **و سیستم‌های غیرمتعارف (ABS)**[2]. یک IDS برمبنای SBS پایگاه داده‌ی وسیعی از امضاهای تهاجم را نگهداری می‌کند. هر امضا مجموعه‌ای از قوانین لازم و وابسته به یک فعالیت تهدیدآمیز شبکه می‌باشد. یک امضا ممکن است فهرستی از مشخصات مربوط به یک بسته مانند شمارگان پورت مبدأ و مقصد، نوع پروتکل و یک رشته از بیت‌های ویژه در محموله‌ی بسته و یا آنکه مربوط به یک گروه از بسته‌ها باشد. امضاها معمولاً از سوی افراد متخصص و ماهر در زمینه‌ی امنیت شبکه در پی تهاجم‌های شناخته شده طراحی می‌شوند. مدیر یک شبکه امضاها را می‌تواند بنابر نیاز یک نهاد پیکربندی نماید و یا امضاهای ویژه‌ی خود را به پایگاه داده‌ی امضاها اضافه نماید.

از نقطه نظر عملیاتی یک سیستم IDS برمبنای امضا (SBS)، هر بسته‌ی عبوری را شنود می‌کند و سپس آن را با امضاهای خود در پایگاه داده مقایسه می‌کند. در صورتی که بسته یا مجموعه‌ای از بسته‌ها با یک امضا در پایگاه داده سازگار باشد آن‌گاه IDS یک هشدار تولید می‌کند. هشدار تولید شده می‌تواند به مدیر شبکه از طریق یک پیام ایمیل گسیل شود و یا با بهره‌گیری از سیستم مدیریت شبکه به‌منظور بازرسی در آینده ثبت گردد.

سیستم‌های IDS برمبنای امضا اگرچه به صورت گسترده مورد استفاده قرار گرفته‌اند اما

1- Signature based system

2- Anomaly based system

محدودیت‌هایی را نیز شامل می‌شوند. از همه مهم‌تر آن‌ها نیازمند آگاهی‌های قبلی از تهاجم می‌باشند تا امکان تولید یک امضای دقیق را داشته باشند. به عبارت دیگر یک IDS برمبنای امضا، کاملاً از تهاجم‌های جدید بی‌اطلاع می‌باشد و بنابراین آن‌ها بایستی در حافظه‌ی آن ثبت گردند. نقطه ضعف دیگر آن است که حتی اگر امضا قابل نگاشت باشد ممکن است نتیجه‌ی یک تهاجم نبوده و بنابراین سبب تولید یک هشدار غلط شده است. سرانجام از آن‌جا که هر بسته بایستی با مجموعه‌ی وسیعی از امضاها مقایسه شود سیستم IDS ممکن است از نقطه نظر پردازش اشباع گردیده و در نتیجه قادر به تشخیص بسیاری از بسته‌های تهدیدآمیز نباشد.

یک سیستم IDS غیرمتعارف یا ABS به تولید یک سابقه‌ی ترافیکی در طول عملیات نرمال خود می‌پردازد. سپس به جستجوی جریان بسته‌ای می‌باشد که از لحاظ آماری غیرمتعارف است. به عنوان مثال وقوع درصد غیرمعمول از بسته‌های ICMP و یا یک رشد تصاعدی ناگهانی در جستجوی پورت‌ها و پینگ‌ها. مهم‌ترین ویژگی ABS آن است که وابسته به دانش قبلی در زمینه‌ی تهاجم‌های موجود نمی‌باشد، به عبارت دیگر آن‌ها قادرند تهاجم‌های بالقوه‌ی جدید وثبت نشده را تشخیص دهند. بدیهی است در این حالت تمایز بین ترافیک معمولی و ترافیک غیرمعمول آماری بسیار دشوار می‌باشد. تا امروز اغلب کاربردهای IDS به صورت SBS بوده است، اگرچه برخی از ویژگی‌های ABS را نیز شامل می‌شوند.

- **سیستم snort**

این سیستم یک IDS آزادافزار است که در حال حاضر شامل صدها هزار کاربر می‌باشد. این نرم‌افزار از طریق سیستم عامل‌های Linux، Unix و ویندوز قابل اجرا می‌باشد. این سیستم IDS از یک واسط شنود ژنریک بهره می‌گیرد، که در بسیاری از سیستم‌های شنود بسته نیز متداول می‌باشند. این سیستم قادر به پردازش ترافیک 100 مگابیت بر ثانیه است و به‌منظور بهره‌گیری آن در سرعت‌های در مقیاس گیگابیت بر ثانیه بایستی از سیاست چندین حسگر snort بهره گرفت. به‌منظور درک بهتر snort به بررسی مثال ذیل که در قالب یک امضا می‌باشد، دقت نمایید:

```
alert icmp $EXTERNAL-NETany → $HOME-NET any (msg:"ICMP PING NMAP";
dsize:0; itype:8,)
```

این امضا در مثال عنوان شده با هر بسته‌ی ICMP که وارد شبکه‌ی نهاد یعنی ($HOME-NET) از طریق یک شبکه‌ی خارجی یعنی ($EXTERNAL-NET) می‌شود و از نوع 8 یعنی (ICMP ping) می‌باشد و دارای یک محموله‌ی تهی (disze=0) است، قابل نگاشت می‌باشد. از آن جایی‌که nmap به تولید بسته‌های پینگ با این مشخصات ویژه می‌پردازد، طراحی این امضا نیز به‌منظور تشخیص جریانی از پینگ‌های nmap می‌باشد. هنگامی که یک بسته با این امضا نگاشت می‌شود، نرم‌افزار snort به تولید یک هشدار می‌پردازد که حامل پیام "ICMP PING NMAP" است.

شاید جالب‌ترین موضوع در رابطه با snort آن است که انبوهی از کاربران و متخصصان امنیت پایگاه داده امضاهای آن را نگه‌داری می‌کنند و بنابراین در فاصله‌ی چندساعت از یک تهاجم جدید، جامعه‌ی snort امضای تهاجمی آن را طراحی و از طریق صدها هزار توزیع کننده‌ی snort در سراسر جهان در اختیار عموم قرار می‌دهند. علاوه بر آن بهره‌گیری از قوانین امضای snort سبب می‌شود، مدیران شبکه قادر به طراحی امضاهای ویژه‌ی شبکه‌ی نهاد خود باشند.

۸-۲۱ روش‌های جدید اعتبارسنجی سراسری در تأمین امنیت شبکه

عملیات **اعتبارسنجی نقطه‌ی انتهایی** (سراسری) فرآیندی است که در آن یک موجودیت هویت

خود را به موجودیتی دیگر از طریق یک شبکه‌ی کامپیوتری معرفی می‌کند. به‌عنوان مثال یک کاربر تأمین‌کننده‌ی هویت به یک سرور پست الکترونیکی از جمله‌ی این موارد است. به‌عنوان انسان‌ها ما اعتبارسنجی یا تشخیص هویت خود را از راه‌های مختلف به یکدیگر ثابت می‌کنیم. از جمله از طریق چهره هنگامی که یکدیگر را ملاقات می‌کنیم و یا از طریق تشخیص صدا در یک مکالمه‌ی تلفنی و سرانجام از طریق مدارک شناسایی مانند گذرنامه و غیره این اعتبارسنجی صورت می‌پذیرد.

در این بخش ملاحظه خواهد شد چگونه یک طرف، اعتبارسنجی طرف دیگر را از طریق برقراری ارتباط در یک شبکه کامپیوتری انجام می‌دهد. در این بخش تمرکز موضوع روی تشخیص هویت یک طرف اصطلاحاً زنده یا فعال است آن هم در نقطه‌ای از زمان که ارتباط واقعی در جریان می‌باشد. نمونه‌ای از این مورد هنگامی است که یک کاربر اعتبارسنجی را با یک سرور ایمیل از طریق معرفی خود انجام می‌دهد. این مهم به‌صورت زیرکانه‌ای بسیار متفاوت با تأمین یک پیام دریافت شده در نقطه‌ای از زمان درگذشته است که از سوی یک فرستنده که ادعای ارسال آن را دارد و در بخش ۸-۳ از همین فصل مورد مطالعه قرار گرفت، به‌وقوع پیوسته است.

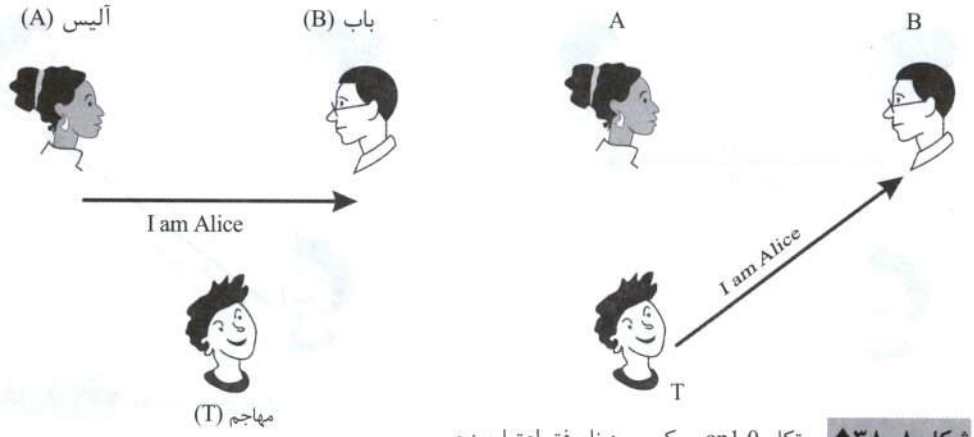
هنگامی‌که اعتبارسنجی از طریق شبکه صورت می‌گیرد، دو طرف ارتباط نمی‌توانند روی اطلاعات بیولوژیکی یکدیگر مانند چهره، صدا و یا اثر انگشت تکیه کنند، بلکه همان‌گونه که قبلاً ملاحظه گردید و در این بخش نیز مشاهده خواهد شد اغلب عناصر شبکه مانند روترها و فرآیندهای سرور – مشتری هستند که بایستی اعتبارسنجی یکدیگر را انجام دهند. در این مقطع اعتبارسنجی بایستی به تنهایی بر پایه‌ی پیام‌ها و داده‌های مبادله شده به‌عنوان بخشی از یک **پروتکل اعتبارسنجی** پایه‌ریزی گردد. به شکل ویژه یک پروتکل اعتبارسنجی **قبل از** انجام هر ارتباط بین دو طرف به اجرا در می‌آید و احتمالاً دو طرف ممکن است بعداً تمایل به استفاده از پروتکل‌های دیگری مانند پروتکل انتقال مطمئن داده یا پروتکل تبادل اطلاعات مسیریابی و یا یک پروتکل ایمیل مانند POP3 را داشته باشند. پروتکل اعتبارسنجی نخست اقدام به تشخیص هویت دو طرف خواهان ارتباط و اطمینان از تمایل آنها به انجام این مهم می‌نماید و سپس دو طرف وارد فعالیت اصلی مورد نظر خویش می‌شوند.

مانند مورد انتقال مطمئن داده (rdt) در فصل ۳ بهینه است اگر در این بخش نیز نسخه‌های مختلف یک پروتکل اعتبارسنجی که به اختصار آن را **ap**[1] می‌نامیم توسعه و تکمیل نمائیم. در طول تکامل پروتکل اعتبارسنجی موردنظر فرض می‌شود فرد A (Alice) نیازمند اعتبارسنجی هویت خود به فرد دیگری به نام B (Bob) می‌باشد.

۱- پروتکل اعتبارسنجی ap1.0

شاید یکی از ساده‌ترین پروتکل‌های اعتبارسنجی که می‌توان تصور نمود حالتی است که A یک پیام به سوی B گسیل می‌دارد و در آن اظهار می‌کند او فرد A می‌باشد. این پروتکل در شکل ۸-۳۸ نشان داده شده است. جریان اطلاعات مبادله شده آشکار است. هیچ حالتی وجود ندارد که B اطمینان حاصل نماید پیام ارسالی که به‌صورت «من آلیس هستم» یقیناً از سوی فردی گسیل شده است که ادعا می‌کند A می‌باشد. به‌عنوان مثال مهاجم (Trudy) یا همان T ممکن است ارسال‌کننده‌ی این پیام باشد.

1- Authenntication protocol



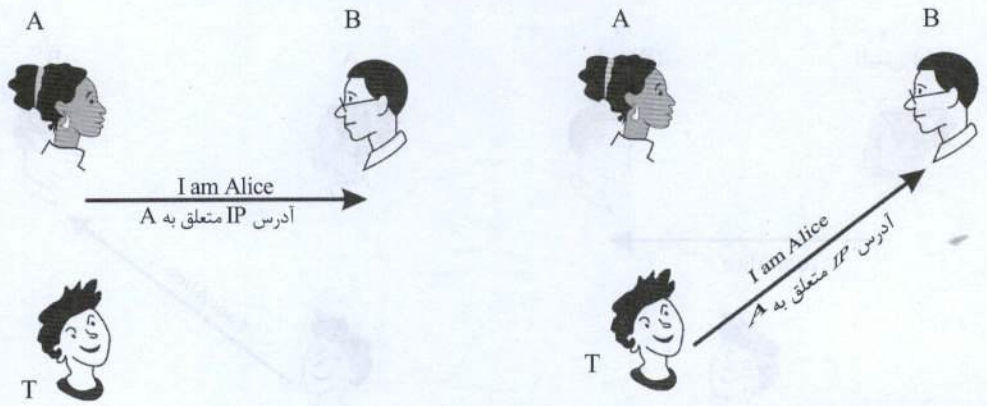
شکل ۸-۳۸♦ پروتکل ap1.0 و یک مورد ناموفق اعتبارسنجی.

۲- پروتکل اعتبارسنجی ap2.0

اگر A بهره‌مند از یک آدرس شبکه‌ی شناخته شده (مشهور) مانند یک آدرس IP باشد که همواره از طریق آن ارتباط برقرار می‌کند آنگاه B می‌تواند تلاش نماید تا اعتبارسنجی A را از طریق تأیید آدرس مبدأ در داده گرام IP حامل پیام اعتبارسنجی انجام دهد. نگاشت آن آدرس با آدرس شناخته شده‌ی A دلیلی بر ادعای صحت A می‌باشد. در این حالت اعتبارسنجی A با موفقیت انجام می‌گیرد. این روش ممکن است مانع یک مهاجم نادان مانند T در جهت جعل هویت خود به جای A شود اما نمی‌تواند مانع دانشجویان خواننده‌ی این اثر و بسیاری از افراد زیرک دیگر باشد!

از مطالعه‌ی قبلی لایه‌های پیوند داده و نیز شبکه دریافتیم که چندان هم دشوار نیست که یک داده گرام IP خلق شود و هر آدرس IP مبداء دلخواه را درون آن داده گرام جایگذاری نموده و سپس از طریق پروتکل لایه‌ی پیوند آن را به نخستین روتر دریچه گسیل نمود. به‌عنوان مثال اگر دسترسی به کدهای سیستم عامل میسر باشد و بتوان هسته‌ی OS خود را مانند Linux و دیگر سیستم عامل‌های دارای کُد آزاد خلق نمود آنگاه به آسانی آدرس شناخته‌شده‌ی فردی مانند A را می‌توان در یک داده گرام IP جایگذاری و سپس آن را به سوی نخستین روتر hop گسیل نمود. از این مقطع به بعد داده گرام با آدرس IP مجعول خود با موفقیت به سوی B گسیل شده و وظیفه‌ی خود که جعل هویت A است را با موفقیت انجام می‌دهد. این رویکرد در شکل ۸-۳۹ نشان داده شده است.

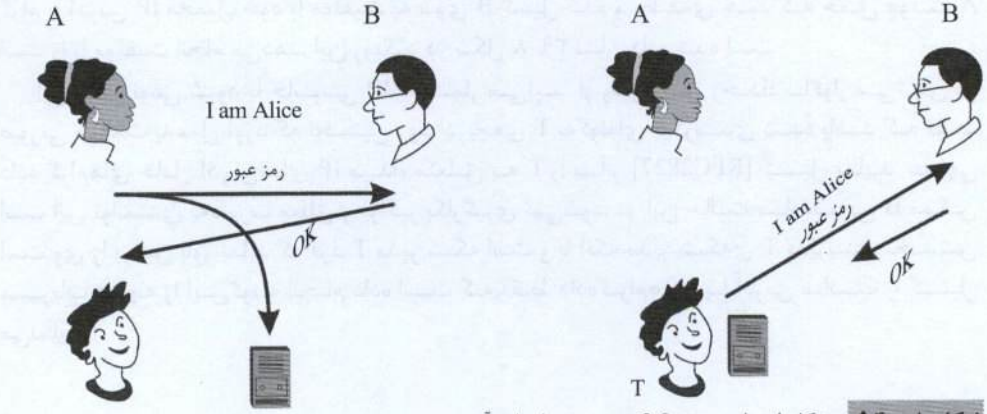
این روش نوعی شنود یا جاسوسی IP به شمار می‌آید. از وقوع این رخداد ناگوار می‌توان در صورتی ممانعت به‌عمل آورد که نخستین روتر دریچه‌ی T به‌گونه‌ای پیکربندی شده باشد که فقط داده گرام‌های حامل آدرس‌های IP مبداء متعلق به T را بنابر [RFC2827] گسیل نماید. بدیهی است این توانمندی به‌صورت مطلق و فراگیر بکارگیری نمی‌شود. در این حالت ساده‌لوحی B ممکن است وی را بر این باور نماید که فرد T مدیر شبکه است و یا آنکه مدیر شبکه‌ی T پیکربندی نخستین مسیریاب دریچه را این‌گونه انجام داده است که فقط داده‌گرام‌های با آدرس مناسب را گسیل می‌نماید.



شکل ۸-۳۹♦ پروتکل اعتبارسنجی ap3.0 و سناریوی ناموفق آن.

۳- پروتکل اعتبارسنجی ap3.0

یک رویکرد کلاسیک اعتبارسنجی استفاده از گذر واژه‌ی سری است. رمز عبور در این حالت یک سرّ مشترک بین اعتبارسنج و فردی است که قرار است تشخیص هویت گردد. خدمات Gmail، فیس‌بوک، تلنت، FTP و بسیاری دیگر از این‌گونه خدمات از رمز عبور یا گذر واژه به‌منظور اعتبارسنجی بهره‌گیری می‌کنند در پروتکل ap3.0 فرد A بایستی گذر واژه‌ی سری خود را به سوی B بر مبنای شکل ۸-۴۰ گسیل نماید. از آنجا که از رمزهای عبور به‌صورت گسترده استفاده می‌شود ممکن است تصور شود که پروتکل ap3.0 کاملاً ایمن است. بدیهی است این تصور کاملاً خطا است! ضعف امنیتی این روش روشن است. اگر T اقدام به شنود ارتباطات A نماید آنگاه او می‌تواند رمز عبور A را یادگیری نماید. به‌منظور آشکار شدن این حفره‌ی امنیتی این حالت را درنظر بگیرید که در حال انجام عملیات تلنت (Telnet) با ماشین دیگری هستید و بنابراین اطلاعات مربوط به هویت کاربرد رمز عبور را تلنت می‌نمایید. در این مقطع رمز عبور به‌صورت خام (رمزگذاری نشده) به سوی سرور تلنت گسیل می‌شود و در صورتی‌که فردی به شبکه‌ی LAN مشتری یا سرور تلنت متصل باشد به‌آسانی تمامی بسته‌های انتقال یافته روی LAN را ذخیره‌سازی و یا مطالعه می‌کند و بنابراین رمزهای عبور را به‌آسانی سرقت می‌کند. این روش بسیار شناخته شده‌ای برای سرقت گذر واژه‌ها می‌باشد بنابراین پروتکل ap3.0 نیز از نقاط ضعف قابل ملاحظه‌ای برخوردار است.



شکل ۸-۴۰♦ پروتکل اعتبارسنجی ap3.0 و مورد ناموفق آن.

۴- پروتکل اعتبارسنجی ap3.1

تفکر بعدی زدودن عیب آشکار پروتکل ap3.0 است. بدیهی است امنیت رمز عبور را در این حالت می‌توان با رمزگذاری گذر واژه به انجام رسانید. با این عمل از شنود و یادگیری T جلوگیری به‌عمل می‌آید. اگر فرض شود A و B هر دو از یک کلید متقارن K_{A-B} استفاده می‌کنند آنگاه A می‌تواند گذر واژه‌ی خود را ابتدا رمزگذاری و سپس پیام هویت خود یعنی «من آلیس هستم» را به‌سوی B گسیل دارد. پس از آن B رمز عبور را رمزگشایی و در صورتی‌که صحیح باشد A را اعتبارسنجی می‌کند. در این حالت B در اعتبارسنجی A احساس آرامش کامل دارد زیرا نه‌تنها A رمز عبور را می‌داند بلکه از کلید رمز مشترک نیز که به‌منظور رمزگذاری گذر واژه لازم است آگاهی دارد. فرض کنید این پروتکل ap3.1 نامیده شود.

در حالی‌که ap3.1 از یادگیری رمز عبور A از سوی T ممانعت به‌عمل می‌آورد اما بهره‌گیری از رمزگذاری مجدد گذر واژه نمی‌تواند چالش اعتبارسنجی را برطرف نماید زیرا B ممکن است در معرض **تهاجم بازپخش** قرار گیرد. به سخن دیگر کافی است T شنود ارتباطات A را انجام داده و سپس نسخه‌ی رمزگذاری شده‌ی گذر واژه را ثبت نماید و آن را مجدداً به سوی B گسیل نماید و وانمود کند که وی A می‌باشد. بهره‌گیری از یک گذر واژه‌ی رمزگذاری شده در پروتکل ap3.1 نیز تفاوتی را نسبت به ضعف پروتکل ap3.0 بر مبنای شکل ۸-۴۰ ایجاد نمی‌کند.

۵- پروتکل اعتبارسنجی ap4.0

شکست سناریوی شکل ۸-۴۰ از این حقیقت سرچشمه می‌گیرد که فرد B قادر به تمایز بین اعتبارسنجی اصلی از A و بازپخش مجدد اعتبارسنجی واقعی A نبود. به سخن دیگر اظهار این موضوع که آیا A تعامل زنده دارد و واقعاً در حال حاضر در طرف دیگر ارتباط است و یا آنکه پیام دریافتی از قبل ثبت شده و اکنون بازپخش می‌شود، برای فرد B غیرممکن است. خواننده‌ی هوشیار به‌خاطر می‌آورد پروتکل دست‌دهی سه جانبه‌ی TCP نیز مایل به حل چالشی مشابه بود زیرا سمت سرور یک ارتباط TCP برای پذیرش یک ارتباط در صورتی که قطعه‌ی SYN دریافتی نسخه‌ی قدیمی یا به عبارت دیگر انتقال مجدد یک قطعه SYN از ارتباط پیشین بود، تمایلی نداشت. چگونه سرور TCP مسئله‌ی زنده بودن واقعی برنامه‌ی مشتری را تشخیص می‌داد؟ پاسخ آن است که یک ارزش زنجیره‌ی اولیه‌ای را که مدت طولانی از استفاده‌ی آن گذشته بود گزینش می‌نمود و سپس آن ارزش را به‌سوی برنامه‌ی مشتری ارسال می‌نمود و در انتظار پاسخ مشتری در قالب یک قطعه‌ی ACK که شامل آن ارزش می‌شد، قرار می‌گرفت. همین اندیشه را می‌توان برای اهداف اعتبارسنجی نیز مورد استفاده قرار داد.

یک شماره ردیف موقت[1] که قبلاً نیز در این فصل به آن اشاره رفته است ارزشی است که یک پروتکل فقط یک مرتبه از آن در یک دوره‌ی حیات بهره‌گیری می‌کند. به سخن دیگر هنگامی که یک پروتکل از شماره ردیف موقت بهره گرفت هرگز مجدداً آن را مورد استفاده قرار نمی‌دهد. پروتکل ap4.0 پیشنهادی نیز از یک شماره ردیف موقت، به صورت ذیل بهره‌گیری می‌کند:

۱- فرد A پیام «من آلیس هستم» را به B ارسال می‌کند.

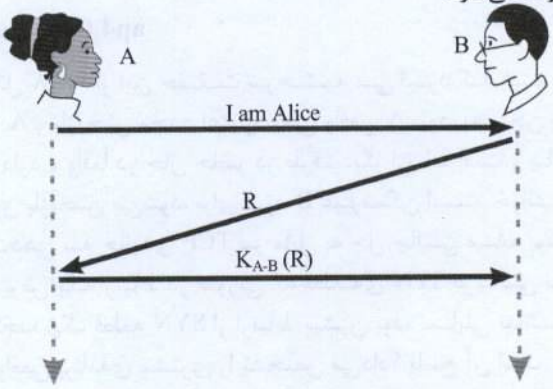
۲- فرد B یک شماره ردیف موقت گزینش و آن را R می‌نامد و به‌سوی A گسیل می‌دارد.

1- Nonce

۳- فرد A ارزش ارسالی R را با استفاده از کلید رمز متقارن A و B یعنی: K_{A-B} رمزگذاری می‌کند و ارزش موقت رمزگذاری شده‌ی $K_{A-B}(R)$ را به B بازگشت می‌دهد. مانند پروتکل ap3.1 این حقیقت وجود دارد که A از وجود کلید K_{A-B} آگاه است و از آن به‌منظور رمزگذاری یک ارزش استفاده می‌کند که برای B این امکان را فراهم می‌آورد که مطمئن باشد پیام دریافتی از سوی A بوده است. از ارزش موقت به منظور اطمینان، از فعال بودن (زنده بودن) A استفاده می‌شود.

۴- پس از دریافت پیام، B آن را رمزگشایی می‌کند و اگر ارزش موقت رمزگشایی شده برابر با آنچه A فرستاده است باشد آنگاه اعتبارسنجی A از سوی B به پایان رسیده است.

پروتکل ap4.0 در شکل ۸-۴۱ به نمایش درآمده است. با بهره‌گیری از ارزش R فقط برای یک مرتبه و سپس ارزیابی ارزش بازگشتی: $K_{A-B}(R)$، فرد B اطمینان حاصل می‌کند که A واقعاً همان فردی است که ادعا می‌کند یعنی از ارزش کلید رمز آگاه است و نیز فعال و زنده است زیرا ارزش R را رمزگذاری نموده است که لحظاتی پیش B آن را خلق نموده بود. بهره‌گیری از یک ارزش موقت و کلید رمزگذاری متقارن مبنای پروتکل ap4.0 است. تنها یک پرسش دیگر باقی می‌ماند و آن عبارت است از این مهم که آیا می‌توان از یک ارزش موقت و یک کلید عمومی رمزگذاری به جای کلید متقارن بهره گرفت تا چالش اعتبارسنجی را برطرف نمود؟ این موضوعی است که پاسخ مطلوب آن برعهده‌ی خواننده‌ی علاقه‌مند گذارده می‌شود.



شکل ۸-۴۱♦ پروتکل ap4.0 و عملکرد موفق آن.

تمرین‌های فصل هشتم

۱- تفاوت بین یک مهاجم فعال و خنثی چیست؟

۲- چه تفاوتی مابین محرمانگی پیام و یکپارچگی پیام می‌باشد؟ آیا بهره‌گیری یکی بدون دیگری ممکن است؟

۳- فرض کنید تعداد N فرد مایل به ارتباط با هریک از N-1 فرد دیگر می‌باشند و مایل به انجام این ارتباط‌ها با بهره‌گیری از کلید رمز متقارن هستند. همه‌ی ارتباط‌های بین هر دو فرد i و j برای بقیه‌ی افراد قابل رؤیت در گروه N می‌باشد و هیچ فرد دیگری خارج از این گروه قادر به

رمزگشایی ارتباط آن‌ها نمی‌باشد. حساب کنید چه تعداد کلید در این سیستم لازم است؟ اگر فرض شود از کلید رمز عمومی استفاده گردد، آن‌گاه چه تعداد کلید لازم می‌باشد؟

۴- فرض کنید یک مهاجم پیام رمز شده و نیز بازگشایی شده‌ی آن را در اختیار دارد. این مهاجم قادر به انجام کدام یک از تهاجم‌های از نوع متن رمز تنها یا متن خام شناخته شده و یا متن خام گزینش شده می‌باشد؟

۵- در چه روشی یک هاش، اعتبارسنجی یکپارچگی پیام بهتری را در مقابل کد مجموع مقابله مانند کد مجموع مقابله‌ی اینترنت تأمین می‌نماید.

۶- هدف یک پیام موقت (nonce) در یک پروتکل اعتبارسنجی نقطه‌ی انتهایی چیست؟

۷- تفاوت مهم بین یک سیستم کلید متقارن و یک سیستم کلید عمومی چیست؟

۸- بیان این نکته که پیام موقت یک ارزش است که فقط در یک دوره‌ی زمانی حیات دارد، چه مفهومی دارد؟

۹- مفهوم مقام گواهی کننده (CA) چیست؟

۱۰- تهاجم فرد میانی چیست؟ آیا این تهاجم هنگامی که از کلیدهای متقارن استفاده می‌شود، امکان‌پذیر است؟

۱۱- تفاوت‌های عمده‌ی تأمین خدمات دو پروتکل AH و ESP را در IPsec به اختصار توضیح دهید.

۱۲- آیا می‌توان هاش یک پیام را رمزگشایی نمود تا پیام اصلی را از آن استنتاج نماید؟ توضیح دهید.

۱۳- مفهوم آنکه یک سند امضا شده قابل اعتبارسنجی و غیرقابل جعل است، چیست؟

۱۴- درچه مواردی هاش پیام رمز کلید عمومی، امضای دیجیتالی بهتری را در مقایسه با پیام رمز کلید عمومی فراهم می‌آورد؟

۱۵- نشان دهید تهاجم متن خام شناخته شده‌ی مهاجم T که در آن او از ترجمه‌ی به‌صورت جفت تعداد 7 حرف متن رمز و متن خام آگاهی دارد سبب کاهش تعداد جایگزینی‌های ممکن به‌منظور ارزیابی در مثال بخش ۸-۳ به اندازه‌ی 10^9 می‌شود.

۱۶- با استفاده از رمز تک الفبا در شکل ۸-۳ دو پیام ذیل را به ترتیب ابتدا رمز نموده و سپس رمزگشایی نمایید:

"This is an easy problem"

پیام رمزگشایی:

"rmiju uamu xyi"

۱۷- سیستم چند الفبایی شکل ۸-۴ را درنظر بگیرید. آیا یک تهاجم متن ساده که قادر به دریافت رمزگشایی متن خام از پیامی مانند ذیل است، برای رمزگشایی همه‌ی پیام‌ها کافی می‌باشد؟ در صورت پاسخ آری و یا خیر علت را توضیح دهید.

"The quick brown fox jumps over the lazy day"

۱۸- با بهره‌گیری از الگوریتم RSA و گزینش P=3 و q=11 و رمز کلمه‌ی "hello" الگوریتم رمزگشایی را به نسخه‌ی رمز شده به‌منظور بازیابی متن خام پیام اعمال نمایید.

۱۹- در تهاجم فرد واسطی شکل ۸-۲۱ اگر Alice اعتبارسنجی Bob را انجام نداده باشد و اگر Alice مایل باشد Bob خود را با بهره‌گیری از ap5.0 اعتبارسنجی نماید، آیا این موضوع باعث ممانعت از تهاجم فردمیانی می‌شود؟ علت را توضیح دهید.

۲۰- پروتکل اعتبارسنجی ap4.0 را درنظر بگیرید که Alice خود را برای Bob اعتبارسنجی می‌کند. اکنون فرض کنید هنگامی که Alice اعتبارسنجی خود را به Bob انجام می‌دهد، Bob نیز بایستی خود را به Alice اعتبارسنجی نماید. سناریویی را ارایه نمایید که T تظاهر می‌کند Alice می‌باشد و خود را به عنوان Alice به Bob اعتبارسنجی می‌کند. یادآوری می‌شود ترتیب عملیات ap4.0 که T آغازگر است و دیگری که Bob آغازگر می‌باشد. این دو به صورت تصادفی نیز قابل جای‌گذاری می‌باشند. به این نکته توجه نمایید که هر دو Bob و Alice از یک پیام موقت بهره می‌گیرند و اگر دقت لازم انجام نگیرد، از همان پیام موقت سو استفاده خواهد شد.

۲۱- پروتکل مسیریابی اینترنت (PGP) از یک MAC به جای کلید رمز عمومی جهت امضای پیام‌های BGP استفاده می‌کند. به نظر شما علت آنکه از یک MAC در مقابل کلید رمز عمومی استفاده شده است، چیست؟

۲۲- یک جدول فیلتر و یک جدول ارتباط را برای یک فایروال SF تعریف کنید که تا سرحد امکان محدود کننده باشد، اما قادر به انجام عملیات ذیل است:

۱- همه‌ی ترافیک درون و خارج از باند را مسدود می‌کند.

۲- اجازه می‌دهد همه‌ی کاربران داخلی تعامل‌های Telnet را با میزبان‌های خارجی برقرار نمایند.

۳- اجازه می‌دهد همه‌ی کاربران خارجی به جستجوی سامانه‌ی وب در آدرس 222.22.0.12:IP بپردازند.

شبکه‌ی داخلی دارای آدرس IP به صورت 222.22/16 می‌باشد. در راهکار خود فرض کنید که جدول ارتباط در حال حاضر سه ارتباط را نهان‌سازی می‌کند که همه‌ی آن‌ها از درون و برون می‌باشند و شما نیازمند آدرس‌های IP و شماره پورت‌های مناسب می‌باشید.

۲۳- شکل ۸-۲۴ عملیاتی را نشان می‌دهد که Alice بایستی به‌منظور تأمین محرمانگی، اعتبارسنجی و یکپارچگی انجام دهد. شکل عملیات مربوطه‌ای که Bob بایستی برروی بسته‌ی دریافتی از Alice انجام دهد را ترسیم نمایید.

۲۴- پیام سومی را محاسبه نمایید متفاوت از دو پیام شکل ۸-۸ که دارای کد مجموع مقابله همانند پیام‌های شکل ۸-۸ می‌باشد.

۲۵- اگر IPsec امنیت لایه‌ی شبکه را تأمین می‌کند، چرا روش‌های امنیتی هنوز هم مورد نیاز لایه‌های بالاتر IP می‌باشد؟



فصل ۹

مدیریت شبکه‌های کامپیوتری

پس از مطالعه‌ی ۸ فصل از مطالب این اثر بایستی این نتیجه آشکار شده باشد که شبکه‌های کامپیوتری شامل تعداد بی‌شماری ادوات سخت‌افزاری و نرم‌افزاری پیچیده می‌باشند که همگی با یکدیگر در تعامل قرار می‌گیرند. این ابزارهای گوناگون شامل پیوندها، راهیاب‌ها، روترها، میزبان‌ها و دیگر ادواتی است که عناصر فیزیکی شبکه و انبوهی از پروتکل‌ها در هر دو حالت سخت و نرم‌افزاری را تشکیل می‌دهند. این گونه تجهیزات را مدیر شبکه کنترل و هماهنگ می‌کند. هنگامی که صدها هزار گونه از این ادوات از سوی یک نهاد بزرگ به یکدیگر تنیده شده‌اند تا تشکیل یک شبکه را دهند، آن‌گاه مایه‌ی حیرت نخواهد بود که این قطعه‌ها در برخی مواقع دچار عیب شده و یا از لحاظ پیکربندی با اشکال‌های چندی روبه‌رو و منابع آن‌ها دچار بهره‌برداری بیش از توان آن‌ها شوند. سرانجام آنکه به سادگی آن‌ها معیوب شده و دچار از کارافتادگی گردند. به عنوان نمونه هنگامی که یک کابل دچار قطعی می‌شود و یا آنکه لیوان نوشابه‌ای روتر را شست‌وشو می‌دهد! وظیفه‌ی مدیر شبکه نگهداری و مدیریت شبکه و منابع آن می‌باشد. با توجه به گستردگی عناصر شبکه در یک محدوده‌ی جغرافیایی وسیع مدیر شبکه در یک مرکز مدیریت عملیات شبکه (NOC) نیازمند ابزاری به‌منظور نظارت، کنترل و مدیریت شبکه می‌باشد. هدف این فصل، بررسی و تحلیل معماری، پروتکل‌ها و اطلاعاتی است که برمبنای آن یک مدیر شبکه این وظایف را انجام می‌دهد.

۹-۱ مدیریت شبکه چیست؟

قبل از گام نهادن به موضوع مدیریت شبکه به تحلیل چند نمونه‌ی واقعی از مواردی می‌پردازیم که شامل سیستم‌های پیچیده با انبوهی از عناصر تعامل‌گرا است، که بایستی از سوی یک مدیر نظارت، مدیریت و کنترل شوند. این سناریوها لزوماً به صورت شبکه نمی‌باشند. به عنوان نمونه نیروگاه‌های تولید الکتریسیته که دارای یک اتاق کنترل و مدیریت می‌باشند و در آن اتاق انواع ابزارهای اندازه‌گیری،

سنجش و نظارت مانند دما، فشار و جریان در شیرها، لوله‌ها و شریان‌ها گرفته تا نوسانات ولتاژ، شدت جریان و بسامد در خطوط نصب شده‌اند. این تجهیزات امکانات لازم مدیریت عملیات و هشدارهای لازم در مواقع اضطراری را در اختیار مدیران کنترل نیروگاه قرار می‌دهند. نمونه‌ی مشابه غیرشبکه‌ی دیگر اتاقک کنترل هواپیما در دماغه‌ی آن می‌باشد. انبوهی از ابزارهای نظارت و سنجش در این بخش حساس از هواپیما نصب شده‌اند تا خلبان به کمک آن‌ها نظارت و کنترل دیگر بخش‌های هواپیما را از نقطه‌نظر عملیاتی تحت نظر قرار دهد. در این دو نمونه **مدیر** نظارت بر تجهیزات دور از دسترس **تحلیل** داده‌های برآمده از آن‌ها را برعهده می‌گیرد تا اطمینان حاصل نماید که همگی آن‌ها در حال کار می‌باشند و عملکرد آن‌ها در محدوده‌های تعریف شده‌ی قابل قبول انجام می‌پذیرد. به عنوان مثال میله‌های سوخت اتمی نیروگاه هسته‌ای در حال ذوب شدن نمی‌باشند و یا هواپیما دچار اشکال سوخت‌رسانی به موتورها نمی‌باشد. این نمونه از نظارت و کنترل یک مدیر اصطلاحاً **کنترل کنش‌گر (انفعالی)**[1] نامیده می‌شود زیرا اعمال تغییرات در نتیجه‌ی واکنش به رخدادهای درون سیستم و یا محیط اطراف آن انجام می‌گیرد. از سوی دیگر همان مدیر از طریق تشخیص رویکرد و رفتار غیرعادی در یک سیستم پیش‌بینی‌های لازم و حتی پیش‌گیری‌های مورد نیاز قبل از وقوع یک اشکال جدی در سیستم را مدیریت می‌نماید که اصطلاحاً به آن **مدیریت فعال**[2] گفته می‌شود. در رویکردی مشابه مدیر شبکه نیز نظارت، کنترل و مدیریت فعال یک شبکه را برعهده می‌گیرد.

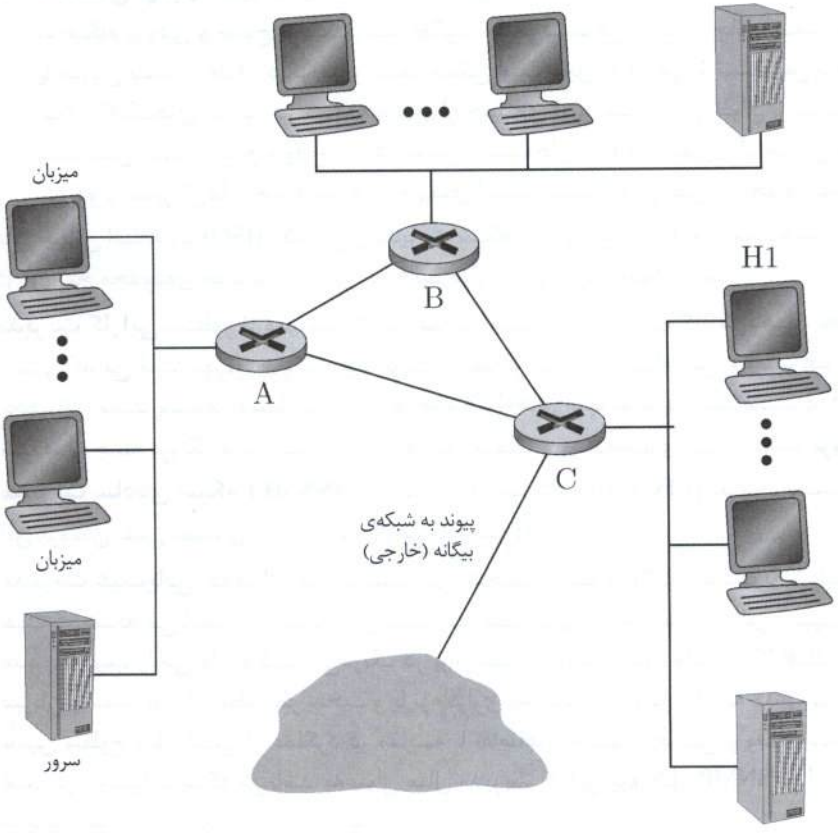
در روزهای نخستین شبکه‌های کامپیوتری که اغلب آن‌ها در حال پژوهش بودند تا زیرساخت‌های حساس امروزی که صدها میلیون کاربر روزانه از آن‌ها استفاده می‌کنند، مدیریت شبکه موضوع ناشناخته‌ای به‌شمار می‌آمد. اگر فردی دچار یک اشکال در شبکه می‌شد، اقدام به اعمال چند فرمان ping می‌نمود تا منبع و مکان عیب را شناسایی و سپس با اعمال تنظیمات لازم و راه‌اندازی مجدد سیستم‌های سخت‌افزاری و یا نرم‌افزاری اشکال را برطرف نماید. با رشد فزآینده‌ی شبکه‌ی اینترنت عمومی و اینترانت خصوصی از شبکه‌های در مقیاس کوچک به زیرساخت‌های عظیم جهانی، نیاز بر نظارت و مدیریت تعداد انبوهی از عناصر سخت و نرم‌افزاری در درون این شبکه‌ها هر روز بیش‌تر آشکار می‌شود. با تحلیل یک مثال ساده در شکل ۹-۱ شامل سه مسیریاب، تعدادی میزبان و سرور، سناریوهای گوناگونی که یک مدیر شبکه حتی در این مقیاس ممکن است با آن روبه‌رو باشد، موضوع مدیریت شبکه را بهتر می‌توان آشکار نمود. ابزار مناسب چنین شبکه‌ای می‌تواند شامل موارد ذیل باشد:

- **تشخیص عیب یک کارت شبکه در یک میزبان یا مسیریاب**. با در اختیار داشتن ابزار مناسب مدیریت شبکه، یک عنصر شبکه مانند مسیریاب ممکن است به مدیر شبکه گزارش دهد یکی از کارت‌های واسط آن دچار عیب شده است. مدیر شبکه که در حال نظارت فعال بر وضعیت ترافیک می‌باشد ممکن است اشکال در کارت واسط شبکه را قبل از وقوع عیب تشخیص داده و آن را با قطعه‌ی جدیدی جایگزین نماید. این واکنش از سوی مدیر شبکه ممکن است به این دلیل روی دهد که وی ملاحظه نموده است تعداد خطاهای کد مجموع مقابله در قاب‌های ارسالی از سوی این کارت شبکه افزایش یافته است.
- **نظارت میزبان**. در این حالت مدیرشبکه ممکن است به صورت متناوب همه‌ی میزبان‌های شبکه را از نقطه‌نظر عملیاتی بودن آن‌ها نظارت و سنجش نماید. بار دیگر مدیر شبکه

1- Reactive control
2- Proactive control

ممکن است نظارت و تصمیم‌گیری فعال را اعمال نماید یعنی قبل از گزارش خرابی میزبان نسبت به برطرف نمودن آن پیش‌گیری لازم را انجام دهد.

- **نظارت بر وضعیت ترافیک شبکه به‌منظور بهره‌گیری از امکانات شبکه.** مدیر یک شبکه ممکن است الگوی ترافیک جاری بین مبدأ و مقصد را سنجش و نظارت نماید. به عنوان مثال با تغییر سرورها بین قطعه‌های LAN مقدار ترافیک عبوری بین چندین LAN را به صورت چشمگیری کاهش دهد. این موضوع سبب افزایش کارایی در شبکه بدون نیاز به افزودن تجهیزات و در نتیجه اعمال هزینه می‌باشد. از طرف دیگر از طریق نظارت بر بهره‌وری پیوندهای شبکه یک مدیر ممکن است از این موضوع آگاه گردد که قطعه‌ای از LAN و یا یک پیوند خارجی با تراکم حجم ترافیک روبه‌رو شده است و بنابراین نیازمند پهنای باند بیش‌تر است که نتیجه‌ی آن تحمیل هزینه‌ی مالی به شبکه نیز می‌باشد. مدیر شبکه ممکن است مایل به نظارت خودکار بر وضعیت ترافیک باشد، به این مفهوم که در هنگام وقوع سطحی از تراکم بر روی یک پیوند که فراتر از آستانه‌ی تعریف شده است، وی قادر به اختصاص پهنای باند لازم با هدف جلوگیری از ایجاد بحران تراکم ترافیک در شبکه باشد.



شکل ۹-۱ ♦ بهره‌گیری از مدیریت شبکه

- **تشخیص تغییرات سریع در جداول مسیریابی**. اختصاص، جایگزینی و تغییر مسیرها مرتباً سبب اعمال تغییرات در جداول مسیریابی می‌شوند. این تغییرات باعث ناپایداری در مسیریابی و یا فروپاشی پیکربندی در مسیریاب‌ها می‌گردد. قطعاً مدیر شبکه‌ای که پیکربندی مناسبی را برای یک روتر انجام نداده است مایل به کشف خطای خود قبل از فروپاشی شبکه می‌باشد.
- **نظارت بر سطح توافقی خدمات (SLA)**[1]. منظوراز سطح خدمات توافقی قراردادهایی است که به تعریف مقیاس‌های کارایی ویژه و سطوح قابل قبول از کارایی تأمین کننده‌ی شبکه با توجه به این تعاریف می‌پردازد. دو نمونه از انبوه تأمین کنندگان شبکه Verizon و Sprint می‌باشند که امکانات لازم جهت تأمین SLA را ضمانت می‌کنند. این SLAها شامل دسترسی به خدمات و به عبارت دیگر درصد خرابی در سیستم‌ها، تأخیر پیوندها، بازدهی شبکه و نیازمندی‌های اعلام عیب می‌باشند. بدیهی است اگر کارایی بخشی از یک توافق خدمات بین یک تأمین کننده‌ی شبکه و کاربران آن باشد آن‌گاه اندازه‌گیری و مدیریت کارایی یکی از موارد مهم موردنظر مدیر شبکه می‌باشد.
- **تشخیص تهاجم**. مدیر شبکه ممکن است مایل به آگاهی از فعالیت‌های مشکوک در شبکه به هنگام ورودی و خروج ترافیک باشد. فعالیت مشکوک ممکن است از جانب یک پورت و یا میزبان باشد. از طرف دیگر مدیر شبکه ممکن است مایل به آگاهی و تشخیص برخی از انواع ترافیک‌های موجود در شبکه باشد. از جمله‌ی این موارد می‌توان از بسته‌های مسیردهی شده از منبع و وقوع تعداد انبوهی از بسته‌های SYN به‌سوی یک میزبان نام برد. این دو و نظایر آن‌ها از جمله موارد تهاجم‌های امنیتی است که در فصل ۸ تحلیل شدند.

سازمان جهانی استاندارد (ISO) یک الگوی مدیریت شبکه را برای موارد اشاره شده تدوین نموده است و در آن پنج محدوده‌ی مدیریت شبکه را در قالب موارد ذیل تعریف نموده است:

۱- **مدیریت کارایی**. منظور از مدیریت کارایی عبارت است از اندازه‌گیری، گزارش‌دهی، تحلیل و کنترل کارایی مانند بهره‌وری و بازدهی هریک از عناصر گوناگون شبکه. این عناصر عبارتند از تجهیزاتی مانند پیوندها، مسیریاب‌ها، میزبان‌ها و انتزاع‌های انتها به انتها مانند مسیری از درون شبکه، از جمله پروتکل‌هایی که در این زمینه تعریف و پیاده‌سازی شده است **پروتکل مدیریت ساده‌ی شبکه (SNMP)** می باشد که در مستند [RFC3410] تعریف شده است. این پروتکل نقش عمده‌ای در مدیریت کارایی اینترنت دارد.

۲- **مدیریت عیب‌یابی**. هدف از مدیریت عیب‌یابی تشخیص، ثبت و واکنش نسبت به شرایط وقوع عیب در شبکه می‌باشد. خط تمایز بین مدیریت عیب‌یابی و مدیریت کارایی مبهم است. مدیریت عیب را می‌توان واکنش بی‌درنگ در برابر عیوب موقتی شبکه مانند از کارافتادن پیوند، میزبان یا مسیریاب از نقطه نظر سخت و یا نرم‌افزاری به شمار آورد، در حالیکه مدیریت کارایی تأمین سطوح قابل قبولی از عملکرد در مقایسه با تقاضاهای متغیر ترافیکی و وقوع به ندرت عیب در تجهیزات شبکه می‌باشد. به‌عنوان مثال مدیریت کارایی پروتکل SNMP در ارتباط با مدیریت عیب نقش مهمی را ایفا می‌کند.

1- Service level agrement

۳- **پیکربندی مدیریت.** آن‌چه پیکربندی مدیریت برای یک مدیر شبکه فراهم می‌آورد، عبارت است از رهگیری تجهیزاتی که شبکه مدیریت می‌نماید و نیز پیکربندی سخت و نرم‌افزاری این تجهیزات. چارچوب کلی پیکربندی مدیریت و نیازهای آن برای شبکه‌های مبتنی بر IP در مستند [RFC3139] موجود می‌باشد.

۴- **مدیریت حسابرسی.** مدیریت حسابرسی، امکان کنترل و ثبت کاربران و دستیابی تجهیزات به منابع شبکه را مدیریت می‌کند. نرخ‌گذاری بهره‌گیری از امکانات شبکه، هزینه‌گذاری بر مبنای بهره‌گیری و اختصاص منابع دستیابی به شبکه از جمله موارد زیرشاخه‌ی مدیریت حسابداری شبکه به‌شمار می‌آیند که در اختیار مدیر شبکه قرار می‌گیرد.

۵- **مدیریت امنیت شبکه.** هدف از مدیریت امنیت شبکه کنترل دستیابی به منابع شبکه براساس سیاست تعریف شده‌ی یک شبکه است. مراکز توزیع کلیدهای رمز که در فصل ۸ از آن‌ها نام برده شد، عناصر و عوامل مدیریت امنیت شبکه به‌شمار می‌آیند. بهره‌گیری از دیواره‌های آتش به‌منظور نظارت و کنترل نقاط دستیابی خارجی شبکه از جمله موارد دیگر مدیریت امنیت شبکه به‌شمار می‌روند.

در این فصل پایه‌های مدیریت شبکه تحلیل می شوند. پروتکل‌های زیرساخت مدیریت شبکه، معماری کلی و نیز اطلاعاتی که از طریق آن یک مدیر شبکه قادر به نظارت، کنترل و راهبرد یک شبکه‌ی کامپیوتری است از جمله مواردی می‌باشند که موضوع‌های تحلیلی این فصل را تشکیل می‌دهد. تعریف مدیریت شبکه در انتهای این بخش را به‌عنوان یک نتیجه‌گیری می‌توان چنین بیان نمود:

«مدیریت شبکه شامل بهره‌گیری، هماهنگی و یکپارچگی عوامل سخت‌افزاری، نرم‌افزاری و انسانی به‌منظور نظارت، سنجش، ارزیابی، پیکربندی، تحلیل و کنترل شبکه و منابع آن به‌منظور کارایی عملیاتی و نیازهای کیفیت خدمات (QoS) بی‌درنگ با حداقل هزینه می‌باشد.»

۹-۲ زیرساخت لازم برای مدیریت شبکه

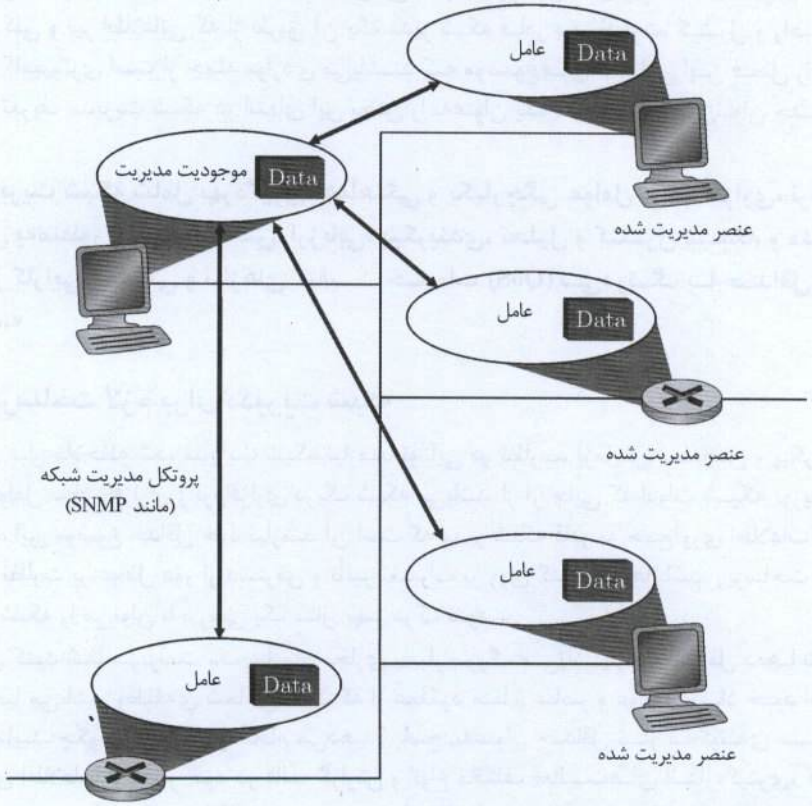
در بخش قبل ملاحظه شد، مدیریت شبکه نیازمند توانایی در نظارت، آزمایش، سنجش و پیکربندی و کنترل عوامل سخت‌افزاری و نرم‌افزاری دریک شبکه می‌باشد. از آن‌جایی که ادوات شبکه توزیع شده می‌باشند، این موضوع حداقل خود نیازمند آن است که مدیر شبکه قادر به جمع‌آوری اطلاعات موردنیاز به‌منظور نظارت بر عوامل دور از دسترس و تأثیر تغییرات بر روی کنترل آن‌ها باشد. زیرساخت موردنیاز مدیریت شبکه را می‌توان با بررسی یک مثال بهتر درک نمود.

فرض کنید شما سرپرست یک سازمان تجاری بسیار بزرگ می‌باشید که شامل ده‌ها شعبه در سراسر دنیا می‌باشد. وظیفه‌ی شما آن است که از عملکرد منظم عناصر و عوامل نهاد خود اطمینان حاصل نمایید. چگونه این عمل را انجام می‌دهید؟ پاسخ: به‌عنوان حداقل نیاز به‌گونه‌ی متناوب به جمع‌آوری اطلاعات از دفاتر خود در قالب گزارش و انواع مختلف فعالیت‌های اندازه‌گیری، کارایی و بودجه‌بندی می‌پردازید. گاهی اوقات و نه همیشه از اشکال‌های پدید آمده در یک شعبه آگاه می‌شوید. مدیر شعبه‌ای که علاقه‌مند به پیشرفت در رده‌ی شغلی خود می‌باشد، گزارش‌هایی را بدون درخواست از جانب مدیر اصلی به‌سوی او گسیل می دارد تا نشان دهد که همه چیز در آن شعبه با مدیریت و نظار

وی تحت کنترل می‌باشد. شما همه‌ی این گزارش‌ها را با این هدف بررسی می‌نمایید که اطمینان از کارکرد طبیعی و عادی دستگاه‌های تحت مسئولیت خود حاصل نموده و بدون شک اشکالی را که نیازمند توجه شما می‌باشد، در میان آن‌ها یافت نمایید. شما ممکن است به‌صورت رودررو اشکال پیش‌آمده در یک شعبه را تحلیل کنید، داده‌های بیش‌تری را جمع‌آوری و سپس دستورات لازم پیرامون اعمال تغییر را به مدیر شعبه به‌منظور برطرف نمودن مشکل صادر نمایید.

آن‌چه در این مثال بدیهی است، وجود یک زیرساخت به‌منظور کنترل سازمانی است که شما مسئول اصلی آن می‌باشید. عوامل این زیر ساخت شامل: سامانه‌های فاصله‌دار تحت نظارت شعبه‌ها و عوامل دوردست نظارتی شامل مدیران شعبه‌ها، پروتکل‌های ارتباطی به‌منظور جمع‌آوری اطلاعات و ارزیابی یک به یک داده‌ها، گزارش‌ها و فعالیت‌های نظارتی، کارایی و بودجه‌ای می‌باشند. هریک از این عوامل انسانی در مدیریت سازمانی دارای یک عامل مقابل خود در مدیریت شبکه می‌باشد.

معماری سیستم مدیریت شبکه از لحاظ اصولی مشابه مثال اشاره شده در بالا است. سامانه‌های مدیریت شبکه دارای واژگان خاص خود می‌باشد. همان‌گونه که در شکل ۹-۲ نشان داده شده است، سه عامل اصلی از معماری مدیریت شبکه عبارتند از: **مدیر اصلی** (مدیر اصلی شرکت)، **ادوات مدیریت شده** (شعبه‌های شرکت) و **پروتکل مدیریت شبکه**.



شکل ۹-۲ ♦ عناصر اصلی معماری مدیریت شبکه

عامل **مدیریت اصلی** یک کاربرد است که در حلقه‌ی آن یک عامل انسانی نیز وجود دارد. او مرکز

عملیات شبکه (NOC) را مدیریت و کنترل می‌کند. مدیر اصلی هسته‌ی فعالیت برای مدیریت شبکه به‌شمار می‌آید. کنترل، جمع‌آوری، پردازش، تحلیل و به نمایش گذاردن اطلاعات را مدیریت شبکه انجام می‌دهد. تمامی این فعالیت‌ها به‌منظور کنترل رفتار شبکه می‌باشد و مدیر انسانی شبکه نیز با تجهیزات شبکه برای این منظور تعامل برقرار می‌سازد.

یک **وسیله‌ی مدیریت شده** قطعه‌ای از تجهیزات شبکه است به‌همراه نرم‌افزار آن که در یک شبکه‌ی مدیریت شده ساکن می‌باشد. این مورد در مثال بالا همان دفتر شعبه‌ی شرکت است. یک ابزار مدیریت شده در شبکه امکان دارد یک میزبان، مسیریاب، چاپگر، HUB و یا مودم باشد. در درون وسیله‌ی مدیریت شده چندین **عنصر مدیریت شده** نیز وجود دارند. این عوامل مدیریت شده قطعه‌های اصلی از سخت افزار درون وسیله‌ی مدیریت شده مانند کارت شبکه و نیز مجموعه‌ای از عوامل پیکربندی برای قطعه‌های سخت افزاری و نرم‌افزاری مانند پروتکل‌های مسیریابی بین دامنه‌ها (RIP) می‌باشند. در مثال اشاره شده‌ی بالا عناصر مدیریت شده دپارتمان‌های درون دفاتر شعبه هستند. این عناصر مدیریت شده حامل بخشی از اطلاعات هستند که مجموعه‌ی جمع‌آوری شده‌ی آن‌ها **مبانی مدیریت اطلاعات (MIB)**[1] نامیده می‌شود. ارزش‌های این اطلاعات در بسیاری از موارد در دسترس عامل مدیریت شبکه می‌باشد. در مثال بالا MIB داده‌های کمیتی مانند اندازه‌گیری فعالیت‌ها، کارآمدی و بودجه هستند که بین دفاتر شعبه و دفتر مرکزی مبادله می‌شوند. در بخش ۹-۴ جزئیات MIB تحلیل خواهد شد. سرانجام در درون هر وسیله‌ی مدیریت شده یک **عامل مدیریت شبکه** وجود دارد. فرآیندی در حال اجرا در وسیله‌ی مدیریت شده که با عامل مدیریت در تعامل است و عملیات محلی را در یک وسیله‌ی مدیریت شده‌ی زیر نظارت و کنترل مدیر اصلی انجام می‌دهد. عامل مدیریت شبکه همان مدیر دفاتر شعبه در مثال اشاره شده می‌باشد.

قطعه‌ی سوم معماری مدیریت شبکه **پروتکل مدیریت شبکه** است. این پروتکل بین عامل اصلی مدیریت و قطعه‌ی مدیریت شده اجرا می‌شود و به مدیر اصلی اجازه‌ی پرسش از وضعیت عناصر مدیریت شده و اتخاذ غیرمستقیم عملیات از سوی عوامل مدیریت در این تجهیزات را می‌دهد. عامل‌های مدیریت می‌توانند با بهره‌گیری از پروتکل مدیریت شبکه عامل اصلی مدیریت را از رخدادهای غیرمتعارف مانند از کارافتادگی قطعه‌ها و عدم رعایت آستانه‌های کارایی آگاه نمایند. این نکته نیز حائز اهمیت است که پروتکل مدیریت شبکه خود به تنهایی شبکه را مدیریت نمی‌کند بلکه توان‌مندی لازم را برای مدیر اصلی شبکه به‌منظور اعمال مدیریت شبکه مورد استفاده قرار می‌دهد.

اگرچه زیرساخت مدیریت به نظر ساده می‌آید اما در اغلب موارد می‌تواند باعث گمراهی از نقطه نظر واژگانی چون **«عامل اصلی مدیریت»**، **«قطعه‌ی مدیریت شده»**، **«عامل‌های مدیریت»** و **MIB** شود. به عنوان مثال در مدیریت شبکه‌ی نظارت بر میزبان‌ها، عوامل مدیریت در محل و درون قطعه‌های مدیریت شده قرار دارند و به صورت متناوب از سوی عامل اصلی مدیریت مورد پرسش قرار می‌گیرند.

بحث مربوط به معماری مدیریت شبکه حالت ژنریک داشته و به شکل گسترده به تعدادی از استانداردهای مدیریت شبکه و تلاش‌های پیشنهاد شده در طول سال‌ها متمادی بازگشت می‌کند. بلوغ این استانداردها از اواخر دهه‌ی 1980 با ظهور پروتکل **عناصر خدماتی مشترک مدیریت اطلاعات** و یا **پروتکل مشترک مدیریت اطلاعات**[2] توصیف شده در مستند [RFC3410] می باشد. از آن‌جائیکه

1- Management information base

2- Common management information service element or common management information protocol

SNMP با شتاب، در زمانی طراحی و به کار گرفته شد که نیاز برای مدیریت شبکه آشکار شده بود، در نتیجه در حال حاضر SNMP دارای کاربرد گسترده‌ای می‌باشد. امروز SNMP از عوامل مهم چارچوب مدیریت شبکه به شمار می‌آید. در ادامه‌ی مطالب این فصل پروتکل SNMP با جزئیات کامل آن بررسی و تحلیل خواهد شد.

۹-۳ اصول در عمل – مرکز عملیات شبکه SPRINTLINKS

شبکه‌ها در همه‌ی ابعاد و مقیاس‌های گوناگون پدیدار می‌شوند از کوچک‌ترین آن شبکه‌ی خانگی تا بزرگ‌ترین آن یعنی Tier1-ISP ، کارکرد معقول و مطلوب عوامل شبکه و نیز کنترل و نظارت آن‌ها برعهده مدیر شبکه است. پرسش آن است که وقایعی که در مرکز عملیات شبکه (NOC) روی می‌دهد و چگونگی فرمان‌های راهبردی مدیر شبکه چیست؟

با یک شبکه‌ی تنیده در سراسر گیتی، شرکت SPRINT یکی از بزرگ‌ترین شبکه‌های Tier1-ISP در سراسر کره‌ی زمین است. معمولاً با نام SPRINT LINK شناخته می شود. شبکه‌ی آن دارای بیش از 70 نقطه‌ی حضور **(POP)** می باشد که شامل روترهای IP و مکان‌های ارتباط کاربران و شبکه با بیش از 800 مسیریاب است. این موضوع به مفهوم پهنای باند بسیار وسیع است! مرکز کنترل شبکه‌ی (NOC) شرکت SPRINT در ویگاس آمریکا و نقاط پشتیبانی آن در بیش از چهار ایالت آمریکا گسترش یافته است. انواع شبکه‌های زیرساخت تنیده در یکدیگر مرکز اصلی مدیریت شبکه‌ها را تشکیل می‌دهند. در هر لحظه‌ی زمانی تیمی از چهار مرکز عملیات شبکه نظارت و مدیریت تجهیزات در هسته‌ی شبکه‌ی IP این شرکت را عهده‌دار می‌باشند. یک تیم ذخیره نیز هرگونه گزارش از سوی مشتریان را در ارتباط با خرابی‌ها دنبال می‌کند. مکانیزه نمودن نظارت به‌منظور دریافت و تحلیل هشدارها، شناسایی و تشخیص عیوب و بازیابی خدمات، پیکربندی مدیریت و گزارش مشکلات مشتریان این امکان را برای گروهی کوچک از مدیران به‌منظور مدیریت و کنترل چنین شبکه‌ی عظیم و پیچیده‌ای فراهم آورده است.

هنگامی که اشکالی پدید می‌آید تمرکز اصلی مدیران عملیاتی SPRINT LINK متمرکز بر بازیابی سریع خدمات به مشتریان می‌باشد. راهبران NOC مراحل تشخیص عیب و بازیابی خدمات را برمبنای مجموعه‌ای از گام‌های عملیاتی در ارتباط با چالش پیش آمده انجام می‌دهند. چالش‌هایی که بی‌درنگ قابل تشخیص ویا در قالب زمانی تعریف شده‌ی 15 دقیقه امکان رفع عیب را ندارند، به سطح بالاتری از پشتیبانی در مرکز پشتیبانی فنی ملی (NTAC) انتقال می‌یابند. اعضای NTAC مسئول ورود به جزئیات عمیق‌تر ریشه‌های عیب ایجاد شده می‌باشند. مراحل عملیاتی NOC از سوی NTAC تعریف می‌شود و مرتباً در تعامل با تأمین کنندگان تجهیزات می‌باشد، تقریباً نزدیک به 90 درصد اشکالات پدید آمده از سوی عوامل NOC برطرف می‌شود. کارکنان NOC و NTAC هماهنگ با یکدیگر عملیات را کنترل و نظارت می‌کنند.

۹-۴ چارچوب مدیریت استاندارد اینترنت

برعکس آن‌چه نام پروتکل ساده‌ی مدیریت شبکه (SNMP) ممکن است در ذهن تعریف کند. مدیریت شبکه در اینترنت به مراتب بیش از آن است که یک پروتکل برای مدیریت جابه‌جایی داده‌ها بین یک موجودیت مدیریت و عوامل آن انجام می‌دهد. موضوع دارای رشد سریع و به مراتب پیچیده‌ای در مقیاس با واژه‌ی «ساده» می‌باشد. **چارچوب مدیریت استاندارد اینترنت (ISMF)**[1] در حال حاضر

1- Internet standard management framework

دارای ریشه‌ای است که از نتیجه‌ی **پروتکل نظارت ساده‌ی مسیریاب دریچه (SGMP)** سرچشمه می‌گیرد و در مستند [RFC1028] توصیف شده است. پروتکل SGMP از سوی گروهی از پژوهشگران دانشگاهی شبکه، کاربران و مدیران طراحی گردید که نتایج حاصل از تجربیات آن‌ها منجر به طراحی و پیاده‌سازی پروتکل SNMP گردید. از زمان طراحی SNMP تا امروز دوره‌ی تکامل این پروتکل از SNMPV1 به SNMPV2 و اخیراً SNMPV3 در مستند [RFC3410] انتشار یافته در دسامبر 2002 میلادی می‌باشد.

هنگامی که چارچوب مدیریت شبکه توصیف می‌شود برای برخی از پرسش‌ها بایستی پاسخ ارایه نمود. این پرسش‌ها عبارتند از:

- **چه مواردی نیازمند نظارت از نقطه نظر مفهوم می‌باشند؟ و چه قالبی از کنترل بایستی از سوی مدیر شبکه تعریف شود؟**
- **چه نوع اطلاعات ویژه‌ای بایستی پشتیبانی و مبادله گردد؟**
- **پروتکل ارتباط تبادل این اطلاعات چیست؟**

مثال سازمان مدیریتی که در ابتدای این فصل به آن اشاره گردید را در نظر بگیرید. مدیر اصلی شرکت و مدیران دفاتر شعبه نیازمند نوعی توافق از لحاظ کمیت فعالیت، کارآمدی و بودجه‌ی استفاده شده به‌منظور گزارش وضعیت دفاتر شعبه‌ها می‌باشند. آن‌ها همچنین نیازمند توافق برروی فعالیت‌های مدیر، به‌عنوان مثال کاهش بودجه، فرمان به مدیر شعبه برای تغییر برخی از عملیات شعبه یا اخراج کارکنان و تعطیلی برخی دفاتر شعبه می‌باشند. در پایین‌ترین سطح از این جزئیات آن‌ها نیازمند توافق بر روی قالبی هستند که این داده‌ها گزارش می‌شوند. به عنوان مثال در چه نوع مقیاس ارزی، دلار یا یورو بودجه بایستی گزارش گردد؟ در چه مقیاسی کارایی بایستی اندازه‌گیری شود؟ اگرچه این موضوعات ساده به نظر می‌آیند، اما بایستی در مورد آن‌ها توافق حاصل شود. سرانجام به عنوان نکته‌ی آخر در مثال شرکت، برای مبادله اطلاعات بین دفتر مرکزی و دفاتر شعبه بایستی روشی یا به بیان دیگر پروتکل ارتباطی مشخص گردد. چارچوب مدیریت استاندارد اینترنت همه‌ی پرسش‌های بالا را در چهار بخش ذیل پاسخ‌گو می‌باشد:

۱- **تعریف اهداف مدیریت شبکه،** اصطلاحاً با نام MIB شناخته می‌شود. در چارچوب مدیریت استاندارد اینترنت (ISMF)، اطلاعات به‌صورت مجموعه‌ای از اهداف مدیریت شده بوده که همراه یکدیگر تشکیل یک انبار اطلاعات مجازی را می‌دهند و به MIB یا مبانی مدیریت اطلاعات معروف می‌باشند. یک هدف یا عنصر MIB ممکن است یک شمارنده مانند شمارگانی از داده گرام‌های IP باشد که در یک مسیریاب به علت رخداد خطاهایی در سرپیام داده گرام IP حذف شده است و یا آنکه شمارگانی از خطاهای حسگر حامل در یک کارت واسط اترنت باشد. اطلاعات توصیفی عبارت است از: نسخه‌ی نرم‌افزار در حال اجرا بر روی یک سرور DNS، وضعیت اطلاعات یعنی آن‌که آیا یک وسیله عملکرد سالمی را دارد؟ و یا اطلاعات ویژه‌ی پروتکلی مانند مسیر به یک مقصد تعریف شده است یا خیر. عناصر یا اشیا MIB بنابراین به تعریف اطلاعات مدیریت که نگه‌داری شده از سوی یک وسیله‌ی مدیریت شده است، می‌پردازند. اشیا MIB وابسته در قالب **مدول‌های MIB** جمع‌آوری می‌شوند. در مثال شرکت اشاره شده، MIB اطلاعات انتقال یافته بین دفتر مرکزی و دفاتر شعبه می‌باشد.

۲- **زبان تعریف داده‌ها** اصطلاحاً به **ساختار اطلاعات مدیریت (SMI)**[1] معروف می‌باشد. معمولاً SMI نوع داده‌ها مانند یک شیء مدل و نیز قوانین نگارش و بازنگری اطلاعات مدیریت را تعریف می‌کند. اشیاء MIB در این زبان تعریف داده‌ها، مشخص می‌شوند. در مثال اشاره شده در ابتدای فصل SMI به‌منظور تعریف جزئیات قالب اطلاعات مبادله شده مورد استفاده قرار می‌گیرد.

۳- **پروتکل SNMP**، پروتکل SNMP به‌منظور انتقال اطلاعات و فرمان‌ها بین یک موجودیت مدیر و یک مجری از سوی آن موجودیت در درون یک وسیله‌ی مدیریت شده‌ی شبکه مورد بهره‌برداری قرار می‌گیرد.

۴- **توان‌مندی‌های مدیریت و امنیت**. افزودن این توان‌مندی‌ها بهینه‌سازی مهمی را در SNMPV3 در مقابل SNMPV2 سبب شده است.

معماری مدیریت شبکه‌ی اینترنت یک طراحی مدولار است همراه با یک پروتکل مستقل زبان تعریف داده‌ها و نیز MIB که حالت یک پروتکل مستقل را دارد. جالب توجه آن است که این معماری مدولار، نخستین بار جهت سهولت در انتقال از یک SNMP برمبنای مدیریت شبکه به یک چارچوب مدیریت شبکه (ISMF) از سوی ISO تعریف گردید. درگذر زمان طراحی مدولار SNMP اجازه‌ی تکامل سه نسخه‌ی مهم آن را سبب شده است که هر کدام شامل چهار بخش مهم از موارد تعریف شده‌ی SNMP می‌باشند، همه‌ی آن‌ها به‌گونه‌ای مستقل هستند، در ادامه چهار عنصر اصلی و مهم ISMF با جزئیات بیش‌تر تحلیل می‌شوند.

۹-۵ ساختار مدیریت اطلاعات (SIM)

ساختار مدیریت اطلاعات (SMI) به‌عنوان بخشی از ISMF، زبان توصیفی است که از آن به‌منظور تعریف اطلاعات مدیریت ساکن در یک شیء مدیریت شده استفاده می‌شود. چنین زبان توصیفی به‌منظور اطمینان از واضح شدن قوانین و مفاهیم داده‌های مدیریت شبکه، طراحی شده است. بدیهی است SMI به تعریف یک گونه داده‌ی ویژه در یک شیء مدیریت شبکه نمی‌پردازد، بلکه زبان تعریف مشخصات این گونه اطلاعات به‌شمار می‌آید. اسنادی که SMI را برای پروتکل SNMPV3 ویا چنانچه در برخی از انتشارات SMIV2 نامیده می‌شود، تعریف می‌کنند، عبارتند از:

[RFC2578, RFC2579, RFC2580]. در ادامه‌ی مطالب این بخش، بررسی SMI با رویکرد از فرود به فراز آن انجام می‌گیرد. آغاز آن انواع داده‌های مبنا در SMI می‌باشد و پس از آن اشیاء مدیریت شده‌ی SMI و آن‌گاه ارتباط اشیاء مدیریت شده در قالب گروه‌های مدل تحلیل می‌شوند.

• انواع داده‌های مبنا در SMI

مستند [RFC2578] به تعریف انواع داده‌های مبنا در زبان توصیف مدول SMI-MIB می‌پردازد. زبان SMI خود برمبنای **ASN.1**[2] می‌باشد که زبانی است شیءگرا و در مستند [ISOX.680] توصیف شده است. آن‌چه برای SMI تعریف گردیده آن را تبدیل به یک زبان توصیف داده نموده است. تعداد 11 گونه داده در مستند [RFC2578] تعریف شده که جدول ۹-۱ آن‌ها را نشان می‌دهد. علاوه بر این اشیاء

1- Structure of management information

2- Abstract syntax notation one

اسکالر، امکان تحمیل یک ساختار جدول برای مجموعه‌ی به نظم درآمده‌ای از اشیاء MIB با بهره‌گیری از ساختار **SEQUENCEOF** را نیز امکان‌پذیر می‌کند. مستند [RFC2578] جزئیات آن را توصیف نموده است. اغلب گونه‌های داده در جدول ۹-۱ برای خوانندگان واضح می باشند. تنها نوع داده‌ای که با جزئیات بیش‌تر تحلیل خواهد شد. نوع **OBJECT IDENTIFIER** می‌باشد، که از آن با هدف نام‌گذاری یک شیء استفاده می‌شود.

جدول ۹-۱ ♦ انواع داده‌های مبنا در SMI

نوع داده	توصیف
IDENTIFIER	ارزش کامل 32 بیت برمبنای تعریف ASN.1 همراه با ارزشی بین 2^{31}-1 ~2^{31}- می‌باشد. همچنین ممکن است، ارزشی از فهرست نامگذاری شده‌ی ارزش‌های ثابت باشد.
Integer32	ارزش کامل 32 بیت بین 2^{31}- و 2^{31}-1 می‌باشد.
Unsigned32	ارزش کامل 32 بیت بدون علامت مابین محدوده‌ی صفر تا 2^{31}-1
OBJECTSTRING	قالب رشته بایت ASN.1 که نماد داده‌ی باینری یا متنی تصادفی با حداکثر طول 65535 بایت می‌باشد.
OBJECT IDENTIFIER	نام ساخت یافته برمبنای قالب مدیریت ASN.1
IP address	آدرس اینترنت 32 بیت در ترتیب بایت شبکه
Counter32	شمارشگر 32 بیت که افزایش آن از صفر تا 2^{32}-1 می باشد و سپس بازگشت به صفر می‌کند.
Counter64	شمارشگر 64 بیت
Gauge32	ارزش کامل 32 بیت که شمارش بالاتر از 2^{32}-1 را انجام نمی‌دهد و از صفر هم در هنگام افزایش یا کاهش پایین‌تر نمی‌آید.
Timer Ticks	اندازه‌گیر زمانی به‌صورت 1 بر یک صدم یک ثانیه از وقوع یک رخداد
Opaque	رشته‌ی تفسیر نشده‌ی ASN.1 مورد نیاز مقایسه‌ی سازگاری عقب گردد

- **سازه‌های سطح بالاتر SMI**

علاوه بر انواع داده‌های مبنا زبان توصیف داده‌های SMI سازه‌های سطح بالا را نیز تأمین می‌نماید. سازه‌ی OBJECT-TYPE به‌منظور تعریف نوع داده، وضعیت یا حالت، و مفاهیم شیء مدیریت شده می‌باشد. به‌صورت جمعی این اشیاء مدیریت شده حامل داده‌های مدیریت در هسته‌ی مدیریت شبکه می باشند. تعداد آن‌ها به بیش از 10000 شیء تعریف شده در مستند اینترنت [RFC3410] می‌رسد. سازه‌ی **OBJECT-TYPE** دارای چهار عنوان توصیفی است. این موارد عبارتند از:

- **عنوان SYNTAX** که مشخصات نوع داده‌ی مبنا وابسته به شیء را تعریف می‌کند.
- **عنوان MAX-ACCESS** این موضوع را آشکار می‌کند که آیا شیء مدیریت شده قابلیت خواندن، نوشتن و تعریف را داردیا اختصاص ارزش به آن در یک توصیف امکان‌پذیر است.
- **عنوان STATUS** به تعریف این موضوع می‌پردازد که شیء تعریف شده جاری، معتبر و یا

از اعتبار آن گذشته است، که در این صورت پیاده‌سازی نمی‌شود و فقط پیشینه‌ی آن ذخیره می‌گردد و یا آنکه منسوخ شده است اما قابل پیاده‌سازی به‌منظور استفاده در پیاده‌سازی‌های قدیمی‌تر می‌باشد.

- **عنوان DESCRIPTION** که حامل توصیف اشیاء در قالب متن قابل فهم برای انسان می‌باشد. این عنوان با هدف مستندسازی اشیاء مدیریت شده بوده و بایستی همه‌ی اطلاعات مفهومی موردنیاز به‌منظور پیاده‌سازی یک شیء را تأمین نماید.

به عنوان مثال سازه‌ی OBJECT-TYPE را درنظر بگیرید که در آن نوع شیء برمبنای مستند [RFC4293] به صورت ذیل می‌باشد:

ip System-statsInDelivers

این شیء به تعریف یک شمارشگر 32 بیت به‌منظور رهگیری تعداد داده گرام‌های دریافت شده از سوی یک وسیله‌ی مدیریت شده می‌پردازد. به‌طوریکه همه‌ی آن‌ها با موفقیت به یک پروتکل لایه‌ی بالاتر انتقال یافته‌اند. آخرین بخش این تعریف نام این شیء می‌باشد که در بخش بعدی تحلیل می‌شود. در ذیل قطعه ساختمان داده‌ی شمارشگر نشان داده شده است:

```
ipSystemStatsInDelivers OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "The total number of datagrams successfully
          delivered to Ipuser-protocols (including ICMP).

          When tracking interface statics, the counter of the
          interface to which these datagrams were addressed is
          incremented.This interface for some of the datagrams.

          Discontinuities in the value of this counter can
          occur at re-initialzation of the management system,
          and at other times as indicated by the value of
          ipSystemStatsDiscontinuityTime."
    ::={ipSystemStatsEntry  18}
```

سازه‌ی **MODULE-IDENTITY** اجازه‌ی ارتباط اشیا را در قالب گروهی در یک مدول امکان‌پذیر می‌سازد. به عنوان مثال مستند [RFC4293] مدول MIB را مشخص می‌کند. اشیاء مدیریت شده شامل (ip System StatsInDelivers)، برای پیاده‌سازی‌های IP و پروتکل وابسته‌ی آن ICMP را مدیریت می‌کند. مستند [RFC4022] مدول MIB را برای TCP تعریف و مستند [RFC4133] مدول MIB را برای UDP و نیز مستند [RFC4502] مدول MIB را به‌منظور نظارت راه دور RMON تعریف می‌کند. علاوه بر حمل تعاریف OBJECT-TYPE از اشیاء مدیریت شده‌ی درون مدول، ساختمان داده‌ی MODULE-IDENTITY شامل عناوین اطلاعات نویسنده‌ی مدول، تاریخ و زمان و غیره می‌باشد. به عنوان مثال به تعریف مدول توصیف مدیریت پروتکل IP در ذیل دقت نمایید:

```
ipMTB MODULE-IDENTITY
    LAST-UPDATED "200602020000Z"
    ORGANIZATION "IETF IPv6 MIP Revision Team"
    CONTACT-INFO
            "Editor:
            Shown A. Routhier
            Interworking Labs
            108 Whispering Pines Dr. Suite 235
            Scotts Valley, CA 95066
            USA
            Email: <sar@iwl.com
    DESCRIPTION
            "The MIB module for managing IP and ICMP
            implementations, but excluding their
            management of IP routes.

            Copyright (C) The Internet Society (2006).
            This version of this MIB module is part of
            RFC 4293; see the RFC itself for full legal
            notices."
    REVISION        "20060202000002"
    DESCRIPTION
            "The IP version neutral revision with added
            IPv6 objects for ND, default routers, and
            router advertisements. As Well as being the
            successor to RFC 2011, this MIB is also the
            successor to RFCs 2465 and 2466. Published
            as RFC 4293."
    REVISION        "1994110100002"
    DESCRIPTION
            "A separate MIB module (IP-MIB) for IP and
            ICMP management objects. Published as RFC
            2011."
    REVISION        "1991033100002"
    DESCRIPTION
            "The initial revision of this MIB module was
            part of MIB-II, which was published as RFC
            1213."
    ::= { mib-2  48}
```

از ساختمان داده‌ی **NOTIFICATION-TYPE** به‌منظور توصیف اطلاعات SNMPV2-Trap و پیام‌های Information Request تولید شده از سوی یک عامل یا مدیر استفاده می‌شود. مستند [RFC2578] جزئیات این موضوع را توصیف نموده است. ساختمان داده‌ی: **MODULE-COMPLIANCE** مجموعه‌ای از اشیا مدیریت شده در یک مدول بوده که عامل آن را پیاده‌سازی می‌نماید. سرانجام ساختمان داده‌ای مانند **AGENT-CAPABILITIES** توان‌مندی‌های عامل‌ها را با توجه به شیء و آگهی رخدادها تعریف می‌کند.

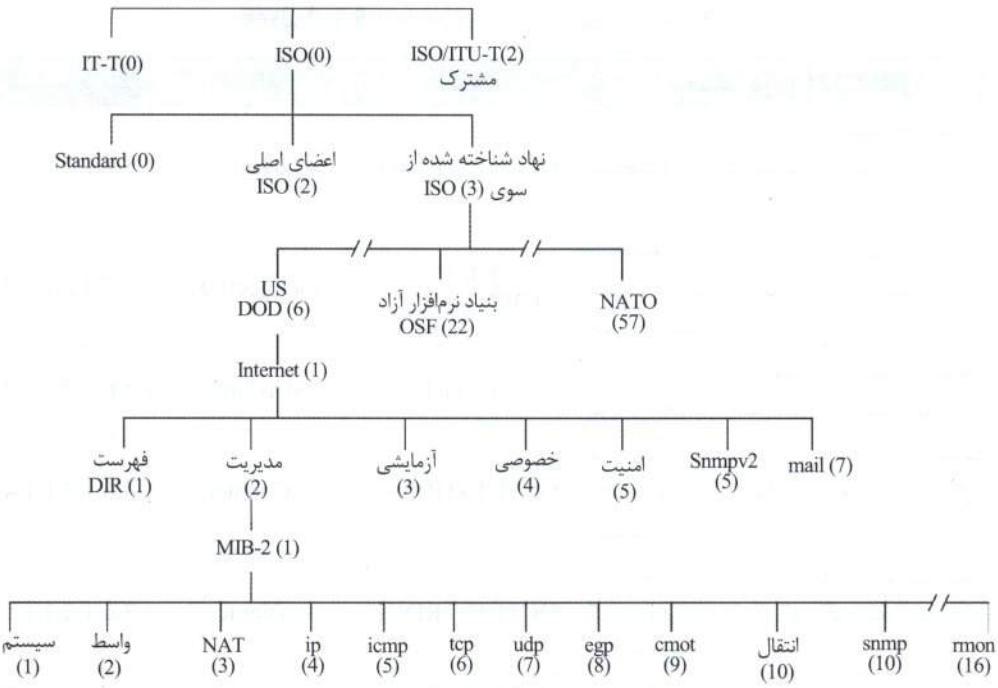
۹-۶ مبنای مدیریت اطلاعات (MIB)

همان‌گونه که در بالا ملاحظه شد مبنای مدیریت اطلاعات (MIB) را می‌توان یک انبار ذخیره‌ی اطلاعات مجازی تصور نمود که نگه‌داری اشیاء مدیریت شده‌ای که ارزش جمعی آن‌ها «حالت» جاری شبکه است را برعهده می‌گیرد. این ارزش‌ها ممکن است مورد پرسش قرار گرفته و یا با یک موجودیت مدیر از طریق ارسال پیام‌های SNMP به عامل مجری یک وسیله‌ی مدیریت شده تنظیم و برنامه‌ریزی گردند. اشیاء مدیریت شده از طریق ساختمان داده‌ی OBJECT-TYPESMI تعریف می‌شوند و جمع‌آوری آن‌ها در درون **مدول‌های MIB** با بهره‌گیری از ساختمان داده‌ی **MODULE-IDENTITY** انجام می‌شود.

نهاد IETF مدول‌های MIB را به‌منظور کاربرد در روترها، میزبان‌ها و دیگر تجهیزات شبکه، استاندارد نموده است. این موارد شامل مبانی شناسه‌ی داده‌ها در یک قطعه‌ی سخت‌افزاری و نیز اطلاعات مدیریت پروتکل‌ها و واسط‌های شبکه می‌شوند. تا سال 2006 میلادی بیش از 200 استاندارد بر پایه‌ی مدول‌های MIB تعریف شده‌اند و تعداد به مراتب بیش‌تری از آن‌ها از سوی شرکت‌های تولید کننده‌ی خصوصی MIB صورت گرفته است. با همه‌ی این استانداردها سازمان IETF نیازمند روشی به‌منظور شناسایی و نام‌گذاری مدول‌های استاندارد شده و اشیاء مدیریت شده در درون یک مدول می‌باشد. به جای آن‌که از نقطه‌ی صفر این عمل صورت پذیرد، IETF شناسایی شیء استاندارد شده‌ای را معیار چارچوب علمی خود قرار داده که از سوی ISO تعریف شده است. هدف ISO شناسایی هر شیء استاندارد شده مانند قالب داده‌ها، پروتکل‌ها و یا قطعه‌های اطلاعاتی در هر شبکه بدون توجه به استانداردهای پیاده‌سازی شبکه‌ها مانند IFTF، ISO، IEEE و یا ANSI می‌باشد. در این رابطه حتی توجهی به تولیدکننده‌های تجهیزات شبکه و مالکین شبکه‌ها نشده است. چارچوب شناسایی اشیاء که از سوی ISO گزینش شده بخشی از ASN.1 به‌شمار می‌آید.

همان‌گونه که در شکل ۹-۳ نشان داده شده است، اشیاء در چارچوب نام‌گذاری ISO تعریف می‌شوند که قالب سلسله مراتبی دارد. بدیهی است هر شاخه در درخت تعریف شده دارای دو عنصر **نام و شماره** می‌باشد. شماره‌ها در ( ) نشان داده شده‌اند. بنابراین هر نقطه‌ای از درخت از طریق ردیفی از نام‌ها و یا اعداد که مسیر آن‌ها را از ریشه‌ی درخت به آن نقطه از درخت شناسه هدایت می‌کند، تعریف می‌شود. برفراز سلسله مراتب اشاره شده ISO و ITU-T دو نهاد مهم بین‌المللی استاندارد ASN.1 قرار گرفته‌اند. در زیرشاخه‌ی درخت ISO همه‌ی استانداردهای ISO با شماره‌های آن‌ها دیده می‌شوند. این درخت شناسه نهادهای دیگری را نیز شامل می‌شود که به علت صرفه‌جویی در فضا از اشاره‌ی به آن‌ها خودداری شده است.

در زیرشاخه‌ی **Internet** یعنی (1.3.6.1) تعداد 7 گروه وجود دارد و در زیرشاخه‌ی **private** یعنی (1.3.6.1.4) فهرستی از اسامی و کدهای شرکت‌های خصوصی و هزاران شرکت ثبت شده با IANA موجود می‌باشد. در زیرشاخه‌ی **management** یعنی (1.3.6.1.2) و شاخه‌های MIB-2 یعنی (1.3.6.1.2.1) از درخت شناسه‌ی اشیاء تعاریف مدول‌های استاندارد MIB را می‌توان شناسایی نمود.



شکل ۹-۳ ♦ درخت شناسه‌ی شیء ASN.1

- **مدول‌های استاندارد MIB**

پایین‌ترین سطح درخت در شکل ۹-۳ برخی از مدول‌های مهم سخت‌افزاری MIB با عنوان: (System and inferfaces) و نیز مدول‌های وابسته‌ی برخی از پروتکل‌های مهم اینترنت را نشان می‌دهد. مستند [RFC5000] فهرستی از همه‌ی مدول‌های استاندارد شده‌ی MIB را تعریف می‌کند. جدول ۹-۲ اشیاء یک گروه را در یک سیستم بر مبنای تعریف آن‌ها در مستند [RFC1213] نشان می‌دهد و جدول ۹-۳ نیز اشیاء مدیریت شده‌ی مدول MIB برای پروتکل UDP را در یک موجودیت مدیریت شده تعریف می‌کند.

۹-۷ عملیات پروتکل SNMP و نگاشت‌های انتقال

نسخه‌ی دوم پروتکل SNMPV2 توصیف شده در مستند [RFC3416] با هدف انتقال اطلاعات MIB بین موجودیت‌های مدیریت شده و عامل‌های مجری به نمایندگی از سوی مدیران اصلی مورد استفاده قرار می‌گیرد. کاربرد متداول SNMP در **روش تقاضا و پاسخ** است، که در آن یک عنصر مدیریت SNMPV2 درخواستی را به یک عامل SNMPV2 ارسال می‌دارد و او پس از دریافت اقلام به انجام عملیات و ارسال پاسخ مناسب آن تقاضا، اقدام می‌نماید. یک تقاضا معمولاً برای بازیابی یا اعمال تغییر ارزش‌های عنصر MIB وابسته به یک وسیله‌ی مدیریت شده است.

جدول ۹-۲ ♦ اشیاء مدیریت شده در گروه سیستم MIB-2

شناسه‌ی شیء	نام	نوع	توصیف در [RFC1213]
1.3.6.1.2.1.1.1	sysDescr	OCTETSTRING	شناسه‌ی کامل نام و نسخه‌ی نوع سخت‌افزار سیستم، سیستم عامل و نرم‌افزار شبکه
1.3.6.1.2.1.1.2	sysObjectID	OBJECT IDENTIFIER	شناسه‌ی ID اختصاص یافته از سوی سازنده. تأمین امکانات لازم و آشکار نوع جعبه‌ی مدیریت شده
1.3.6.1.2.1.1.3	sysUpTime	Time Ticks	زمان در قالب صدم ثانیه از لحظه‌ی آخرین بازیابی بخشی از سیستم مدیریت شده.
1.3.6.1.2.1.1.4	sysContact	OCTETSTRING	فرد مورد تماس برای این گره‌ی مدیریت شده به‌همراه اطلاعات مربوط به چگونگی ارتباط با وی
1.3.6.1.2.1.1.5	sysName	OCTETSTRING	نام اختصاص یافته از سوی مدیر به این گره‌ی مدیریت شده. این گره بنابر قرارداد نام محدوده‌ی کاملاً معتبر می‌باشد.
1.3.6.1.2.1.1.6	sysLocation	OCTETSTRING	آدرس فیزیکی این گره
1.3.6.1.2.1.1.7	sysServices	Integer32	یک ارزش کد شده برای بیان مجموعه‌ای از خدمات در دسترس در این گره فیزیکی به مفهوم اپراتور، پیوند داده یا زیرشبکه مانند مسیریاب دریچه‌ی ID و انتها به انتها مانند میزبان‌ها و سرانجام کاربردها.

جدول ۹-۳ ♦ اشیا مدیریت شده‌ی گزینشی در مدول UDP وابسته به MID-2

شناسه‌ی شیء	نام	نوع	توصیف برمبنای [RFC4113]
1.3.6.1.2.1.7.1	udpInDatagram	Counter32	مجموع تعداد داده گرام‌های UDP تحویل شده به کاربران
1.3.6.1.2.1.7.2	udpNoPorts	Counter32	مجموع تعداد داده گرام‌های UDP دریافتی که برای آن‌ها هیچ‌گونه کاربردی در پورت مقصد وجود ندارد
1.3.6.1.2.1.7.3	udpInErrors	Counter32	مجموع تعداد داده گرام‌های UDP تحویل نشده به دلایلی به غیر از عدم وجود کاربرد در پورت مقصد
1.3.6.1.2.1.7.4	udpOutDatagrams	Counter32	مجموع تعداد داده گرام‌های UDP ارسالی از این موجودیت

دومین کاربرد متداول SNMP برای یک عامل ارسال پیام‌های بدون درخواست قبلی به مدیر است که اصطلاحاً به آن‌ها **پیام‌های دام (TM)**[1] گفته می‌شود. از پیام‌های TM به‌منظور آگاهی مدیر اصلی از یک حالت اضطراری که سبب ایجاد تغییرات در ارزش عنصر MIB شده است، استفاده می‌شود. قبلاً در بخش ۹-۱ ملاحظه شد که مدیر شبکه ممکن است مایل به دریافت یک پیام TM هنگامی که یک کارت شبکه معیوب شده و یا زمانی‌که تراکم ترافیک در یک پیوند به آستانه‌ی از پیش تعریف شده‌ی خود رسیده و یا دیگر رخدادها باشد. لازم به توضیح است که تفاوت‌های مهمی بین پیام‌های TM و تعامل‌های به‌صورت پرسش و پاسخ وجود دارد. پروتکل SNMPV2 هفت نوع پیام به صورت ژنریک را به عنوان پروتکل اجزا داده **(PDU)**[2] تعریف می‌کند. جدول ۹-۴ به ترتیب توصیف این PDUها و قالب آن‌ها را نشان می‌دهد.

- **اجزای داده‌ی GetRequest، GetNextRequest ,Get BulkRequest**. این اجزا داده همگی از سوی یک موجودیت مدیر به عامل مدیر ارسال می‌شوند تا تقاضای یک یا چندین ارزش اشیاء MIB در وسیله‌ی مدیریت شده‌ی عامل را نمایند. شناسه‌گرهای شیء MIB عبارتست از اشیایی که ارزش‌های آن مشخص و درخواست آن‌ها در قالب یک بخش متغیر PDU می‌باشد. سه عنصر GetRequest، GetNextRequest و GetBulkRequest از نقطه نظر مقیاس‌گذاری و درجه‌بندی در تقاضای داده‌ها گوناگون می باشند. عنصر GetRequest تقاضای مجموعه ارزش‌های دلخواه از MIB را می‌نماید و نیز در قالب چندین تقاضا می‌تواند ردیفی را از درون یک فهرست یا جدول از اشیاء MIB درخواست نماید. از سوی دیگر عنصر GetBulkRequest اجازه‌ی بازگشت توده‌ای از داده‌ها را می‌دهد و نتیجه‌ی آن باعث جلوگیری از وقوع سرآیند درصورت ارسال چندین پیام GetRequest و GetNextRequest می باشد. در هریک از سه مورد اشاره شده واکنش عامل یک ResponsePDU خواهد بود که حامل شناسه‌گرهای شیء و ارزش‌های وابسته به آن‌ها می‌باشد.
- **اجزای داده‌ی SetRequestPDU**. از این جز داده به‌منظور تنظیم مجموعه‌ی ارزش‌ها از یک یا چندین شیء MIB در یک وسیله‌ی مدیریت شده از سوی موجودیت مدیر استفاده می‌شود. یک عامل، پاسخ لازم را در قالب جزء ResoibsePDU با وضعیت "noerror" ارسال می‌کند تا انجام تنظیم مجموعه را به ارزش وابسته‌ی آن تأیید نماید.
- **اجزای داده‌ی InformRequestPDU**. این عنصر از سوی یک موجودیت مدیر به‌منظور آگهی به مدیر دیگر در مورد اطلاعات MIB که وی در حال دریافت آن‌ها می‌باشد، بهره‌برداری می‌شود. موجودیت دریافت کننده پاسخ این آگهی را با یک عنصر داده‌ی به‌صورت InformRequestPDU تأیید می‌نماید.
- آخرین نوع جزء داده‌های PDU عبارتست از SNMPV2PDU که یک پیام داده (TM) می‌باشد. معمولاً TMها همزمان تولید می‌شوند. به ا ین مفهوم که تولید در اثر واکنش به یک تقاضای دریافت شده نمی‌باشد بلکه در واکنش به یک رخداد است که موجودیت مدیر نیازمند آگهی آن‌ها می‌باشد. مستند [RFC3418] انواع معروف و متداول TMها را توصیف نموده است. از جمله‌ی آن‌ها می‌توان از **شروع سرد و یا گرم (فعال و غیرفعال) یک**

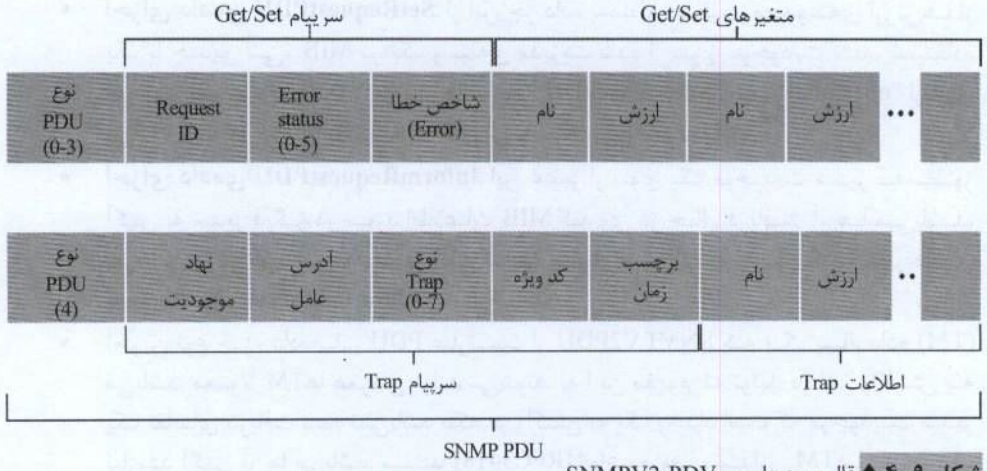
1- Trap message

2- Protocol data unit

وسیله، ورود و خروج یک پیوند به مدار، خروج مجاور از مدار و یا عدم تأیید اعتبارسنجی نام برد. یک TM دریافتی هیچ نوع واکنشی را از سوی موجودیت مدیر تقاضا نمی‌کند.

جدول ۹-۴ ♦ انواع اجزای داده‌ی **SNMP-PDU**

نوع SNMPV2 PDU	فرستنده – گیرنده	توصیف
GetRequest	مدیر به عامل مدیریت	دریافت ارزش یک یا چندین شیء MIB در لحظه‌های گوناگون
GetNextRequest	مدیر به عامل مدیریت	دریافت ارزش بعدی شیء MIB در یک فهرست یا جدول
GetBulkRequest	مدیر به عامل مدیریت	دریافت ارزش‌های یک توده‌ی بزرگ داده مانند ارزش‌های درون یک جدول
InformRequest	مدیر به عامل مدیریت	موجودیت مدیر راه دور را ارزش‌های MIB دور از دسترس آن آگاه می‌کند
SetRequest	مدیر به عامل مدیریت	تنظیم ارزش یک یا چندین شیء MIB
Response	عامل به مدیر یا مدیر به مدیر	GetRequest GetNextRequest GetBulkRequest SetRequest PDU InformRequest
SNMPV2-Trap	عامل به مدیر	آگهی به مدیر از وقوع یک رخداد غیرعادی



شکل ۹-۴ ♦ قالب جز داده‌ی SNMPV2-PDV

با توجه به طبیعت تقاضا و پاسخ پروتکل SNMPV2 یادآوری این نکته ضروری است، اگرچه اجزای داده از نوع SNMP PDU از طریق تعداد انبوهی از پروتکل‌های مختلف انتقال، امکان حمل را دارند، اما SNMP PDU به‌صورت ویژه در یک محموله از یک داده گرام UDP حمل می‌شود. مستند

[RFC3417] براین موضوع تأکید دارد که UDP پروتکل انتقال ترجیحی نگاشت PDU می‌باشد، زیرا UDP یک پروتکل انتقال نامطمئن است که ضمانت دریافت تقاضا و یا پاسخ را از سوی گیرنده برعهده نمی‌گیرد. دامنه‌ی ID برای یک تقاضا در PDU از سوی مدیر به‌منظور شماره‌گذاری تقاضاهای آن مدیر به عامل مورد استفاده قرار می‌گیرد. پاسخ یک عامل، شناسه‌ی یک درخواست را از دامنه‌ی ID یک تقاضا دریافت می‌دارد. بنابراین مدیر از دامنه‌ی ID به‌منظور تشخیص مفقودی تقاضاها و یا پاسخ‌ها استفاده می‌کند. این مهم برعهده‌ی موجودیت مدیر است که در مورد ارسال مجدد یک تقاضا درصورت عدم پاسخ آن پس از گذشت یک زمان معین تصمیم‌گیری نماید. به ویژه استاندارد SNMP اجباری را از نقطه نظر انتقال مجدد تقاضا از سوی مدیر مشخص نکرده است. آن‌چه این استاندارد مورد تأکید قرار داده، واکنش مسئولیت‌پذیر در قبال تعداد مرتبه‌ای است که مدیر می‌تواند تکرار تقاضای خود را به عامل آگهی کند. بدیهی است مسئولیت‌پذیری پروتکل خود موضوع بحث برانگیز و جالبی می‌باشد.

۹-۸ مدیریت و امنیت

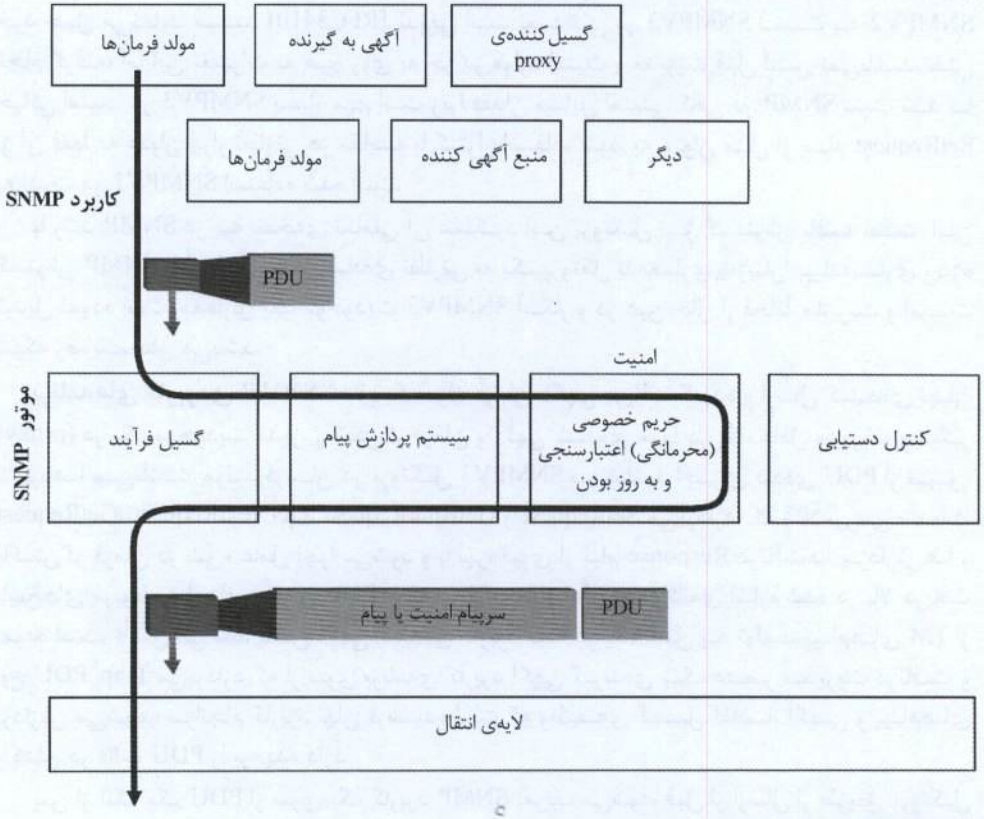
طراحان پروتکل SNMPV3 براین اعتقاد می‌باشند که پروتکل SNMPV3 را می‌توان همان پروتکل SNMPV2 تصور نمود، با این تفاوت که توان‌مندی‌های افزون‌تری از لحاظ مدیریت و امنیت را در درون خود حمل می‌نماید. مستند [RFC3410]. بدیهی است تغییراتی در SNMPV3 نسبت به SNMPV2 انجام گرفته، اما این تغییرات به هیچ روی به جز در موارد امنیت و مدیریت قابل لمس نمی‌باشند. نقش حیاتی امنیت در SNMPV3 بسیار مهم است زیرا فقدان مسایل امنیتی کافی در SNMP سبب شد تا از آن فقط به عنوان ابزار نظارتی در مقایسه با کنترل استفاده شود. به عنوان مثال از پیام SetRequest به ندرت در SNMPV1 استفاده شده است.

با رشد SNMP در سه نسخه‌ی تکاملی آن عملکرد این پروتکل نیز گسترش یافته است. این گسترش SNMP را از یک پروتکل ساده‌ی نظارتی به یک پروتکل با معماری و زبان پیاده‌سازی ویژه تبدیل نموده است. معماری یک موجودیت SNMPV3 آشکار و در عین حال از لحاظ مدیریت و امنیت شبکه رضایت‌بخش می‌باشد.

برنامه‌های کاربردی SNMP شامل یک مولد فرمان، آگهی دریافت کننده و ارسال کننده‌ی نهان (proxy) در یک موجودیت مدیر، واکنش‌گر فرمان و آگهی کننده‌ی مبدأ در یک عامل مدیریت و دیگر کاربردها می‌باشد. مولد فرمان در پروتکل SNMPV3 به تولید اجزای داده‌ی PDU از قبیل: GetRequest، GetNextRequest، GetBulkRequest و SetRequest می‌پردازد، که قبلاً بررسی شده‌اند. واکنش‌گر فرمان در شیء عامل اجرا می‌شود و با بهره‌گیری از پیام Response دریافت‌ها، پردازش‌ها و پاسخ‌های مربوط به اجزای داده‌ی PDU را که در قالب تقاضاهای چهارگانه‌ی اشاره شده در بالا دریافت نموده است، تأمین می‌کند. بخش آگهی کننده‌ی کاربرد مبدأ در یک عامل به تولید پیام‌های TM از نوع Trap PDU می‌پردازد که از سوی برنامه‌ی کاربرد آگهی گیرنده‌ی یک عنصر مدیریت دریافت و پردازش می‌شوند. سرانجام کاربرد نهان فرستنده است که وظیفه‌ی گسیل تقاضا، آگهی و پیام‌های واکنش در قالب PDU را برعهده دارد.

پس از آنکه یک PDU از سوی یک کاربرد SNMP تعریف می‌شود، قبل از ارسال از طریق پروتکل انتقال مناسب آن از درون یک **موتور** عبور می‌نماید. شکل ۹-۵ چگونگی تولید یک PDU از سوی مولد فرمان کاربرد را نشان می دهد که ابتدا وارد مدول ارسال می‌شود و در آن‌جا نسخه‌ی SNMP مشخص می‌گردد. پس از آن PDU در سیستم پردازش پیام تحلیل می‌شود و در این نقطه است که سرپیام

حامل شماره‌ی نسخه‌ی SNMP، ID پیام و اندازه یا طول اطلاعات در پیام به آن افزوده می‌شود. در صورت نیاز به رمزگذاری و یا اعتبارسنجی، دامنه‌های مناسب آن‌ها نیز برمبنای مستند [RFC3411] اضافه می‌گردند. سرانجام پیام SNMP که از این مجموعه تشکیل شده، به پروتکل انتقال مناسب آن گسیل می‌شود. پروتکل ترجیحی انتقال پیام‌های SNMP، پروتکل نامطمئن UDP است. شماره پورت ترجیحی پیام‌های SNMP نیز گذرگاه 161 می‌باشد. از پورت 162 نیز برای پیام‌های TM استفاده می‌شود. قبلاً ملاحظه شد پیام‌های SNMP فقط به‌منظور نظارت شبکه مورد بهره‌برداری قرار نمی‌گیرند، بلکه کنترل عناصر شبکه را از طریق فرمانی مانند SetRequest نیز برعهده دارند. بدیهی است یک مهاجم قادر به شنود پیام‌های SNMP و تولید بسته‌های SNMP خود در زیرساخت مدیریت که نتیجه‌ی آن تخریب و فروپاشی شبکه است، نیز می‌باشد. بنابراین بسیار اهمیت دارد که پیام‌های SNMP با امنیت کامل منتقل شوند. ویژگی امنیت SNMPV3 معروف به **امنیت بر مبنای کاربر (USB)**[1] می‌باشد و در مستند [RFC3414] توصیف شده است. این عنوان امنیتی به مفهوم آن است که یک کاربر از طریق شناسه‌ی کاربری شناخته می‌شود که شامل یک رمز عبور، ارزش کلید و درجه‌ی دسترسی وی می‌باشد. پروتکل SNMPV3 عوامل رمزگذاری، اعتبارسنجی و حفاظت در برابر تهاجم بازپخش اطلاعات و کنترل دسترسی را نیز تأمین می‌نماید.



شکل ۹-۵ موتور SNMPV3 و کاربردهای آن

1- User base security

- **رمزگذاری SNMPV3**. پیام‌های PDU در SNMP با بهره‌گیری از استاندارد رمزگذاری داده (DES) و نیز بهره‌گیری از روش توده زنجیره‌ی رمز (CBC) رمزگذاری می‌شوند. آشکار است از آن‌جایی که DES یک سیستم کلید مشترک می باشد، کلید رمز کاربر رمزکننده‌ی داده بایستی از سوی موجودیت دریافت کننده که داده‌ها را رمز می‌کند، شناخته شده باشد.
- **اعتبارسنجی در SNMPV3**. پروتکل SNMP با بهره‌گیری از روش رمز اعتبارسنجی پیام (MAC) هر دو موضوع اعتبارسنجی و جعل پیام را براساس مستند [RFC2401] تأمین می‌کند. یادآوری می‌شود MAC نیازمند آن است که فرستنده و گیرنده هر دو از کلید رمز مشترک آگاه باشند.
- **محافظت در برابر بازپخش پیام در SNMPV3**. در فصل ۸ ملاحظه شد از پیام‌های موقت (nonce) به‌منظور حفاظت در برابر تهاجم بازپخش پیام استفاده می‌شود. پروتکل SNMPV3 نیز از همین خاصیت استفاده می‌کند. به‌منظور اطمینان از این موضوع که یک پیام دریافتی بازپخش پیام قبلاً ارسال شده‌ای نمی‌باشد. گیرنده نیازمند آن است که فرستنده ارزشی را برای هر پیام برمبنای یک شمارشگر در گیرنده تعریف نماید. این شمارشگر که عمل nonce را انجام می‌دهد، منعکس کننده‌ی مقدار زمانی است که از آخرین راه‌اندازی مجدد نرم‌افزار مدیریت شبکه در گیرنده گذشته است. علاوه بر آن مجموع تعداد راه‌اندازی‌های مجدد از آخرین مرتبه‌ای که پیکربندی نرم‌افزار مدیریت شبکه انجام شده را نیز شامل می شود. تا هنگامی که شمارشگر یک پیام دریافتی، درحد قابل قبولی از خطای تعریف شده برای گیرنده می‌باشد، پیام دریافت شده به عنوان پیام بدون حالت بازپخش قلمداد می‌شود و بنابراین بایستی مورد اعتبارسنجی و یا رمزگشایی قرار گیرد. مستند [RFC3414] جزئیات این موضوع را تشریح نموده است.
- **کنترل دستیابی SNMPV3**. پروتکل SNMPV3 کنترل دسترسی بر مبنای دیداری را بر طبق مستند [RFC3415] تأمین می‌کند. این مهم از طریق کنترل اطلاعات مدیریت شبکه که از سوی کاربران می‌تواند مورد تقاضا و یا تنظیم قرار گیرد، انجام می‌شود. یک موجودیت SNMP به نگه‌داری اطلاعات در مورد حقوق دسترسی و سیاست‌گذاری‌های **پیکربندی پایگاه داده‌های محلی (LCD)**[1] می‌پردازد. بخشی از LCD از سوی اشیاء مدیریت شده قابل دستیابی است که در یک پروتکل **الگوی پیکربندی کنترل دستیابی MIB برمبنای دیداری (VACMC)**[2] و نیز مستند [RFC3415] تعریف می‌گردد. این امکان، مدیریت و کنترل شبکه را به صورت راه دور از طریق پروتکل SNMP فراهم می‌آورد.

۹-۹ پروتکل ASN.1

در این اثر عناوین متعدد و جالبی را در ارتباط با شبکه‌های کامپیوتری مطالعه نموده‌ایم. این بخش از مباحث یعنی ASN.1 ممکن است در فهرست 10 عنوان برتر مطالب کتاب قرار نگیرد اما ناگزیر مانند سبزیجات که خوردن آن‌ها برای سلامتی ضروری می‌باشد، مطالعه‌ی ASN.1 نیز لازم است. پروتکل ASN.1 یک استاندارد ISO به شمار می‌آید که از آن در تعدادی از پروتکل‌های اینترنت به‌ویژه در

1- Local configuration datastore

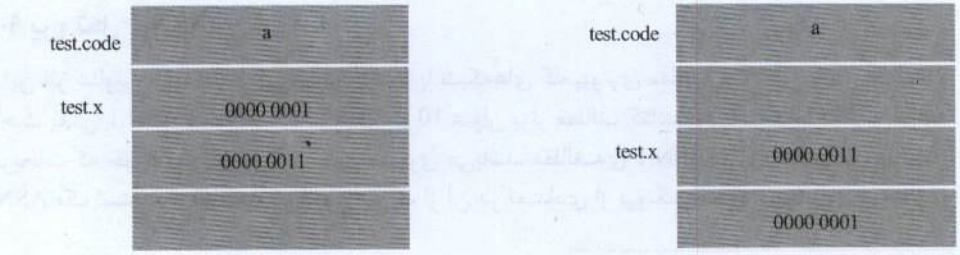
2- View based access control model configuration MIB

ارتباط با مدیریت شبکه استفاده می‌شود. به عنوان مثال ملاحظه شد متغیرهای MIB در پروتکل SNMP بسیار وابسته به ASN.1 می‌باشند. به‌منظور درک اهمیت ASN.1 بار دیگر مطلب با اشاره به یک مثال دنبال می‌شود. فرض کنید فردی با موفقیت نسخه‌ای از داده‌های یک حافظه در کامپیوتری را مستقیماً وارد حافظه‌ی یک کامپیوتر دوردست می‌کند. اگر این عمل امکان‌پذیر باشد آیا چالش ارتباط برطرف می‌گردد؟ پاسخ به این پرسش بستگی به تعریف و برداشت ازعبارت «چالش ارتباط» دارد. بدیهی است یک نسخه‌برداری حافظه به حافظه عیناً تعامل بیت‌ها و بایت‌ها از یک ماشین به دیگری را امکان‌پذیر می‌سازد. اما چنین نسخه‌برداری عینی از بیت‌ها و بایت‌ها به مفهوم آن است هنگامی که نرم‌افزاری در کامپیوتر گیرنده، این اطلاعات را دریافت می‌دارد همان مفهومی را از آن استنباط می‌کند که در حافظه‌ی کامپیوتر فرستنده ذخیره شده است؟ پاسخ این پرسش الزاماً «خیر» است. علت نیز معماری متفاوت کامپیوترها می‌باشد. تفاوت در OS آن‌ها و سرانجام تفاوت در برنامه‌های مفسر آن‌ها است، که از قوانین و مفاهیم یکسان به‌منظور ذخیره و بازیابی داده‌ها بهره نمی‌گیرند. چنان‌چه داده‌ها بایستی مورد تعامل و ذخیره‌سازی در بین چندین کامپیوتر مانند آن‌چه در هر شبکه‌ی ارتباطی روی می‌دهد، قرار گیرند آن‌گاه چالش بازیابی و نمایش داده‌ها بایستی به‌صورت آشکار راهکاری را شامل شود.

به‌عنوان یک مثال برای این چالش، قطعه برنامه‌ی ساده‌ی C در ذیل را درنظر بگیرید. چگونه این ساختمان داده در حافظه قرار می‌گیرد؟

```
struct {
  char code;
  int x;
  } test;
test.x = 259;
test.code = 'a';
```

بخش چپ شکل ۹-۶ جایگذاری احتمالی این داده را در یک معماری فرضی نشان می‌دهد. یک بایت از حافظه شامل کاراکتر a به دنبال یک کلمه‌ی 16 بیت است که ارزش کامل 259 را به‌صورت ابتدا ذخیره‌ی با ارزش‌ترین بایت خود در حافظه نگه‌داری می‌کند. شکل ۹-۶ در بخش سمت راست خود قرارگیری همین اطلاعات را در کامپیوتر دیگری نشان می دهد. در حافظه‌ی کامپیوتر سمت راست شکل ۹-۶ به دنبال کاراکتر a ارزش کامل 259 به صورت نخست کم‌ارزش‌ترین بایت با ارزش کامل 16 بیت هم طراز شده با آغاز مرز کلمه‌ی 16 بیت ذخیره شده است. بدیهی است اگر نسخه‌برداری از حافظه‌ی این دو کامپیوتر با بهره‌گیری از یکی از این دو معماری انجام گیرد، نتیجه‌ی حاصل از این دو رویکرد کاملاً متفاوت خواهد بود.



شکل ۹-۶ ♦ دو قالب ذخیره‌سازی داده بر روی دو معماری مختلف

این حقیقت که معماری‌های گوناگون قالب داده‌های داخلی متفاوت را شامل می‌شوند، یک چالش واقعی و مهم به‌شمار می‌آید. اشکال در ذخیره‌سازی ارزش‌های کامل در قالب‌بندی‌های مختلف چنان متداول است که دارای نام‌های **big-endian** در صورتی است که، با ارزش‌ترین بایت از نقطه نظر ذخیره‌سازی ابتدا در اولویت باشد، یعنی مفهوم آن ذخیره‌ی با ارزش‌ترین بایت در پایین‌ترین بخش آدرس حافظه است و **little-endian** که برعکس آن می‌باشد. دو واژه‌ی نام برده از نقطه نظر ذخیره‌سازی به داستان تخیلی لی‌لی‌پوت برمی‌گردد که دو گروه مردمان آن سرزمین تصمیم می‌گیرند، شکستن تخم مرغ را از دو سوی آن آزمایش نمایند. یک گروه در سرزمین لی‌لی‌پوت اصرار بر شکستن تخم‌مرغ از قاعده‌ی بزرگ‌تر آن داشتند، که این همان مفهوم big-endian است و گروه دیگر اصرار بر شکستن از قاعده‌ی کوچک‌تر تخم‌مرغ می‌ورزیدند و بنابراین واژه little-endian مفهوم این موضوع می‌باشد. در آن سرزمین و روزگاران این اختلاف نظر سبب وقوع جدال‌های داخلی بزرگ بنابر سرآینده‌ی داستان شده است!

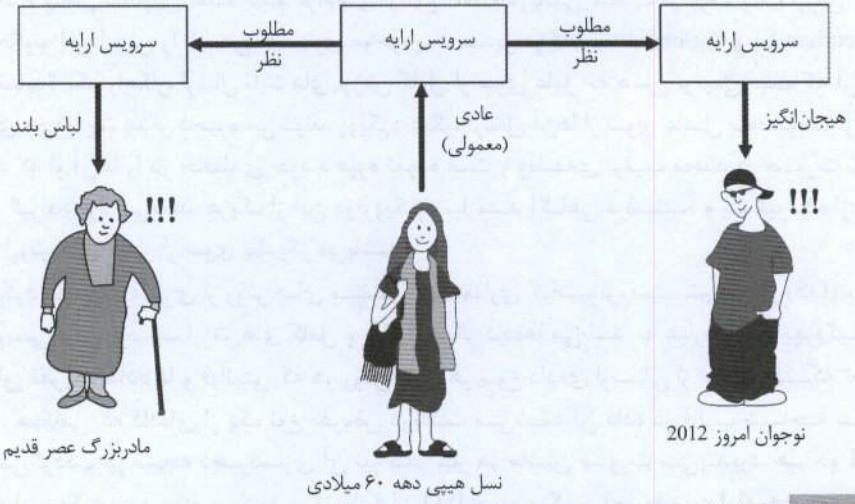
با توجه به روش‌های ذخیره‌سازی متفاوت و نیز نمایش داده‌ها در کامپیوترهای مختلف، پروتکل‌های شبکه با این اختلاف چگونه برخورد می‌کنند؟ به عنوان مثال، یک عامل SNMP هنگام ارسال پیام Response شامل شمارش تعداد داده گرام‌های از نوع UDP دریافتی که به‌صورت یک ارزش کامل است، چگونه این ارزش را بایستی به سوی موجودیت مدیر در قالب big-endian و یا little-endian ارسال نماید؟ یک راهکار، ارسال بایت‌های ارزش کامل از سوی عامل به همان ترتیبی است که آن‌ها در حافظه‌ی موجودیت مدیر ذخیره می‌شوند. رویکرد دیگر، ارسال آن‌ها از سوی عامل به‌صورت ترتیبی می‌باشد که او آن‌ها را در حافظه‌ی خود ذخیره نموده است و وظیفه‌ی ترتیب مجدد در صورت نیاز را برعهده گیرنده قرار می‌دهد. هریک از این دو رویکرد نیازمند آگاهی فرستنده و یا گیرنده از قالب نمایش ارزش‌های کامل از سوی یکدیگر می‌باشند.

رویکرد سوم، بهره‌گیری از روش‌های مستقل از معماری کامپیوتر، سیستم عامل (OS) و زبان برنامه‌نویسی برای توصیف ارزش‌های کامل و یا انواع دیگر داده‌ها می‌باشد. به عبارت دیگر بهره‌گیری از یک **زبان تعریف داده‌ها** و قوانینی که هر روش برای هر نوع داده‌ی ارسالی از طریق شبکه تعریف می‌کند، هنگامی که داده‌ای از یک نوع مفروض دریافت می‌شود آن داده در قالب شناخته شده‌ای وصول می‌گردد و در نتیجه ذخیره‌سازی آن نیز بنابر نیاز هر ماشین صورت می‌پذیرد. هر دو SMI و ASN.1 از رویکرد سوم بهره می‌گیرند. در واژگان ISO این رویکرد معروف به **ارایه‌ی سرویس** می‌باشد. مفهوم آن سرویس ارسال و دریافت اطلاعات از یک ماشین با قالب ویژه به موجودیت دیگر است. شکل ۹-۷ نمادهای واقعی این چالش را نشان می‌دهد. در این شکل گیرنده، فلسفه‌ی اصلی تعامل که فرستنده مایل به آن می‌باشد را درک نمی‌کند. همان‌گونه که شکل ۹-۸ نشان می‌دهد، با بهره‌گیری از امکانات ارایه‌ی سرویس چالش ترجمه به زبان قابل فهم دو نسل مختلف و به عبارت بهتر سه نسل گوناگون برطرف می‌گردد. هریک از سه مورد اشاره شده در شکل ۹-۸ از لحاظ زبان و محاوره مستقل از دیگری می‌باشد.



شکل ۹-۷ ♦ چالش موجود در تفهیم مطلب

جدول ۹-۵ چندین نوع از داده‌های ASN.1 را نشان می‌دهد. داده‌های از نوع INTEGER، OCTET STRING و OBJECT IDENTIFIER از پروتکل SMI که قبلاً مورد مطالعه قرار گرفتند. خواننده‌ی علاقه‌مند جهت مطالعه‌ی عمیق ASN.1 به سامانه‌ی اینترنتی ASN.1 هدایت می‌شود.

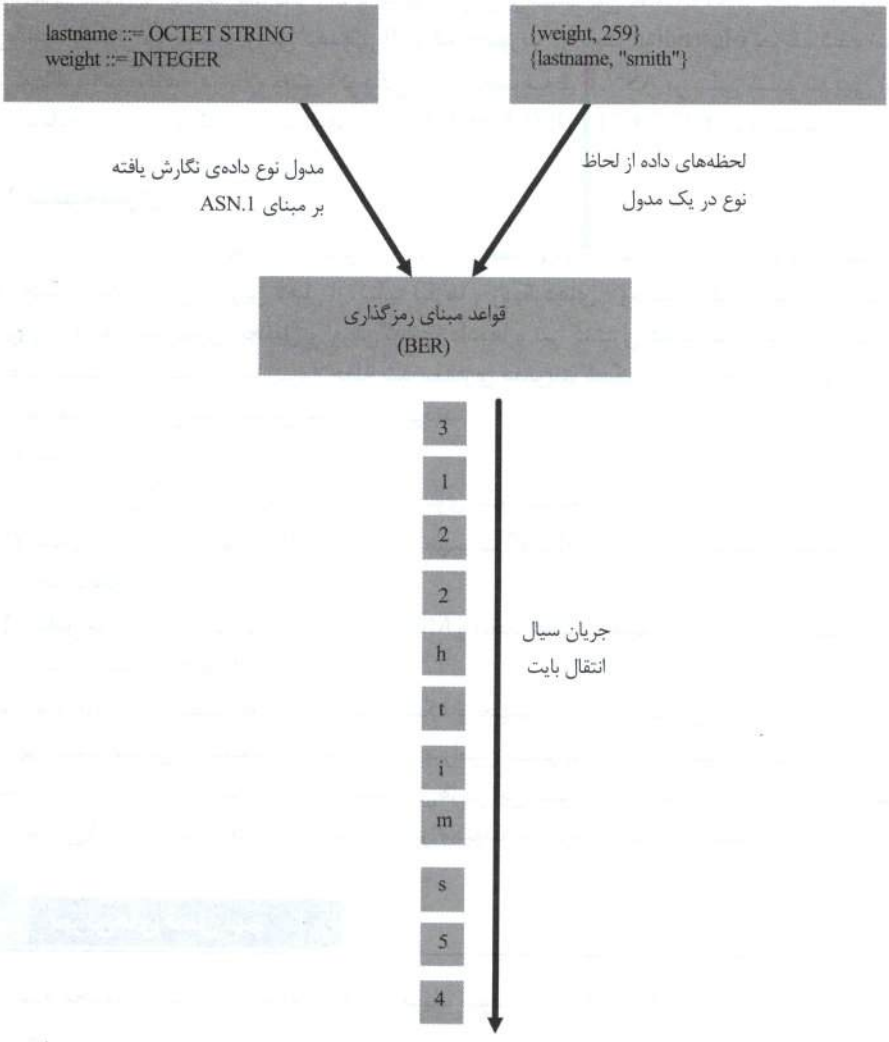


شکل ۹-۸ ♦ راهکار ارایه‌ی سرویس به‌منظور چالش تفهیم داده‌ها

جدول ۹-۵ ♦ انواع داده‌های ASN.1 گزینش شده

برچسب	نوع	توصیف
1	BOOLEAN	دو ارزش درست "T" و غلط "F"
2	INTEGER	ارزش دلخواه بزرگ
3	BIT STRING	فهرستی از یک یا چندین بیت
4	OCTET STRING	فهرستی از یک یا چندین بایت
5	NULL	بدون ارزش (تهی)
6	OBJECT IDENTIFIER	نام، در درخت نام‌گذاری استاندارد ASN.1
9	REAL	ارزش ممیز شناور

علاوه بر تأمین زبان توصیف داده، ASN.1 **قوانین رمزگذاری مبنا (BER)**[1] را نیز تأمین می‌کند، که از آن‌ها به‌منظور تعریف ویژگی اشیاء تعریف شده به‌وسیله‌ی زبان توصیف داده ASN.1 برای ارسال از طریق شبکه مورد استفاده قرار می‌گیرد. قواعد BER از رویکردی به نام **رویکرد TLV** به مفهوم **(Type, Length, Value)** رمزگذاری داده را برای انتقال انجام می‌دهند. برای هر قلم از داده ارسالی، نوع، طول و ارزش داده و آن‌گاه ارزش واقعی داده به‌ترتیب عنوان شده ارسال می‌گردد. با این تعبیر ساده، داده‌ی دریافتی حالت خود شناسه‌گر دارد.



شکل ۹-۹ ♦ نمونه‌ای از رمزگذاری BER

شکل ۹-۹ نشان می‌دهد، چگونه دو قلم در یک مثال ساده ارسال می‌شوند. در این مثال، فرستنده

1- Basic Encoding Rules

مایل به ارسال رشته کاراکتر "smith" و به دنبال آن ارزش دهدهی 259 در قالب باینری: (0000 0001 0000 0001) و یا یک بایت با ارزش 1 و به دنبال آن یک بایت ارزش 3 با فرض ترتیب ذخیره bigendian می‌باشد. نخستین بایت در این جریان ارسالی دارای ارزش 4 بوده که مفهوم آن نوع داده‌ی ارسالی به‌صورت OCTET STRING است. در مفهوم TLV رمز آن "T" می‌باشد. بایت دو در جریان داده‌ها حاصل طول OCTET STRING است و در این مثال 5 می‌باشد، بایت سوم باز هم از نوع COTET STRING و طول 5 بوده، که همراه آن نماد ASCII حرف s است. ارزش‌های T، L و V از قلم داده‌های بعدی به ترتیب عبارتند از 2 به مفهوم ارزش برچسب نوع INTEGER، 2 به مفهوم یک ارزش کامل با طول دو بایت و ارزش دهدهی 259 که به‌صورت دو بایت big-endian تعریف شده است.

در مطالب اشاره شده در بالا، بخش کوچکی از زیرمجموعه‌ی ASN.1 بررسی شد. بدیهی است مطالب بنیادی‌تر این پروتکل در مستندهای [ISO X.680, 2002] و [OSS2007] توصیف شده‌اند.

۹-۱۰ نتیجه‌گیری

مطالب پیرامون مدیریت شبکه و در مقیاس بزرگ‌تر آن همه‌ی موارد شبکه‌های کامپیوتری اکنون به انتهای خود رسید! در این آخرین فصل از کتاب نیازها و رویکردهای لازم به‌منظور نظارت، سنجش، جمع‌آوری داده‌ها، پیکربندی، تحلیل و ارزش‌گذاری داده‌ها و نیز کنترل عملیات در واژگان مدیریت شبکه‌های کامپیوتری مطالعه شدند. ملاحظه شد معماری مدیریت شبکه در اطراف 5 موضوع کلیدی گردش می‌نمود. این پنج مورد شامل موارد ذیل بودند:

(۱) مدیر شبکه.

(۲) مجموعه‌ای از تجهیزات مدیریت شده از سوی مدیر شبکه.

(۳) مبانی مدیریت اطلاعات (MIB) در این تجهیزات که شامل داده‌های وضعیت تجهیزات و نیز حالت‌های عملیاتی آن‌ها می‌باشد.

(۴) عامل‌های راه دور که گزارش اطلاعات MIB و اتخاذ عملیات مربوطه را با کنترل و نظارت مدیر شبکه تهیه و انجام دادند.

(۵) یک پروتکل به‌منظور تعامل بین مدیر شبکه و تجهیزات دور از دسترس وی.

بدیهی است عناوین و موضوعات بسیار زیادی پیرامون مدیریت شبکه وجود دارند که خارج از چارچوب این اثر بوده و بنابراین از عنوان نمودن آن‌ها در این فصل پرهیز شده است. اگرچه این عناوین بسیار مهم می‌باشند، اما در حقیقت نیازمند یک اثر مکتوب جداگانه درخور خود هستند.



تمرین‌های فصل نهم

۱- پنج محدوده‌ی مدیریت شبکه که ISO تعریف نموده است، کدامند؟

۲- تفاوت بین مدیریت شبکه و مدیریت خدمات چیست؟

۳- مفاهیم ذیل را تعریف نمایید:

الف) موجودیت مدیریت

ب) وسیله‌ی مدیریت شده

ج) عامل مدیریت

د) پروتکل مدیریت شبکه

۴- تعداد 7 نوع پیام به‌کار گرفته شده در SNMP کدامند؟

۵- نقش SMI در مدیریت شبکه چیست؟

۶- مفهوم یک موتور SNMP چیست؟

۷- تفاوت مهم بین یک پیام واکنش تقاضا و یک پیام trap در SNMP چیست؟

۸- هدف درخت شناسه‌ی شیء ASN.1 چیست؟

۹- مفهوم رمزگذاری TLV چیست؟

۱۰- نقش ASN.1 در لایه‌ی ارایه‌ی مدل مرجع ISO/OSI چیست؟

۱۱- دو روش ارتباط بین یک مدیر و موجودیت مدیریت شده عبارتست از روش پرسش و پاسخ و trapping. نقاط قوت، ضعف این رویکردها از نقطه نظر (۱) سرآیند (۲) زمان آگهی هنگام رخدادهای غیرمتعارف و (۳) قدرت‌مندی با توجه به اتلاف پیام‌ها بین مدیر و تجهیزات کدامند؟

۱۲- شکل ۹-۹ را در نظر بگیرید. رمزگذاری BER برای عبارت ذیل چیست؟

{weight,276} {lastname, "Julia"}

۱۳- شناسه‌گر شیء ASN.1 برای پروتکل ICMP چیست؟ از شکل ۹-۳ استفاده کنید.

۱۴- چرا طراحان SNMP استفاده از پروتکل UDP را بر TCP ترجیح داده‌اند؟

۱۵- فرض کنید در استخدام یک شرکت می‌باشید که مایل است یک خط تولید مدیریت MIB طراحی نماید. درخت شناسه‌ی شیء شکل ۹-۳ در کدام نقطه بایستی ثبت گردد؟

۱۶- با توجه به شکل ۹-۹ رمزگذاری BER عبارت ذیل چیست؟

{weight,160} {lastname,"child"}

۱۷- علاوه بر نیروگاه الکتریکی و دماغه‌ی هواپیما چه موارد دیگری از سیستم‌های توزیع شده که بایستی کنترل شوند را می‌شناسید، جزئیات آن‌ها را توضیح دهید.

یادداشتی بر آدرس‌های مراجع URL

در مراجع اشاره شده در ذیل، آدرس‌های URL مربوط به وب و اسناد انحصاری وب به غیر از انتشارات مطبوعاتی و اسناد همایش‌های علمی در زمینه‌ی شبکه‌های کامپیوتری فراهم آمده‌اند. برعکس رویکرد نسخه‌های پیشین کتاب آدرس URL همایش‌ها و انتشارات آنها در اثر فعلی در ذیل عنوان مراجع به این دلیل قید نشده‌اند که این گونه اسناد را می‌توان از طریق موتورهای جست‌وجو و سامانه‌های آنها (مانند مجموعه مقالات ACM Sigcomm) در یک آدرس URL مانند http://www.acm.org/sigcomm: و یا اشتراک در کتابخانه‌های دیجیتالی دستیابی نمود. اگرچه همگی URLهای ارائه شده در ذیل عنوان مراجع کتاب حاضر تا تاریخ نشر این کتاب یعنی ماه فوریه 2012 میلادی معتبر بوده‌اند، اما برخی از آنها ممکن است از رده خارج شده و در شبکه‌ی جهانی اینترنت حضور نداشته باشند. بنابراین خوانندگان دعوت می‌شوند به‌منظور دستیابی آخرین آدرس‌های URL در این عرصه به سامانه‌ی انحصاری این اثر در آدرس URL ذیل مراجعه نمایند:

http://www.awl.comlkurose-ross

- **یادداشتی بر اسناد RFC**

نسخه‌های اسناد RFC از طریق مراجعه به سامانه‌های بی‌شماری قابل دستیابی می‌باشند. سردبیری جامعه‌ی اینترنت این اسناد(RFCEIS) دارای سامانه‌ی ویژه خود با آدرس URL ذیل است:

http://www.rfc-editor.org

این سامانه امکانات لازم را به‌منظور جست‌وجو و رهگیری یک RFC در اختیار علاقه‌مندان قرار می‌دهد. معمولاً RFCهای اینترنت از طریق نشر RFCهای جدید به‌روزرسانی و یا از رده خارج می‌شوند. سامانه‌ی پیشنهادی به خوانندگان برای منابع اصلی اسناد RFC همان سامانه‌ی اشاره شده در بالا می‌باشد.