مدرسان شریف

فصل اول

« مفاهیم پایه شبکه‌های کامپیوتری »

سال‌ها قبل از ایجاد و شکل‌گیری شبکه‌های کامپیوتری، از واژه شبکه، در حوزه‌های مختلف استفاده می‌شد (برای مثال شبکه راه آهن کشور). با این حال از اواسط قرن گذشته بود که شبکه‌های کامپیوتری رشد خود را آغاز کردند. به هر حال آنچه که در این کتاب با یکدیگر بررسی خواهیم نمود، شبکه‌های کامپیوتری می‌باشد. بنابراین از این جا به بعد منظور ما از «شبکه»، «شبکه‌های کامپیوتری» خواهد بود. قبل از شروع بحث، اجازه دهید تا ابتدا تعریفی را برای شبکه‌های کامپیوتری ارائه دهیم.

شبکه‌های کامپیوتری

شبکه کامپیوتری به مجموعه‌ای از تجهیزات (device) به هم مرتبط گفته می‌شود که توانایی تبادل داده و اطلاعات را با یکدیگر داشته باشند. به اعضای شبکه در اصطلاح «گره (node)» گفته می‌شود.

نکته۱: دقت کنید که منظور از «تجهیزات» یا گره، لزوماً کامپیوتر نیست. به‌عنوان مثال یک تلفن همراه یا لپ‌تاپ و یا چند کامپیوتر با چاپگر نیز می‌توانند با همدیگر تشکیل شبکه دهند.

تذکر۱: گاهی اوقات به جای واژه «گره»، از واژه «میزبان (host)» نیز استفاده می‌شود.

اهداف ایجاد شبکه

شاید این سوال برای شما هم به وجود آمده باشد که اصولاً چه نیازی به ایجاد و استفاده از شبکه‌های کامپیوتری وجود دارد؟ در ذیل، برخی از اهداف و کاربردهای شبکه را با همدیگر مرور خواهیم نمود. البته کاربرد شبکه به همین چند مورد محدود نیست و مواردی که در ادامه می‌آید رایج‌ترین آن‌ها می‌باشد.

به اشتراک گذاشتن منابع: منابع، می‌توانند هم به شکل سخت‌افزار باشند و هم نرم افزار. به‌عنوان مثالی برای منابع **سخت‌افزار**، می‌توان به جای آنکه برای هر کامپیوتر حافظه (یا CPU یا چاپگر) جداگانه‌ای در نظر گرفت، حافظه (یا CPU یا چاپگر) مشترکی برای آن‌ها ایجاد کرده و آن را به اشتراک گذاشت که با این کار در هزینه نیز صرفه‌جویی می‌شود.

از طرفی دیگر، منابع می‌توانند به شکل **نرم‌افزاری** نیز باشند. به‌عنوان مثال شرکتی را با 50 پرسنل در نظر بگیرید. اگر کارمندان این شرکت برای انجام کارهای خود نیاز به نرم‌افزار خاصی داشته باشند، یک راه این است که پشت هر 50 کامپیوتر نشسته و آن نرم افزار خاص را روی تک‌تک دستگاه‌ها نصب نمود که مسلماً پروسه طولانی مدت خواهد بود. اما راه دیگر، به اشتراک گذاشتن آن نرم‌افزار خاص بر روی سرور شبکه شرکت است تا هر فردی که نیاز به آن داشته باشد، از طریق سرور کار خود را انجام دهد. به همین شکل می‌توان هر گونه داده یا اطلاعاتی را نیز که اعضای شرکت به آن نیاز دارند، از طریق شبکه به راحتی در اختیار آن‌ها قرار داد.

ایجاد ارتباط: یکی از اهداف مهم شبکه، ایجاد بستری مناسب برای امکان ارتباط بین افراد مختلف، در مکان‌ها و شرایط گوناگون و در یک کلام، انتقال داده است. پست الکترونیکی یا همان Email (مانند o.mohabati@gmail.com)، چَت، کنفرانس‌های ویدئویی، تلفن‌های اینترنتی و ... مثال‌هایی از این دست هستند.

همان‌طور که ذکر شد سرویس‌های ارائه شده توسط شبکه به همین چند مورد محدود نمی‌شود. به‌عنوان مثال می‌توان خدماتی از قبیل: خرید اینترنتی (تجارت الکترونیک)، موتورهای جستجو، آموزش الکترونیکی، بازی‌های شبکه‌ای و ... را نیز نام برد.

مثال۱: کدام مورد، از اهداف استفاده از شبکه به شمار می‌رود؟

۱) افزایش سرعت ۲) اشتراک منابع ۳) ایجاد ارتباط ۴) همه موارد

☑ **پاسخ:** گزینه «۴» دقت کنید که مواردی مانند افزایش سرعت، افزایش قابلیت اطمینان و سرگرمی نیز می‌توانند از اهداف دیگر شبکه باشند.

زیرشبکه (Subnet)

به مجموعه واسطه‌های میانی و کانال (لینک)ها، زیرشبکه (Subnet) گفته می‌شود. با این تعریف مشخص است که کاربرد اصلی زیرشبکه، انتقال داده‌ها است. منظور از واسطه‌های میانی، دستگاه‌هایی است که برای اتصال گره به شبکه از آن‌ها استفاده می‌شود (مانند کارت شبکه).

تذکر۲: در فصل‌های آتی تعاریف دیگری از زیرشبکه را خواهیم دید.

پروتکل

به مجموعه قوانین و قراردادهایی که بین فرستنده و گیرنده باید تنظیم شود تا بتوانند با هم ارتباط داشته باشند یا در اصطلاح زبان همدیگر را متوجه شوند، پروتکل گفته می‌شود. پروتکل وظایف فرستنده، گیرنده و نحوه ارسال و دریافت داده‌ها را دقیقاً مشخص می‌کند. از انواع پروتکل می‌توان به مواردی مانند: HTTP، FTP، TCP، IP و... اشاره کرد. در فصل‌های آینده بیش‌تر با وظایف پروتکل‌های مختلف آشنا خواهیم شد.

مثال۲: مجموعه قوانینی که باعث می‌شود دو طرف ارتباط با هم رابطه مناسب و مشخصی داشته باشند چه نامیده می‌شود؟

۱) DNS ۲) Subnet ۳) Protocol ۴) Hub

☑ **پاسخ:** گزینه «۳» به مجموعه قوانین و قراردادهایی که بین فرستنده و گیرنده باید تنظیم شود تا بتوانند با هم ارتباط داشته باشند یا در اصطلاح زبان همدیگر را متوجه شوند پروتکل گفته می‌شود.

شبکه‌های کامپیوتری را می‌توان بر اساس معیارهای مختلف طبقه‌بندی نمود. از جمله این معیارها، می‌توان به حوزه و وسعت جغرافیایی تحت پوشش، نحوه سرویس‌دهی و سرویس‌گیری و سیمی یا بی‌سیم بودن آن‌ها اشاره نمود. در ادامه، این موارد را بررسی می‌کنیم.

انواع شبکه از نظر وسعت ناحیه تحت پوشش

از این نظر شبکه‌ها را معمولاً به سه دسته LAN، MAN و WAN تقسیم می‌کنند.

شبکه‌های LAN (Local Area Network) معمولاً وسعت محدودی در حدود یک یا چند ساختمان دارند. حتماً تا به حال متوجه شده‌اید که در برخی نقاط شهر، امکان اتصال به اینترنت بی‌سیم وجود دارد. اغلبِ این شبکه‌ها، از نوع استاندارد IEEE 802.11 هستند که نوعی از شبکه‌های بی‌سیم LAN به شمار می‌روند. احتمالاً در دفتر آموزش دانشکده خود دیده‌اید که چندین کامپیوتر در یک اتاق به یکدیگر و یا به یک چاپگر متصل هستند. در این حالت نیز یک شبکه LAN ایجاد شده که در اکثر مواقع و در چنین حالاتی از پروتکل اِترنت برای برقراری ارتباط استفاده می‌کنند. از خصوصیات شبکه‌های LAN می‌توان به ساده بودن مدیریت، تعداد کم گره‌ها، ارزان بودن، نرخ انتقال بالا و نرخ خطای کم اشاره کرد.

تذکر۳: در مورد مفاهیمی همچون استاندارد 802.11 و پروتکل اِترنت در فصول بعدی توضیح داده خواهد شد.

شبکه‌های MAN (Metropolitan Area Network)، منطقه یک شهر را تحت پوشش خویش قرار می‌دهند. وایمکس (WiMax)که امروزه در کشور ما هم ایجاد شده مثال معروفی از شبکه‌های MAN می‌باشد.

در نهایت **شبکه‌های WAN (World Area Network)**، وسعتی در حد کشور و یا حتی جهان را دارند که از اتصال چندین LAN یا MAN به وجود می‌آیند. اینترنت بهترین مثال برای شبکه‌های WAN می‌باشد. در این شبکه‌ها برخلاف شبکه‌های محلی، مدیریت پیچیده، هزینه بالا و تعداد گره‌ها زیاد است. در حقیقت اینترنت را می‌توان شبکه‌ای از شبکه‌ها فرض نمود که از اتصال میلیون‌ها شبکه به یکدیگر به وجود آمده است. برخی از رایج‌ترین کاربردهای اینترنت عبارتند از: پست الکترونیکی، موتورهای جستجو، خرید اینترنتی، حراج اینترنتی، ویدئو کنفرانس‌ها، انتقال فایل‌ها، جستجو در وب، بازی‌های تحت شبکه، تلفن اینترنتی، آموزش الکترونیکی (مجازی) و ...

لازم به ذکر است که برخی منابع در این طبقه‌بندی، شبکه‌های دیگری همچون شبکه‌های PAN و یا GAN را نیز در نظر می‌گیرند. **شبکه‌های PAN** (Personal Area Network) از شبکه‌های LAN کوچکتر بوده و وسعت آن‌ها از چندین متر (مثلاً دو سه متر) تجاوز نمی‌کند. به‌عنوان مثال وقتی شما از طریق بلوتوث یا اینفرارد (مادون قرمز) ارتباط برقرار می‌کنید، تشکیل یک شبکه PAN داده‌اید.

شبکه‌های GAN (Global area Network) نیز بزرگتر از WAN در نظر گرفته می‌شوند. معمولاً گستره WAN در حد یک کشور یا قاره در نظر گرفته می‌شود و گستره GAN در حد کره زمین.

انواع شبکه از نظر نحوه سرویس‌دهی (client/server و peer-to-peer)

برخی اوقات نحوه سرویس‌دهی شبکه را «**نرم افزار شبکه**» نیز می‌نامند. در این رابطه می‌توان دو نوع شبکه client/server و peer-to-peer را نام برد.

در **شبکه‌های** client/server، برخی از تجهیزات، نقش سرویس‌دهنده (**سرور**) و برخی دیگر نقش سرویس‌گیرنده (**کِلاینت**) را دارند. به عبارت دیگر هر عنصر شبکه یا سرویس‌گیرنده است یا سرویس‌دهنده. در این حالت باید روی دستگاه سرور، سیستم عامل خاصی نصب شده باشد (مثلاً Windows Server 2003,2008 و یا لینوکس) تا بتواند وظایف خود را در شبکه به درستی انجام داده و به درخواست‌های کلاینت پاسخ درستی دهد. مدیریت در شبکه‌های client/server به خوبی قابل پیاده‌سازی است و به علت وجود همین مدیریت، امنیت آن‌ها به طور معمول، بیش‌تر از شبکه‌های peer-to-peer است.

چنانچه تعداد گره‌ها زیاد باشد، از این شبکه‌ها استفاده می‌شود. البته از آنجا که ممکن است با خرابی سرور، کل شبکه از کار بیفتد معمولاً از چندین سرور استفاده می‌شود تا در صورت بروز مشکل برای سرور اصلی، سرورهای دیگر برای سرویس‌دهی آمادگی داشته باشند.

تذکر۴: دقت کنید که منظور از سرور، لزوماً یک کامپیوتر پیشرفته نیست.

این تصور غلطی است که سرور لزوماً باید یک کامپیوتر بسیار قدرتمند باشد. حتی کامپیوتر خانگی شما با سیستم عامل Windows XP و یا مشابه آن نیز در برخی کاربردها می‌تواند نقش سرور را ایفا کند. علاوه بر این تعداد سرورها در شبکه لزوماً، یک عدد نیست. بدین معنی که در یک شبکه می‌توان سرورهای مختلفی را متصور بود از جمله: Web server، File Server، Database Server، Proxy Server ، DNS Server و ... .

در **شبکه‌های** peer-to-peer، هر دستگاه همزمان، ضمن اینکه از برخی دستگاه‌ها سرویس می‌گیرد، به برخی دیگر نیز سرویس ارائه می‌دهد. به عبارت دیگر یک دستگاه هم نقش سرویس‌دهنده را بازی می‌کند هم سرویس گیرنده را. بنابراین در این نوع از سرویس‌دهی، اعضای شبکه برتری خاصی نسبت به همدیگر ندارند.

از مزایای شبکه‌های peer-to-peer ، ارزان قیمت بودن آن‌ها است. ضمناً کار با آن‌ها از آنجا که به سیستم عامل خاصی نیاز ندارند، ساده است. اما عیب بزرگ آن‌ها محدودیت در تعداد گره‌ها (حداکثر 20 عدد) است. در این نوع از شبکه هر فردی مسئول دستگاه خویش است. لذا از قبل باید آموزش‌های لازم به کاربران در این خصوص صورت گیرد.

مثال۳: کدام عبارت در مورد شبکه‌های peer-to-peer صحیح می‌باشد؟

۱) تعدادی از گره‌ها نقش سرور و تعدادی دیگر نقش کلاینت دارند.

۲) هر گره همزمان می‌تواند هم سرور باشد هم کلاینت

۳) تعداد گره‌هایی که نقش سرور دارند با تعداد گره‌هایی که نقش کلاینت دارند برابر می‌باشد.

۴) هیچکدام

☑ **پاسخ:** گزینه «۲» دقت کنید که گزینه «۱» در رابطه با شبکه‌های client/server مصداق دارد. در ضمن در هیچ نوعی از شبکه، هیچ الزامی به برابر بودن تعداد گره‌های سرور با کلاینت وجود ندارد.

مثال۴: کدام گزینه از انواع سرورها در نظر گرفته نمی‌شود؟

۱) DNS ۲) Database ۳) ENDS ۴) Web

☑ **پاسخ:** گزینه «۳» هر سه گزینه دیگر از انواع مختلف سرورها به شمار می‌روند.

توپولوژی شبکه

به نحوه و الگوی چیدمان عناصر شبکه در کنار یکدیگر و چگونگی ارتباط آن‌ها با یکدیگر، در اصطلاح توپولوژی یا همبندی گفته می‌شود. مهمترین انواع توپولوژی عبارتند از: BUS (خطی)، Ring (حلقوی)، Star (ستاره)، Mesh (مش)،Tree (درختی) و Hybrid (ترکیبی) که در ادامه به بررسی آن‌ها می‌پردازیم.

توپولوژی BUS (خطی)

در این نوع توپولوژی، ارتباط بین اعضای شبکه از طریق یک کابل (گذرگاه، باس) مشترک (که گاهاً **ستون فقرات یا backbone** نیز نامیده می‌شود) صورت می‌گیرد؛ بدین معنی که کلیه عناصر شبکه، به آن کابل متصل هستند. هر دستگاهی که بخواهد ارسال داده داشته باشد مجبور است داده‌های خود را روی کابل مشترک قرار داده و از طریق آن داده خود را به مقصد ارسال کند. به دلیل مشکلاتی که این توپولوژی دارد در حال حاضر کاربرد بسیار کمی دارد. از ویژگی‌های توپولوژی باس می‌توان به موارد زیر اشاره نمود:

۱ـ سادگی

۲ـ تعداد کابل‌های مورد استفاده (نسبت به برخی توپولوژی‌ها) کم است.

۳ـ گسترش شبکه ساده است. بدین معنی که برای افزایش گره‌ها و اعضای جدید، کار چندان سختی نباید صورت گیرد. تنها باید عنصر جدید را به کابل مشترک وصل نمود.

۴ـ هر گره برای اتصال به شبکه، تنها نیاز به یک پورت دارد.

۵ـ امنیت پایین: اگر نفوذگر موفق شود کنترل باس را در دست گیرد، به کلیه اطلاعات مبادله شده دسترسی پیدا خواهد کرد.

۶ـ باس در هر لحظه، تنها باید در اختیار یک گره باشد. بدین مفهوم که دو گره به طور همزمان نمی‌توانند برای انتقال داده‌های خود از باس استفاده نمایند. بنابراین اگر باس مشغول باشد تا زمان آزاد شدن آن، هیچ گره‌ای حق آغاز تبادل داده خود را نخواهد داشت؛ در غیر این صورت **تصادم (collision)** رخ خواهد داد که باعث خراب شدن داده‌های ارسالی می‌شود.

۷ـ اگر کابل مشترک صدمه‌ای ببیند، عملکرد کل شبکه مختل خواهد شد.

۸ـ برای جلوگیری از انعکاس سیگنال از انتهای باس، باید در انتهای کابل از **خاتمه دهنده (terminator)** استفاده نمود.

۹ـ سخت و مشکل بودن عیب‌یابی و رفع خطا.

۱۰ـ وجود پدیده تضعیف و محدودیت در طول کابل مشترک

تذکر۵: در فصل چهارم به طور مفصل در خصوص کنترل دسترسی به رسانه‌ی مشترک، صحبت خواهیم نمود.

شکل ۱ نمونه‌ای از یک توپولوژی bus را نشان می‌دهد.

شکل ۱: نمونه‌ای از توپولوژی bus

مثال۵: نیاز به terminator در کدام توپولوژی وجود دارد؟

۱) Ring ۲) Bus ۳) Star ۴) Mesh

☑ **پاسخ:** گزینه «۲» همان‌طور که در بالا ذکر شد استفاده از terminator در توپولوژی Bus رایج است.

توپولوژی Ring (حلقوی)

در این توپولوژی گره‌های شبکه، حلقوی‌وار به یکدیگر متصل شده‌اند. بنابراین هر گره، نقش مسیر و رسانه شبکه را نیز ایفا می‌کند. بدیهی است که در چنین الگویی، هر گره تنها با دو گره دیگر به طور مستقیم در تماس است. حرکت داده‌ها می‌تواند در جهت ساعتگرد و یا پادساعتگرد صورت گیرد.

ویژگی‌های این توپولوژی عبارتند از:

۱ـ تعداد کابل مورد استفاده کم است. ۲ـ هر گره برای اتصال به شبکه، تنها نیاز به دو پورت دارد. ۳ـ حذف پدیده تضعیف (زیرا هر گره اطلاعات دریافتی خود را تکرار می‌کند). ۴ـ در صورت خرابی یکی از کابل‌ها و یا گره‌ها، عملکرد کلی شبکه مختل خواهد شد؛ چرا که امکان ارتباط اعضا با یکدیگر از بین می‌رود. ۵ـ امنیت پایینی دارد.

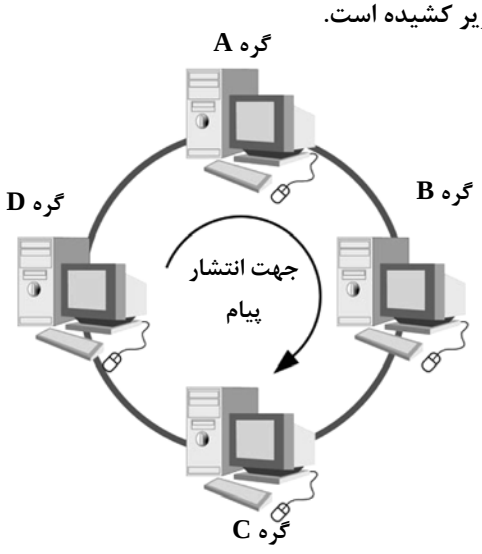
برای غلبه بر مشکلات فوق، معمولاً در توپولوژی حلقه، از دو حلقه در دو جهت متفاوت استفاده می‌شود تا اگر برای یکی از حلقه‌ها مشکلی بروز کرد، بتوان از حلقه جایگزین بهره گرفت.

۶ـ بسط شبکه و افزودن گره‌ی جدید، با از کار افتادن شبکه همراه است.

نکته۲: تعداد کابل‌های مورد نیاز در توپولوژی حلقوی یکطرفه با n گره، برابر n است.

نکته۳: کم‌ترین و بیش‌ترین کابل پیموده شده برای تبادل داده در یک شبکه حلقوی یکطرفه با n گره، به ترتیب عبارتند از: 1 و $n-1$.

شکل ۲ نمونه‌ای از توپولوژی Ring را به تصویر کشیده است.



شکل ۲: نمونه‌ای از توپولوژی Ring

مدرسان شریف

فصل سوم

«لایه پیوند داده»

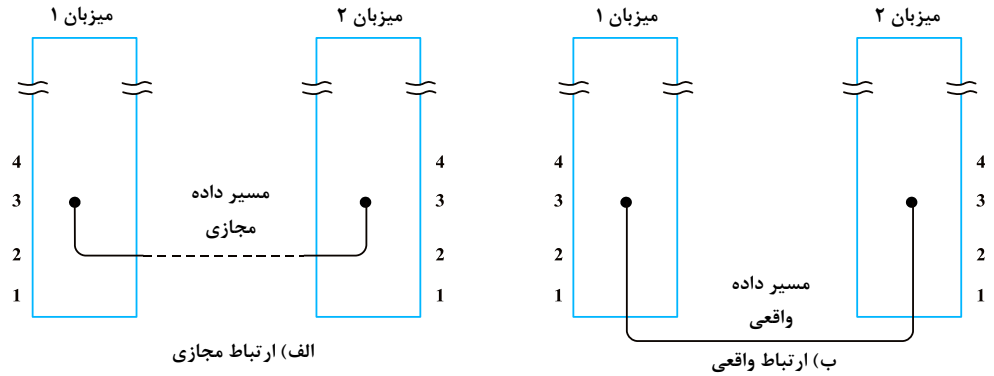
پس از مطالعه لایه فیزیکی در فصل گذشته، در این فصل به بررسی و مطالعه لایه پیوند داده می‌پردازیم. این بررسی، شامل مطالعه الگوریتم‌ها و روش‌هایی برای رسیدن به شبکه قابل اطمینان و ارتباط کارا و موثر بین دو ماشین همسایه در لایه پیوند داده می‌باشد.
متاسفانه، مدارهای ارتباطی باعث شکل‌گیری خطا می‌شوند. علاوه بر این، آن‌ها دارای پهنای باند محدودی هستند که باعث ایجاد محدودیت در حداکثر نرخ ارسال می‌شوند. ضمن اینکه وجود لینک ارتباطی باعث ایجاد تاخیر انتشار نیز می‌شود. کلیه این موارد، بر کارایی نرخ انتقال داده اثر سوئی می‌گذارند. پروتکل‌هایی که برای ارتباط استفاده می‌شوند باید کلیه این عوامل را در نظر گیرند. در این فصل با این پروتکل‌ها آشنا خواهیم شد.

وظایف لایه پیوند داده

لایه پیوند داده چندین وظیفه مهم دارد که باید آن‌ها را انجام دهد. این وظایف شامل موارد زیر می‌شود:
۱ـ تامین واسط سرویس مناسب برای لایه بالاتر از خود (لایه شبکه)
۲ـ تعیین چگونگی گروه‌بندی بیت‌های ارسالی توسط لایه فیزیکی در قالب فریم‌ها (فریم بندی)
۳ـ تنظیم جریان فریم‌ها به گونه‌ای که یک گیرنده‌ی آهسته، به واسطه دریافت داده‌های بسیار زیاد از فرستنده پرسرعت دچار مشکل نشود (کنترل جریان).
۴ـ بررسی خطاهای انتقال (کنترل خطا)
در ادامه همین فصل، این اهداف را مورد بررسی قرار خواهیم داد. البته لازم به ذکر است که لایه پیوند داده، خود به دو زیر لایه **MAC** و **LLC** تقسیم می‌شود. زیرلایه MAC با لایه فیزیکی و زیر لایه LLC با لایه شبکه در ارتباط است. به عبارت دیگر وظیفه ارائه سرویس به لایه شبکه و مدیریت ارتباطات میان دو کامپیوتر در یک کانال، بر عهده زیرلایه LLC است. زیرلایه MAC نیز دسترسی به رسانه مشترک را مورد توجه قرار می‌دهد. در این رابطه در فصل ۴ بیشتر سخن خواهیم گفت.

ارائه سرویس به لایه شبکه

وظیفه لایه پیوند داده، ارائه سرویس به لایه شبکه است. مهمترین سرویس، انتقال داده‌ها از لایه شبکه‌ی ماشینِ منبع، به لایه شبکه‌ی ماشین مقصد است. در ماشین منبع و در لایه شبکه، موجودیتی (entity) که ما آن را پردازش (process) می‌نامیم وجود دارد که تعدادی بیت را برای ارسال به مقصد، به لایه پیوند داده تحویل می‌دهد. وظیفه لایه پیوند داده انتقال این بیت‌ها به ماشین مقصد است به گونه‌ای که آن‌ها بتوانند به لایه شبکه تحویل داده شوند. (شکل ۱ الف).



شکل ۱: ارتباط مجازی و واقعی

انتقال واقعی، مسیری را دنبال می‌کند که در شکل ۱ـ ب نشان داده شده است. اما مدل ساده‌تر همان است که ارتباط بین دو فرآیند را که از پروتکل پیوند داده استفاده می‌کنند، در نظر بگیریم.

لایه پیوند داده می‌تواند برای ارئه سرویس‌های متفاوتی طراحی شود به طوری که سرویس ارائه شده در هر سیستم، می‌تواند متفاوت با دیگری باشد. با این حال، سرویس‌های معمولی که ارائه می‌شوند عبارتند از:

۱ـ سرویس بدون اتصال بدون تصدیق (Unacknowledged connectionless service)

۲ـ سرویس بدون اتصال با تصدیق (Acknowledged connectionless service)

۳ـ سرویس اتصال‌گرای با تصدیق (Acknowledged connection-oriented service)

در **سرویس بدون اتصال بدون تصدیق**، ماشین منبع، فریم‌های مستقلی را به مقصد ارسال می‌کند؛ بدون آنکه ماشینِ مقصد، دریافت آن‌ها را تصدیق کند. هیچ ارتباطی از قبل بین فرستنده و گیرنده ایجاد نمی‌شود (به فصل ۱ مراجعه کنید). اگر فریمی به خاطر نویز در لینک از دست برود، هیچ تلاشی در لایه پیوند داده برای جبران آن انجام نمی‌شود. **این نوع سرویس زمانی اهمیت پیدا می‌کند که نرخ خطا بسیار پایین باشد** به طوری که بتوان عمل بازیابی را به عهده لایه‌های بالاتر واگذار کرد. علاوه بر این، برای **ترافیک‌های بیدرنگ (real-time)** مانند صوت که در آن‌ها سرعت دریافت داده (نرخ انتقال) از صحت داده دریافتی، اهمیت بیش‌تری دارد، نیز مناسب است. **اغلب LANها** در لایه پیوند داده خود از سرویس بدون اتصال بدون تصدیق استفاده می‌کنند.

مرحله دیگر در سلسله مراتب قابلیت اطمینان، **سرویس بدون اتصال با تصدیق** است. در این سرویس برای هر فریم ارسالی، تصدیق باید به صورت جداگانه دریافت شود. به این وسیله فرستنده متوجه دریافت یا دریافت نشدن فریم‌های ارسالی می‌شود. اگر پیام تصدیق در بازه زمانی مشخصی دریافت نشود، فرستنده مجدداً فریم قبلی را ارسال می‌کند. **این سرویس روی کانال‌های غیر قابل اطمینان مانند سیستم‌های بی‌سیم مفید است. بدیهی است که نرخ ارسال داده در این جا از سرویس بدون اتصال بدون تصدیق کمتر است.**

نکته۱: قابلیت تصدیق در لایه پیوند داده، اختیاری است و نه الزامی. چراکه لایه انتقال می‌تواند همیشه پیغامی را بفرستد و برای تصدیق آن منتظر بماند.

اگر تصدیق قبل از اتمام زمان از پیش تعیین شده، دریافت نشود و اصطلاحاً **time out** اتفاق بیفتد، فرستنده بار دیگر پیغام را می‌فرستد. مشکل این رویکرد این است که اگر فرض کنیم کل پیغام به 10 فریم تقسیم شده باشد و 20 درصد فریم‌ها گم شوند، زمان بسیار زیادی برای ارسال پیغام لازم است.

مهم‌ترین سرویسی که لایه پیوند داده برای لایه شبکه فراهم می‌آورد، **سرویس اتصال گرای با تصدیق** است. علاوه بر ارائه تضمین دریافت هر فریم، این سرویس به ترتیب دریافت شدن فریم‌ها را نیز تضمین می‌کند. در حالی که در سرویس بدون اتصال، ممکن است یک تصدیقِ گم شده، باعث ارسال چندین باره یک فریم شده و بدین ترتیب چند نسخه از آن فریم دریافت شود.

زمانی که سرویس اتصال‌گرا استفاده می‌شود، انتقال شامل سه مرحله جداگانه می‌شود. در اولین مرحله، ارتباطی بین دو طرف انتقال ایجاد می‌شود که در آن مقدارها و شمارش گره‌های مورد نیاز برای تعیین و پی‌گیری اثر (track) هر فریم دریافتی و یا دریافت نشده، مقداردهی اولیه می‌شود. در دومین مرحله، ارتباط اصلی شکل گرفته و داده‌ها مبادله می‌شوند. مرحله سوم نیز خاتمه ارتباط است.

نکته۲: نرخ ارسال داده در سرویس اتصال‌گرای با تصدیق، کم‌تر از سرویس بدون اتصال بدون تصدیق و سرویس بدون اتصال با تصدیق می‌باشد.

تذکر۱: در فصل اول در رابطه با سرویس‌های اتصال‌گرا و بدون اتصال توضیح داده شده است.

مثال۱: کدام گزینه صحیح نیست؟

۱) اغلب LANها در لایه پیوند داده خود از سرویس بدون اتصال بدون تصدیق استفاده می‌کنند.

۲) قابلیت تصدیق در لایه پیوند داده، الزامی است.

۳) سرویس بدون اتصال با تصدیق، روی کانال‌های غیر قابل اطمینان مانند سیستم‌های بی‌سیم مفید است.

۴) سرویس بدون اتصال بدون تصدیق زمانی اهمیت پیدا می‌کند که نرخ خطا بسیار پایین باشد.

☑ **پاسخ:** گزینه «۲» قابلیت تصدیق در لایه پیوند داده، اختیاری است و نه الزامی. با مراجعه به متن درس، درستی گزینه‌های دیگر مشهود است.

روش‌های فریم‌بندی

همان‌طور که پیش از این نیز اشاره شد، منظور از فریم‌بندی، ارسال داده‌ها در یک قالب مشخص و معین برای فرستنده و گیرنده است به طوری که ابتدا و پایان آن‌ها نیز مرزبندی شده باشد. قبل از آغاز بحث تکنیک‌های مختلف فریم بندی، لازم است به دو موضوع مهم اشاره کنیم:

۱ـ در بسیاری از شبکه‌ها، مخصوصاً در WANها، به هدف رسیدن به سطح بالاتری از قابلیت اطمینان و کاراییِ لایه کنترل پیوند داده، اندازه فریم کوچکتر از اندازه بسته انتخاب می‌شود. بنابراین بسته‌ای که قرار است ارسال شود، اغلب اوقات با فریم‌بندی و حدبندی مناسب، به چندین فریم تقسیم می‌شود و سپس به سمت مقصد ارسال می‌شود. از آن طرف، گیرنده نیز باید بتواند با چینش مناسب فریم‌ها، قادر به ادغام آن‌ها و تولید بسته اولیه باشد.

۲ـ بحث فریم‌بندی که در ادامه خواهیم دید، در ارتباط با انتقال سنکرون است (در رابطه با انتقال سنکرون و آسنکرون در همین بخش مفصل صحبت خواهیم کرد).

۳ـ عمل همگام‌سازی که در لایه فیزیکی انجام می‌شود محدود به یک بیت و یا یک کاراکتر است در حالی که انجام همین عمل در لایه پیوند داده، برای همگام‌سازی بلوکی از داده‌ها می‌تواند صورت گیرد.

بی‌کار (idle)	جدا کننده پایانی	دنباله (Trailer)	INFO	سرآیند (Header)	جدا کننده آغازین	بی‌کار (idle)

فریم

شکل ۲: فرمت کلی یک فریم

فرمت کلی یک فریم در شکل ۲ نشان داده شده است. در این شکل، فیلد INFO که طول متغیری دارد، حاوی کل بسته یا قسمتی از بسته است که از لایه شبکه به لایه پیوند داده ارسال شده است. موضوع این قسمت، تکنیک‌های مختلف برای فریم بندی یعنی نحوه انتخاب جداساز (delimiter) می‌باشد.

برای فریم‌بندی، معمولاً از یکی از سه روش زیر استفاده می‌شود:

۱ـ فریم‌بندی کاراکترگرا (Character-Oriented Framing)

۲ـ فریم‌بندی بیت‌گرا (Bit -Oriented Framing)

۳ـ فریم‌بندی تخطیِ کدگرا (Code violation -Oriented Framing)

فریم‌بندی کاراکترگرا (Character-Oriented Framing)

این روش یکی از قدیمی‌ترین الگوها برای جداسازی است. فریم‌بندی کاراکترگرا از چهار کاراکتر کنترلی در کد ASCII برای فریم بندی استفاده می‌کند: وضعیت بیکار لینک یا SYN، آغاز متن یا STX (Start of Text)، پایان متن یا ETX (End of Text) و گریز پیوند داده یا DLE (Data Link Escape).

فرض کنید قرار است دو کاراکتر مستقل "MAY" و "2000" با فاصله زمانی مشخص، پشت سر هم ارسال شود. داده ارسالی که عملاً توسط لایه کنترل لایه پیوند داده (DLC) ارسال می‌شود به صورت زیر است:

SYN SYN STX MAY ETX SYN SYN STX 2000 ETX SYN SYN

کاراکتر کنترلی SYN به منزله بیکار بودن (idle) لینک است و معمولاً به‌صورت 01010101 در نظر گرفته می‌شود. وظیفه این کاراکتر کنترلی، همگام‌سازی فرستنده و گیرنده است و در حقیقت به‌عنوان کلاک مشترک بین آن دو عمل می‌کند. بنابراین گیرنده در هر زمان می‌تواند با جستجوی SYN در رشته داده، آغاز داده را شناسایی کند. مشکلی که اینجا ممکن است به وجود آید این است که خود داده‌ای که قرار است ارسال شود، حاوی کاراکترهای کنترلی نیز باشد. در این صورت چگونه می‌توان فهمید که مثلاً کاراکتری مانند SYN، نقش کنترلی دارد یا واقعاً قسمتی از داده است؟

یک روش که **character stuffing** نام دارد به این صورت است که جداکننده‌هایی که در متن داده ظاهر می‌شوند را با جفت کاراکتر کنترلی DLE STX و DLE ETX نشان دهیم. ضمناً اگر کاراکتر کنترلی DLE نیز بخواهد در متن داده قرار گیرد، کافی است قبل از آن، یک DLE دیگر اضافه کنیم. گیرنده نیز به سادگی در صورت مشاهده دو DLE پشت سر هم، اولین DLE را حذف می‌کند. فرض کنید قرار است داده زیر را منتقل نماییم:

"x y DLE z STX"

در صورتی که character stuffing برای ارسال این داده استفاده شود، نتیجه کار در شکل ۳ نشان داده شده است.

SYN	SYN	STX	x	y	DLE	DLE	z	DLE	STX	ETX	SYN	SYN

شکل ۳: مثالی از character stuffing

نکته ۳: سربار از رابطه زیر محاسبه می‌شود:

$$\text{سربار} = \frac{\text{تعداد بیت غیر مفید ارسالی (داده‌هایی که جزو اصل داده نیستند اما ارسال می‌شوند)}}{\text{تعداد کل بیت ارسالی}}$$

برای بیان درصد سربار، کافی است کسر بالا را در 100 ضرب نماییم.

مثال ۲: یک منبع داده‌ها کاراکترهای ASCII هفت بیتی تولید و از طریق یک سیستم انتقال سنکرون با سرعت 300bps ارسال می‌کند. انتقال به صورت کاراکترگرا بوده و هر فریم از 8 کاراکتر کنترلی و 120 کاراکتر اطلاعات تشکیل شده است. چنانچه به همراه هر کاراکتر یک بیت پریتی در کاراکترها اضافه شود، مقدار کاراکترهای ارسالی در ثانیه (گذردهی) چقدر است؟ (برحسب کاراکتر در ثانیه)

۱) 30 ۲) 36 ۳) 24 ۴) 42

☑ **پاسخ:** گزینه «۲» گذردهی در این حالت برابر است با تعداد کاراکترهای ارسالی در هر فریم تقسیم بر مدت زمان لازم برای ارسال یک فریم (t). از آنجا که هر کاراکتر معادل 7 بیت به علاوه یک بیت پریتی است. تعداد بیت‌های هر فریم (N) برابر است با:

$$N = (120+8)\times(7+1) = 2^{10} = 1024\,\text{bit}$$

برای محاسبه t از تناسب زیر استفاده می‌کنیم:

$$\frac{\text{بیت}}{\text{ثانیه}} : \frac{300}{1} = \frac{2^{10}}{t} \Rightarrow t = \frac{2^{10}}{300}$$

بنابراین گذردهی (TP) برابر می‌شود با:

$$TP = \frac{128}{\frac{2^{10}}{300}} = 37.5 \simeq 36$$

فریم‌بندی بیت گرا (Bit -Oriented Framing)

در فریم‌بندی بیت‌گرا که در پروتکل HDLC استفاده می‌شود، جداساز d، توسط الگوهای بیتی خاصِ "01111110" که **پرچم (فلگ)** نام دارند، انجام می‌شود. در اینجا نیز ممکن است این جداساز در مواقعی جزو داده باشد. در این صورت باید از روش **bit stuffing** استفاده کرد. در این روش، هر گاه فرستنده پنج رشته "1" پشت سر هم را در جریان داده ببیند، یک عدد صفر به رشته اضافه می‌کند. از آن طرف، گیرنده نیز هر جا پس از پنج عدد یکِ متوالی، یک صفر ببیند متوجه می‌شود که این صفر را فرستنده اضافه کرده و آن را حذف می‌کند.

به این نکته دقت کنید که **سربار (overhead) روش بیت‌گرا نسبت به روش کاراکترگرا کمتر است** (منظور از سربار، اطلاعاتی است که جزو داده اصلی نیستند اما بنا بر دلایل مختلف ـ مانند دلایل کنترلی ـ به اجبار باید آن‌ها را نیز همراه داده اصلی انتقال داد). ضمن اینکه در این روش به جای محدود سازی فیلد داده به کاراکترهای 8 بیتی، امکان حضور هر تعداد بیت در فیلد داده، میسر است. هر چند مطابق روال کلی، داده به شکل بایت سازمان‌دهی می‌شود.

نکته۴: تعداد داده‌های ارسالی در روش فریم‌بندی بیت‌گرا مضربی از بایت نمی‌باشد (چرا؟).

مثال۳: فرض کنید در یک سیستمی که از فریم‌بندی بیت‌گرا استفاده می‌کند، گیرنده پیغام 011111100111110010111111O را از فرستنده دریافت می‌کند. در این صورت اصل داده ارسالی چه بوده است؟

۱) 01110101 ۲) 01111110 ۳) 01111001 ۴) 01111101

☑ **پاسخ:** گزینه «۴» گیرنده هر جا پس از پنج عدد یکِ متوالی، یک صفر ببیند متوجه می‌شود که این صفر را فرستنده اضافه کرده و آن را حذف می‌کند. کاراکترهای 01111110 نیز که در ابتدا و انتها آمده است اشاره به ابتدا و انتهای کاراکتر دارند.

فریم‌بندی تخطیِ کدگرا (Code violation -Oriented Framing)

به جای آن که از کاراکترها یا الگوهای بیتی خاص استفاده کنیم، می‌توان عمل فریم‌بندی را در سطح واقعی بیت، در هنگام رمزگذاری و کدینگ بیت‌ها و تبدیل آن‌ها به سیگنال انجام داد. اجازه دهید با یک مثال توضیح بیش‌تری بدهیم. در کدینگ منچستر (که در فصل گذشته آنرا مطالعه کردیم)، هر گذار high به low معرف داده "1" و گذارlow به high معرف داده "0" است. ویژگی اصلی این کد این است که همیشه در وسطِ زمان هر بیت، باید یک گذار اتفاق بیفتد. با تخطی از همین ویژگی (به‌عنوان مثال وجود یک زوج high-high به دنبال یک زوجlow-low)، می‌توان به تعیین جداکننده پرداخت. این الگوی ابتکاری از گذارهای مورد انتظار در وسط زمان هر بیت جلوگیری کرده و می‌تواند مرز و حد فریم را به اطلاع گیرنده برساند. این تکنیک فریم‌بندی که از یک کدینگِ لایه فیزیکیِ غیر معتبر استفاده می‌کند در استانداردهای 802 LAN کاربرد دارد.

قبل از آنکه این بخش را به پایان ببریم باید این نکته را نیز متذکر شویم که از ارسال تعداد کاراکترها یا بیت‌ها در هر فریم ارسالی نیز می‌توان برای آگاه‌سازی گیرنده از انتهای فریم استفاده کرد. این تعداد می‌تواند در سرآیند فریم (frame header) قرار گیرد. به هر حال این روش چندان کاربرد ندارد چراکه خطای بیتی می‌تواند منجر به اشتباهات بزرگ و شدیدی شود. البته اگر این روش به همراه 3 روش قبلی که پیش از این توضیح داده شد استفاده شود، می‌تواند قابلیت اطمینان پروسه فریم‌بندی را تا حد بسیار زیادی ارتقا بخشد. استفاده همزمان از این روش‌های فریم‌بندی در پروتکل‌هایی مانند Token Ring و Ethernet کاربرد دارد.

تذکر۲: پروتکل‌هایی مانند Token Ring و Ethernet در فصل چهارم بررسی خواهند شد.

بحث فریم‌بندی را با بررسی سه روش انتقال سنکرون (همگام) و غیر سنکرون (غیر همگام) و متقارن (Isochronous) به پایان می‌بریم.

مثال۴: کدام گزینه صحیح است؟

۱) فریم بندی تخطیِ کدگرا از قدیمی‌ترین روش‌های فریم بندی محسوب می‌شود.
۲) سربار (overhead) روش بیت‌گرا نسبت به روش کاراکترگرا بیش‌تر است.
۳) ارسال تعداد کاراکترها یا بیت‌ها در هر فریم ارسالی یکی از مطمئن‌ترین روش‌های فریم بندی است.
۴) فریم بندی تخطیِ کدگرا در استانداردهای 802 LAN کاربرد دارد.

☑ **پاسخ:** گزینه «۴» فریم بندی کاراکترگرا از قدیمی‌ترین الگوها برای جداسازی بسته‌های شامل بسته داده است (نادرستی گزینه ۱). سربار (overhead) روش بیت‌گرا نسبت به روش کاراکترگرا کمتر است (نادرستی گزینه ۲). در ارسال تعداد کاراکترها یا بیت‌ها در هر فریم ارسالی به هدف فریم بندی، خطای بیتی می‌تواند منجر به اشتباهات بزرگ و شدیدی شود (نادرستی گزینه ۳).

انتقال سنکرون و آسنکرون

به طور کلی سه حالت اصلی انتقال عبارتند از:
۱ـ انتقال آسنکرون (غیر همگام-Asynchronous) ۲ـ انتقال سنکرون (همگام-Synchronous) ۳ـ انتقال Isochronous

در این کتاب ما به طور کلی انتقال سریال را مورد توجه قرار خواهیم داد که طبق آن، داده‌ها پشت سر هم و از طریق تنها یک کانال ـ به جای چندین خط موازی با هم ـ ارسال می‌شوند. انتقال موازی بیش‌تر در دستگاه‌های I/O و مسیرهای درونی کامپیوترها کاربرد دارد. با استفاده از انتقال سریال، در هر نوبت المان‌های سیگنال از طریق خط ارسال می‌شوند.

هر المان سیگنال ـ همان‌طور که در فصل گذشته دیدیم ـ ممکن است:

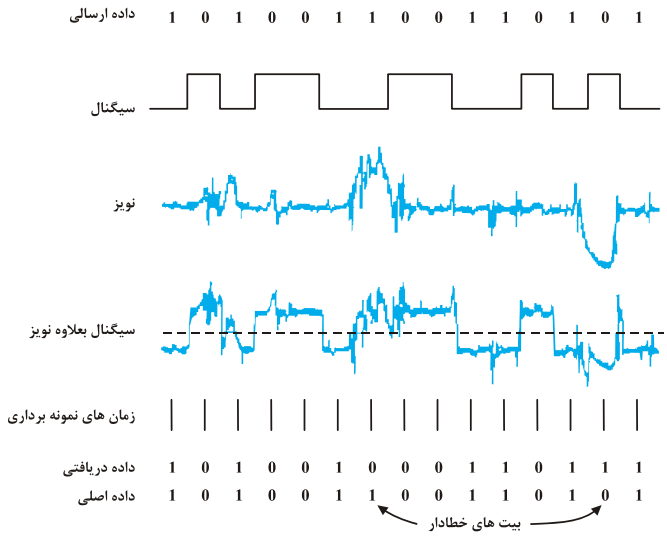
ـ کم‌تر از یک بیت باشد. مانند آنچه در کد منچستر رخ می‌دهد.

ـ برابر با یک بیت باشد. مثال‌های دیجیتال و آنالوگ از این حالت به ترتیب عبارتند از: NRZ-L و FSK.

ـ بیش‌تر از یک بیت باشد. مانند QPSK

برای سادگی بحث، ما حالت دوم را در نظر می‌گیریم؛ یعنی هر المان سیگنال معادل یک بیت باشد. البته ماهیت قضیه با در نظر گرفتن این فرض، خدشه‌دار نخواهد شد.

به شکل ۴ دقت کنید. در این شکل هر دریافت داده دیجیتال، شامل نمونه‌برداری از سیگنال‌های ورودی به ازای هر زمان بیت است تا مقدار باینری آن مشخص شود. یکی از مشکلاتی که در فصول گذشته نیز کاملاً آن را مورد بررسی قرار دادیم، این است که امکان رخداد انواع نویز روی خط و سیگنال وجود دارد که می‌تواند تولید خطا کند. این مسئله برای زمان‌بندی نیز می‌تواند باعث مشکلاتی شود. برای آنکه گیرنده بتواند نمونه‌برداری بیت‌های دریافتی را به شکل صحیح انجام دهد، باید از زمان ورود و دوره‌ی زمانی (duration) هر بیتی که دریافت می‌کند، آگاه باشد.



شکل ۴: اثر نویز بر سیگنال دیجیتال

فرض کنید فرستنده‌ای که جریانی از بیت‌های داده را ارسال می‌کند، دارای کلاکی است که زمانبندی ارسال بیت‌ها را کنترل و مدیریت می‌کند. برای مثال، اگر قرار باشد داده با سرعت یک میلیون بیت در ثانیه (1 Mbps) ارسال شود، آنگاه یک بیت باید در هر میکروثانیه ($\frac{1}{10^6}$ ثانیه) که توسط کلاک فرستنده محاسبه می‌شود، ارسال شود (می‌توانید با یک تناسب ساده صحت این ادعا را بررسی نمایید). معمولاً گیرنده سعی می‌کند تا نمونه‌برداری را در وسط زمان هر بیت انجام دهد. گیرنده، زمان‌بندی خود را با فاصله‌های زمانی هر بیت تنظیم می‌کند. در مثال ما، نمونه‌برداری باید در هر یک میکرو ثانیه انجام شود.

اگر گیرنده زمان‌بندی خود را با توجه به کلاک خاص خودش انجام دهد، آنگاه چنانچه کلاک گیرنده و فرستنده دقیقاً با هم تنظیم نشوند، احتمال بروز مشکل وجود خواهد داشت. اگر این اختلاف تنها در حدود 1% هم باشد (کلاک گیرنده 1% سریعتر یا کندتر از کلاک فرستنده باشد)، آنگاه اولین نمونه‌برداری، 0.01 زمانِ بیت (0.01μs)، از مرکز بیت فاصله خواهد گرفت (مرکز بیت، به اندازه 0.5μs نسبت به آغاز و انتهای بیت قرار گرفته است). پس از 50 مورد نمونه‌برداری یا تعدادی بیش‌تر، گیرنده ممکن است با خطا مواجه شود؛ چراکه نمونه‌برداری در زمان اشتباهی صورت می‌گیرد. $(50\times 0.01=0.5\mu s)$ برای اختلاف زمان‌بندی‌های کمتر، ممکن است خطا در زمان دیرتری رخ دهد اما به هر حال، در نهایت اگر فرستنده جریان بزرگی از داده را ارسال کند و در این بین، عمل همگام سازی بین فرستنده و گیرنده انجام نشود، گیرنده، همزمانی خود را با فرستنده از دست خواهد داد.

نکته۵: اگر سرعت نمونه‌برداری در فرستنده را با T، درصد اختلاف ساعت گیرنده و فرستنده را با ΔT و حداکثر تعداد نمونه‌برداری که در آن خطایی به وجود نیاید را با n نشان دهیم آنگاه می‌توان نوشت:

$$n \le \frac{T}{2\Delta T}$$

مثال۵: اگر سرعت نمونه‌برداری در فرستنده برابر یک میکروثانیه بوده و اختلاف کلاک گیرنده با فرستنده دو درصد باشد، حداکثر تعداد نمونه برداری که در آن خطایی به وجود نیاید چقدر است؟

۱) 50 ۲) 100 ۳) 25 ۴) 200

☑ **پاسخ:** گزینه «۳»

$$n \le \frac{T}{2\Delta T} \rightarrow n \le \frac{1}{2\times 0.02} = 25$$

انتقال آسنکرون (غیر همگام ــ Asynchronous)

رویکردهای مختلفی برای رسیدن به همگامی و همزمانی مورد نظر، رواج دارد. اولین روش انتقال آسنکرون نامیده می‌شود. اساس این روش این است که برای پیشگیری از بروز مشکل در زمان‌بندی، جریان بزرگی از داده‌ها بدون وقفه ارسال نشود. بلکه در هر نوبت، داده‌ها به صورت «تک کاراکتر، تک کاراکتر» ارسال می‌شوند (هر کاراکتر شامل 5 تا 8 بیت است). تعداد بیت‌های هر کاراکتر، بستگی به کد استفاده شده دارد. به‌عنوان مثال در کد IRA، هر کاراکتر، هفت بیتی است. در کد دیگری به نام Extended Binary Coded Decimal Interchange Code (EBCDIC) که در ابر کامپیوتر (mainframe)های IBM کاربرد دارد، هر کاراکتر شامل 8 بیت است. زمان‌بندی و همگام‌سازی باید در هر کاراکتر حفظ شود. به این صورت، گیرنده در ابتدای هر کاراکترِ جدید، فرصت همزمان سازی مجدد را خواهد داشت. مثالی از ارتباط آسنکرون، نوع ارتباط بین کامپیوتر و صفحه کلید است.

آدرس‌دهی (Addressing)

زمانی که یک پردازشِ اپلیکیشن (کاربرد) می‌خواهد ارتباطی را با یک اپلیکیشن راه دور دیگر برقرار کند، باید مشخص شود که کدام آدرس انتقال می‌خواهد به کدام پردازش متصل شود. در اینترنت این نقاط انتهایی، زوج آدرس‌های IP و پورت‌های محلی می‌باشند. در شبکه‌های ATM، آن‌ها AAL-SAPS نامیده می‌شوند. ما از اصطلاح **TSAP** (**Transport Service Access Point**) استفاده می‌کنیم. نقاط انتهایی متشابه در لایه شبکه، **NSAP** (**Network Service Access Point**) نام دارند. آدرس‌های IP که در فصل قبل با آن‌ها آشنا شدیم، مثالی از NSAP ها هستند.

یک سناریوی ممکن برای ارتباط انتقال روی یک لایه شبکه اتصال‌گرا با توجه به شکل ۲، به قرار زیر است:

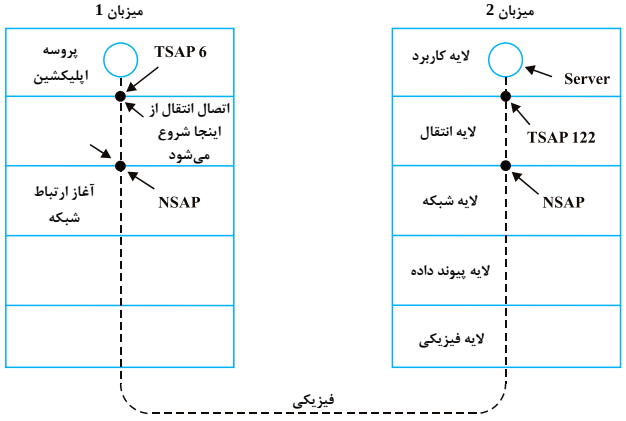
۱ـ پروسه‌ی سرور در میزبان شماره 2، با اتصال به TSAP 122، برای یک تماس دریافتی به انتظار می‌نشیند. چگونگی اتصال پروسه به TSAP، خارج از بحث مدل شبکه است و کاملاً وابسته به سیستم عامل می‌باشد. برای مثال تماسی مانند LISTEN، می‌تواند استفاده شود.

۲ـ یک پروسه اپلیکیشنِ خواهان در میزبان 1، درخواست CONNECT را با مشخص کردن TSAP 6 به‌عنوان مبدا و TSAP 122 به‌عنوان مقصد، صادر می‌کند.

۳ـ موجودیت انتقال در میزبان 1، آدرس شبکه‌ای را روی ماشین خود انتخاب کرده و اتصال شبکه‌ای را ایجاد می‌کند. با استفاده از این اتصال شبکه، موجودیت انتقال میزبان 1 می‌تواند با موجودیت انتقال 2 ارتباط برقرار کند.

۴ـ اولین چیزی که موجودیت انتقال میزبان 1 به نظیر خود در میزبان 2 می‌گوید، این است: «سلام! من می‌خواهم ارتباطی بین TSAP 6 خودم و TSAP 122 متعلق به تو ایجاد کنم؛ نظر تو چیست؟»

۵ـ پس از این، موجودیت انتقالِ میزبان 2 از سرور موجود در TSAP 122 جویا می‌شود که آیا مایل به پذیرش ارتباط جدیدی می‌باشد یا خیر. اگر موافقت لازم صورت گیرد، ارتباطِ انتقال ایجاد می‌شود.



شکل ۲: TSAP، NSAP و ارتباط شبکه

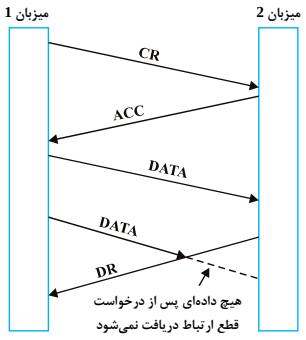
ایجاد یک ارتباط (Establishing a Connections)

ایجاد یک ارتباط به نظر ساده می‌رسد اما در عمل باید دانست که این کار چندان هم ساده نیست. در نگاه اول شاید این طور به نظر رسد که تنها کافی است موجودیت انتقال، یک REQUEST TPDU CONNECTION را به مقصد ارسال کند و منتظر پاسخ CONNECTION ACCEPTED بماند. مشکل زمانی رخ می‌دهد که در شبکه احتمال گم شدن، ذخیره شدن و تکراری شدن بسته‌ها (duplicate packets) وجود داشته باشد.

در بخش‌های بعدی (پروتکل TCP) بیشتر در خصوص ایجاد ارتباط صحبت خواهیم کرد.

آزادسازی یک ارتباط (Releasing a Connection)

آزادسازی ارتباط از ایجاد آن ساده‌تر است. با این حال حتی این کار هم پیچیده‌تر از آن چیزی است که به نظر می‌رسد. همانطور که پیش از این نیز اشاره شد، دو روش برای خاتمه یک ارتباط وجود دارد: آزادسازی غیرمتقارن (Asymmetric) و متقارن (Symmetric). یک مثال از **آزاد سازی غیرمتقارن**، تماس تلفنی است. زمانی که یکی از طرفین به تماس خاتمه می‌دهد، ارتباط قطع می‌شود. در **آزادسازی متقارن**، به هر ارتباط، مانند دو ارتباط تک جهته‌ی مجزا از هم نگاه می‌شود که برای قطع آن، باید هر دو طرف به طور مستقل اقدام به قطع ارتباط نمایند.



شکل ۳: قطع ناگهانی ارتباط و از دست رفتن داده در آزادسازی غیرمتقارن

نکته۴: از آنجا که قطع ارتباط غیرمتقارن به طور ناگهانی صورت می‌گیرد، می‌تواند منجر به از دست رفتن داده شود.

سناریویی را که در شکل ۳ نشان داده شده است، در نظر بگیرید. پس از برقراری ارتباط، میزبان 1، یک TPDU ارسال می‌کند که به طرز صحیح توسط میزبان 2 دریافت می‌شود. سپس میزبان 1، TPDU دیگری را ارسال می‌کند. متاسفانه میزبان 2، قبل از آنکه TPDU ارسال شده را دریافت کند، پیام DISCONNECT را صادر می‌کند. نتیجه این امر، قطع ارتباط و از دست رفتن داده خواهد بود.

واضح است که برای افزایش قابلیت اطمینان و جلوگیری از، از دست رفتن داده به پروتکل آزادسازی پیشرفته‌تری نیاز است. یک روش، استفاده از آزادسازی متقارن است که در آن هر طرف، مستقل از طرف دیگر، روند آزادسازی را انجام می‌دهد. در این حالت، میزبان حتی پس از ارسال DISCONNECT TPDU، می‌تواند به روند واگذاری داده ادامه دهد.

نکته۵: روش متقارن، زمانی می‌تواند کار خود را انجام دهد که هر پردازش، میزان داده مشخصی برای ارسال داشته و زمان ارسال نیز مشخص باشد.

در موقعیت‌های دیگر، تعیین اینکه کلیه کارها انجام شوند و ارتباط خاتمه یابد، کاملاً مشخص و واضح نیست. برای درک بهتر از پروتکل می‌توان این طور فرض کرد که میزبان می‌گوید:«من کار خود را تمام کردم؛ آیا تو هم کار خود را به پایان رساندی؟» اگر پاسخ میزبان ۲ این‌گونه باشد که: «من هم انجام دادم. به درود!»، ارتباط به شکل مطمئنی خاتمه می‌یابد.

مثال۶: در مورد انواع روش‌های آزادسازی ارتباط، کدام یک صحیح نیست؟

۱) در روش متقارن، زمانیکه یک طرف DISCONNECT می‌کند، معنی‌اش آن است که داده دیگری برای ارسال ندارد اما همچنان به دریافت داده از طرف دیگر ادامه می‌دهد.

۲) استفاده از روش غیرمتقارن می‌تواند منجر به از دست رفتن داده شود.

۳) در آزادسازی متقارن، به هر ارتباط، مانند دو ارتباط تک جهته‌ی مجزا از هم نگاه می‌شود که برای قطع آن باید هر دو طرف به طور مستقل ارتباط را قطع نمایند.

۴) یک مثال از آزاد سازی متقارن، تماس تلفنی است.

☑ **پاسخ:** گزینه «۴» تماس تلفنی مثالی از آزاد سازی غیرمتقارن است. سایر گزینه‌ها با توجه به متن درس صحیح هستند.

کنترل جریان و بافرینگ

تا اینجا ما نحوه ایجاد و خاتمه ارتباط را با یکدیگر بررسی کردیم. در این بخش می‌خواهیم روند مدیریت ارتباط را در زمان استفاده از آن مورد مطالعه قرار دهیم. یکی از مهمترین مسائلی که در فصول گذشته هم به آن اشاره داشتیم، کنترل جریان است. از برخی جهات، مسئله کنترل جریان در لایه انتقال، مشابه کنترل جریان در لایه پیوند داده است. اما از جهاتی دیگر، تفاوت‌هایی نیز وجود دارد. تشابه اصلی در این است که هر دو از پنجره لغزان یا سایر الگوهای دیگر استفاده می‌کنند تا سرعت فرستنده با سرعت گیرنده همخوانی داشته باشد. مهم‌ترین تفاوت نیز آن است که مسیریاب معمولاً تعداد خطوط محدودی دارد در حالیکه یک میزبان ممکن است تعداد زیادی ارتباط داشته باشد.

در پروتکل پیوند داده، فریم‌ها در هر دو مسیریاب فرستنده و گیرنده بافر می‌شوند. علت بافر شدن در طرف فرستنده این است که شاید بعداً نیاز به ارسال مجدد بسته‌هایی باشد که قبلاً ارسال شده باشند. چنانچه زیر شبکه، سرویس داده‌گرام را تامین می‌کند، موجودیتِ انتقالِ فرستنده نیز به دلیل مشابه باید عملیات بافرینگ را انجام دهد. اگر گیرنده بداند که فرستنده حتی واحدهای TPDU ای را که تصدیقشان نیز دریافت شده، بافر می‌کند، دیگر می‌تواند عملیات بافرینگ را انجام ندهد.

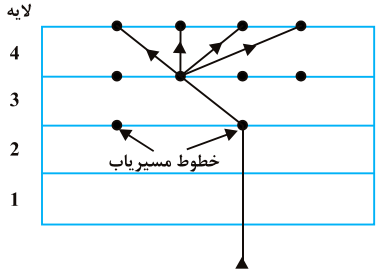
مالتی پلکسینگ

مالتی پلکس نمودن چندین محاوره(conversation) روی مدارهای مجازیِ ارتباطی و لینک‌های فیزیکی، نقش مهمی در لایه‌های مختلف معماری شبکه ایفا می‌کنند. در لایه انتقال، احتیاج به مالتی پلکسینگ بیش‌تر می‌تواند به چشم آید. اصولاً دو نوع مالتی پلکس در این جا قابل تصور است: مالتی پلکسینگ رو به بالا و مالتی پلکسینگ رو به پایین.

مالتی پلکسینگ رو به بالا(Upward Multiplexing)

هزینه بالا و سنگین داشتن تعداد زیادی مدار مجازی آماده برای مدت زمان طولانی، اهمیت مالتی پلکسینگ ارتباطات انتقال مختلف به درون ارتباط شبکه را برجسته می‌سازد. این فرم از مالتی پلکسینگ که در شکل ۴ نشان داده شده است، مالتی پلکسینگ رو به بالا نامیده می‌شود.

در این شکل، چهار ارتباط انتقالِ مستقل، از ارتباط شبکه مشابهی استفاده می‌کنند. اگر تعداد ارتباطِ انتقالِ زیادی به یک ارتباط شبکه نگاشت شود، کارایی کاهش خواهد یافت. چراکه پنجره معمولاً پر است و کاربران باید برای آنکه نوبت ارسالشان شود، منتظر بمانند.

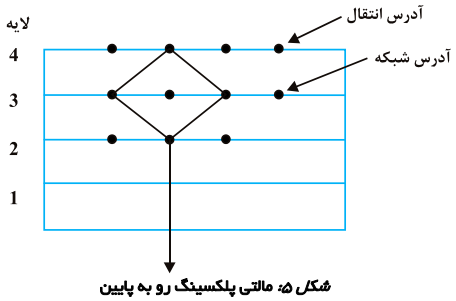


شکل۴: مالتی پلکسینگ رو به بالا

تذکر۱: گاهی اوقات به مالتی پلکسینگ رو به بالا، Demultiplexing و یا ارتباط One-to-Many نیز گفته می‌شود.

مالتی پلکسینگ رو به پایین(Upward Multiplexing)

عمل مالتی پلکسینگ می‌تواند برای لایه انتقال به شکل دیگری نیز در نظر گرفته شود. به این نحو که یک ارتباطِ انتقالِ فعال، می‌تواند با چندین ارتباط شبکه در ارتباط باشد و ترافیک را مابین آن‌ها با استفاده از الگوی round robin تقسیم کند (شکل ۵). این گونه مالتی پلکسینگ، مالتی پلکسینگ رو به پایین نامیده می‌شود.



شکل ۵: مالتی پلکسینگ رو به پایین

تذکر۲: به مالتی پلکسینگ رو به پایین، ارتباط Many-to-one نیز گفته می‌شود.

نکته زیر اهمیت استفاده از مالتی پلکسینگ رو به پایین را بیشتر نمایان می‌کند.

نکته۶: با داشتن K ارتباط شبکه باز (open) و آماده، کارایی پهنای باند با استفاده از مالتی پلکسینگ رو به پایین، با ضریب K افزایش پیدا می‌کند.

مثال۷: فرض کنید از روش مالتی پلکسینگ رو به پایین در یک شبکه که در لایه شبکه خود از مدار مجازی استفاده می‌کند، بهره برده‌ایم. چنان‌چه 16 مدار مجازی باز وجود داشته باشد و ظرفیت هر مدار مجازی برای هر کاربر 32kbps باشد، حداکثر نرخ ارسالی که می‌توان با توجه به این شرایط برای کاربر فراهم آورد چقدر است؟

۱) 32kpbs ۲) 128 kpbs ۳) 512 kpbs ۴) هیچ‌کدام

☑ **پاسخ:** گزینه «۳» یکی از کاربردهای مالتی پلکسینگ رو به پایین در شبکه‌هایی است که از مدار مجازی استفاده می‌کنند. با توجه به نکته فوق، حداکثر نرخ ارسال می‌تواند به 32 kpbs ×16=512 kpbs برسد.

مثال۸: کدام گزینه در مورد مالتی پلکسینگ رو به پایین صحیح است؟

۱) شماره پورت مبدأ داده باید منحصر به فرد باشد. ۲) نیازی نیست شماره پورت مقصد داده منحصر به فرد باشد.
۳) هر دو ۴) هیچ‌کدام

☑ **پاسخ:** گزینه «۳» در هنگام مالتی پلکسینگ کردن رو به پایین باید برای هر پروسس مبدأ شماره پورت جدایی در نظر گرفته شود تا از آن طرف در سمت گیرنده امکان بازیابی و دی مالتی پلکسینگ کردن داده‌ها وجود داشته باشد.

مثال۹: کدام یک از گزاره‌های زیر در مورد انواع مالتی پلکسینگ صحیح می‌باشد؟

الف) در مالتی پلکسینگ رو به پایین، با داشتن K ارتباط شبکه باز(open) و آماده، کارایی پهنای باند با ضریب K افزایش پیدا می‌کند.

ب)در مالتی پلکسینگ رو به بالا، اگر تعداد ارتباط انتقال زیادی به یک ارتباط شبکه نگاشت شود، کارایی کاهش خواهد یافت.

۱) فقط الف ۲) فقط ب ۳) هر دو ۴) هیچکدام

☑ **پاسخ:** گزینه «۳» با توجه به توضیحات داده شده، هر دو گزاره صحیح هستند.

بازیابی از خطا (Crash Recovery)

چنانچه میزبان‌ها و مسیریاب‌ها در معرض خطا و یا صدمه باشند، انجام عملیات بازیابی بااهمیت می‌شود. اگر موجودیت انتقال کاملاً در میزبان باشد، بازیابی از خطای پیش آمده برای شبکه و مسیریاب، کار ساده‌ای خواهد بود.

همان‌طور که حتماً با مطالعه کتاب تا اینجا متوجه شده‌اید، چنانچه خطا در یک لایه‌ای اتفاق بیفتد به طوری که عملکرد آن لایه را دچار اخلال کند، اگر لایه بالاتر از آن اطلاعات کافی را در دسترس داشته باشد، می‌تواند شرایط را سر و سامان دهد.

زمانی که لایه شبکه، سرویس داده‌گرام را مهیا می‌سازد، موجودیت انتقال در تمام زمان، انتظار گم شدن TPDUها را دارد. همچنین زمانی که لایه شبکه، سرویس اتصال‌گرا را فراهم آورده، تلفات یک مدار مجازی، توسط ایجاد یک جایگزین جدید و پرسش از موجودیت انتقال راه دور در خصوص اینکه کدام TPDU دریافت شده و کدام یک دریافت نشده، انجام می شود. واحد TPDU دریافت نشده، می‌تواند مجدداً ارسال شود.

یک مشکل پردردسر، چگونگی بازیابی از خطای میزبان است. برای درک بیش‌تر این مشکل، میزبانی را که فرضاً یک کلاینت است در نظر بگیرید که در حال ارسال یک فایل بزرگ به میزبان دیگری که یک سرور است، می‌باشد. کلاینت از پروتکل ساده انتظار و توقف استفاده می‌کند. لایه انتقال بر روی سرور به سادگی، TPDU های دریافتی را یکی یکی به داخل لایه انتقال، منتقل می‌کند. در حین این تبادل داده، سرور دچار مشکل می‌شود. وقتی که سرور برمی‌گردد، جداول خود را مجدداً راه‌اندازی می‌کند و دیگر نمی‌داند که دقیقاً از کجا باید ادامه دهد.

سرور به دو صورت زیر می‌تواند برنامه نویسی شود:

۱ـ اول تصدیق (Acknowledge first) ۲ـ اول نوشتن (Write first)

هر کلاینت نیز می‌تواند در یکی از حالات زیر باشد:

۱ـ یک TPDU عقب افتاده (outstanding) یا S1 ۲ـ بدون TPDU عقب افتاده یا S0

کلاینت می‌تواند به یکی از چهار حالت زیر برنامه نویسی شود:

۱ـ همیشه آخرین TPDU را مجدداً ارسال کند. ۲ـ هیچ وقت آخرین TPDU را مجدداً ارسال نکند.

۳ـ تنها حالت S0 را ارسال مجدد کند. ۴ـ تنها حالت S1 را ارسال مجدد کند.

این موارد منجر به هشت حالت ترکیبی می‌شود. برای هر ترکیب، برخی وقایع می‌تواند منجر به خطا در عملکرد پروتکل شود. سه رخداد در سرور قابل تصور است:

۱ـ ارسال یک تصدیق (A) ۲ـ نوشتن در خروجی پردازش (W) ۳ـ خطا (C)

این وقایع می‌توانند به شش ترتیب مختلف AC(W)، AWC، C(AW)، C(WA)، WAC و WC(A) رخ دهند. پرانتزها حاکی از آن هستند که هیچ کدام از حالت‌های A یا W نمی‌توانند پس از حالت C رخ دهند(بدین معنی که با وقوع خطا، کار تمام می‌شود).

مثال۱۰: بازیابی از خطای پیش آمده برای شبکه و مسیریاب در چه زمانی کار ساده‌ای خواهد بود؟

۱) زمانی که کلاینت، همیشه آخرین TPDU را مجدداً ارسال کند. ۲) اگر کلاینت هیچ وقت آخرین TPDU را مجدداً ارسال نکند.

۳) اگر موجودیت انتقال کاملاً در میزبان باشد. ۴) اگر موجودیت انتقال کاملاً خارج از میزبان باشد.

☑ **پاسخ:** گزینه «۳» چناچنه میزبان‌ها و مسیریاب‌ها در معرض خطا و یا صدمه باشند، انجام عملیات بازیابی با اهمیت می‌شود. اگر موجودیت انتقال کاملاً در میزبان باشد، بازیابی از خطای پیش آمده برای شبکه و مسیریاب، کار ساده‌ای خواهد بود.

مثال۱۱: فرض کنید مشکلی در لایه شبکه اتفاق افتاده است. کدام گزینه صحیح است؟

۱) لایه پیوند داده تحت هر شرایطی و بدون هیچ پیش شرطی می‌تواند مشکل لایه شبکه را برطرف کند.

۲) لایه انتقال داده تحت هر شرایطی و بدون هیچ پیش شرطی می‌تواند مشکل لایه شبکه را برطرف کند.

۳) لایه پیوند داده در صورت داشتن اطلاعات کافی می‌تواند مشکل لایه شبکه را برطرف کند.

۴) لایه انتقال داده در صورت داشتن اطلاعات کافی می‌تواند مشکل لایه شبکه را برطرف کند.

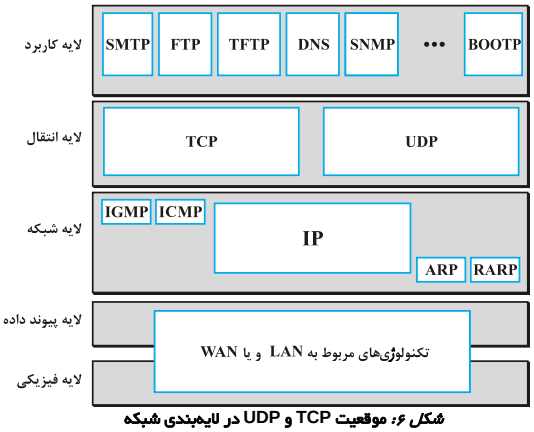
☑ **پاسخ:** گزینه «۴» با توجه به نکته ذکر شده، حل مشکل به عهده لایه بالاتر است آن هم در صورتی که اطلاعات کافی در اختیار آن وجود داشته باشد.

پروتکل‌های لایه انتقال

از پروتکل‌های مشهور لایه انتقال می‌توان به دو پروتکل **TCP** و **UDP** اشاره کرد. همانطور که در ابتدای فصل اشاره شد UDP و TCP، برخلاف IP که بصورت میزبان به میزبان می‌باشد، بصورت پردازش به پردازش یا انتها به انتها می‌باشند. UDP و TCP از شماره پورت، جهت مشخص کردن پردازش استفاده می‌کنند. شماره پورت، یک مقدار 16 بیتی می‌باشد که هر چند در بخش‌های قبلی، به آن اشاره داشتیم اما در همین فصل بیش‌تر در رابطه با آن صحبت خواهیم کرد.

فرض کنید قرار است نامه‌ای را به آدرس مشخصی برسانید. آیا نوشتن نام خیابان و نام کوچه و حتی پلاک برای این کار کفایت می‌کند؟ قطعاً خیر. چرا که ممکن است در یک پلاک، تنها یک واحد مستقر باشد و یا برجی با صدها واحد! آدرس IP نیز داده را تا «دَمِ درِ» میزبان مقصد می‌رساند اما داخل خود میزبان ممکن است چندین پروسس به صورت مستقل مشغول فعالیت باشند. برای تشخیص آن که داده دریافتی متعلق به کدام پروسس است، از مفهوم شماره پورت استفاده می‌شود.

«آدرس سوکت» نیز عبارت است از آدرس IP و آدرس پورت. به‌عنوان یک نمونه از آدرس سوکت می‌توان به 192.168.0.1:21 اشاره داشت. لازم به ذکر است که شماره پورت TCP و UDP از یکدیگر جدا هستند. در ادامه به بررسی پروتکل TCP خواهیم پرداخت و سپس UDP را مطالعه خواهیم نمود. شکل ۶، موقعیت این دو پروتکل را در لایه بندی شبکه نشان می‌دهد.



شکل ۶: موقعیت TCP و UDP در لایه‌بندی شبکه

پروتکل TCP

پروتکل TCP (Transmission Control Protocol)، یک پروتکل اتصال‌گرا (connection-oriented) محسوب می‌شود. این پروتکل اساساً برای تامین یک جریان داده انتها به انتهای **قابل اطمینان** (reliable end-to-end byte stream)، روی یک internetwork غیر قابل اطمینان طراحی شده است. علاوه بر این هدف دیگر TCP، انطباق پویا برای خصوصیات internetwork و همچنین ایجاد مقاومت در مقابل بسیاری از خطاها و مشکلات

است. پروتکل TCP به طور رسمی در RFC 793 تعریف شده است. ضمن اینکه جزئیات بیش‌تر و گونه‌های توسعه یافته آن را می‌توان در RFC 1122 و RFC 1323 پیدا نمود. دقت کنید که TCP یک پروتکل اتصال‌گرا می‌باشد.

هر ماشین TCP، یک موجودیت انتقال TCP(TCP transport entity) دارد که جریان TCP را مدیریت نموده و به‌عنوان واسطی با لایه IP عمل می‌کند. موجودیت TCP، جریان داده کاربر را - با هر طولی - از پردازش‌های محلی دریافت کرده و آن‌ها را به قطعاتی که اندازه‌شان از **64 کیلوبایت** تجاوز نمی‌کند، تقسیم می‌کند. زمانی که داده‌گرام IP که حاوی داده TCP می‌باشد توسط ماشین دریافت می‌شود، به موجودیت TCP تحویل داده می‌شود. موجودیت TCP آن را مجدداً بازسازی کرده تا جریان بایت اولیه را تولید کند. گاهی اوقات منظور از "TCP"، همان موجودیت انتقال TCP (قسمتی از نرم افزار) یا پروتکل TCP (مجموعه ای از قوانین) می‌باشد. TCP باید قابلیت اطمینان را که مورد پسند کاربران است اما IP آن را تامین نمی‌کند، فراهم آورد.

نکته۷: به علت خصوصیاتی که در ادامه خواهیم دید، بسیاری از برنامه‌های کاربردی از TCP استفاده می‌کنند. از جمله این موارد می‌توان به پروتکل‌های مشهور لایه کاربرد همچون HTTP، FTP و SMTP اشاره کرد.

تذکر ۳: در فصل آتی در رابطه با لایه کاربرد صحبت خواهیم کرد.

مثال۱۲: کدام گزینه صحیح نیست؟

۱) از شماره پورت درUDP و TCP برای مشخص کردن پردازش استفاده می‌شود.

۲) انتقال داده در UDP و TCP بصورت پردازش به پردازش می‌باشند و نه انتها به انتها.

۳) TCP باید قابلیت اطمینان را که مورد پسند کاربران است اما IP آن را تامین نمی‌کند، فراهم آورد.

۴) از اهداف TCP، انطباق پویا برای خصوصیات internetwork و همچنین ایجاد مقاومت در مقابل بسیاری از خطاها و مشکلات است.

☑ **پاسخ:** گزینه «۲» پردازش به پردازش و انتها به انتها مفاهیم یکسانی هستند. همانطور که شرح داده شد، UDP و TCP، برخلاف IP که بصورت میزبان به میزبان می‌باشد، بصورت پردازش به پردازش یا انتها به انتها می‌باشند.

مدل سرویس TCP (TCP Service Model)

سرویس TCP زمانی بدست می‌آید که هم فرستنده و هم گیرنده، نقاط انتهایی را که **سوکت (socket)** نامیده می‌شوند را ایجاد کنند. هر سوکت، دارای شماره (آدرس) سوکت است که شامل آدرس IP میزبان و شماره ۱۶ بیت محلی برای آن میزبان که **پورت (port)** نامیده می‌شود، می‌باشد. **به عبارت دیگر آدرس سوکت، ترکیبی از آدرس IP و شماره پورت است (مانند 193.20.26.5:20)** «پورت»، نامگذاری TCP برای TSAP است. برای فراهم آوردن سرویس TCP، یک ارتباط (connection)، باید صریحاً مابین سوکتِ ماشینِ فرستنده و سوکتِ ماشینِ گیرنده ایجاد شود. اسامی سوکت‌ها در جدول سوکت‌های پایه (جدول ۳) لیست شده است.

هر سوکت می‌تواند به طور همزمان، برای چندین ارتباط مورد استفاده قرار گیرد. در این حالت ارتباط‌های مختلف با مشخصه‌های سوکت (socketidentifiers) در هر دو انتها مشخص می‌شوند (سوکت 1، سوکت 2) و نیازی به شماره‌های مدارهای مجازی و یا هیچ مشخصه دیگری نیز وجود ندارد.

پورت‌های با شماره‌های کوچک‌تر از 1024 که به **پورت‌های معروف (well-known)** مشهور هستند، برای سرویس‌های استاندارد در نظر گرفته شده‌اند. به‌عنوان نمونه، برای ایجاد یک ارتباط از راه دور با استفاده از TELNET، از پورت 23 استفاده می‌شود. لیستی از پورت‌های معروف در جدول ۴ قابل مشاهده می‌باشد.

جدول ۴: لیستی از پورت‌های معروف

شماره پورت	توضیح
7	Echo : اکویی از داده‌گرام دریافت شده را به فرستنده بر می‌گرداند.
9	Discard : از هر داده‌گرام دریافتی، چشم پوشی می‌شود.
20	File Transfer Protocol (FTP) : ارتباط داده‌ای FTP
21	File Transfer Protocol (FTP) : ارتباط کنترلی FTP
22	Secure Shell (SSH)
23	Telnet remote login service: شبکه ترمینال
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS) service
80	Hypertext Transfer Protocol (HTTP) used in the World Wide Web
110	Post Office Protocol (POP)
119	Network News Transfer Protocol (NNTP)
143	Internet Message Access Protocol (IMAP)
161	Simple Network Management Protocol (SNMP)
443	HTTP Secure (HTTPS)

ویژگی‌های TCP

کلیه اتصالات TCP، به صورت full-duplex و انتها به انتها هستند. TCP از multicasting و یا پخش همگانی (broadcasting)، پشتیبانی نمی‌کند. اتصال TCP، به صورت جریان بایتی است و نه جریان پیامی به عبارتی دیگر، در TCP جریانی از بایتها بین فرستنده و گیرنده منتقل می‌شود. فرستنده و گیرنده برای ارسال و دریافت بایتها از بافرهای ارسال و دریافت استفاده می‌کنند. جهت بهبود ارسال، بایتها در سگمنت‌هایی قرار گرفته و سپس ارسال می‌گردد. بایت‌های داده ارسالی در هر اتصال بوسیله TCP شمرده می‌شود. شمارش، با یک عدد تصادفی آغاز می‌شود.

نکته۸: مقدار شماره توالی در هر سگمنت، شماره بایت اول داده موجود در هر سگمنت را مشخص می‌کند. مقدار فیلد Acknowledgment در یک سگمنت، شماره بایت بعدیِ یک دسته که انتظار دریافت آن می‌رود را مشخص می‌نماید. شماره Acknowledgment بصورت تجمعی می‌باشد.

مثال۱۳: یک اتصال TCP قصد ارسال یک فایل 6000 بایتی را دارد. اولین بایت با عدد دهدهی 10010 شماره گذاری شده است. شماره توالی هر سگمنت را با فرض ارسال ۴ سگمنت 1000 بایتی و یک سگمنت 2000 بایتی در پایان، مشخص کنید.

☑ **پاسخ:**

10,010 (10,010 to 11,009) Segment 1 -->
Segment 2 --> 11,010 (11,010 to 12,009)
Segment 3 --> 12,010 (12,010 to 13,009)
Segment 4 --> 13,010 (13,010 to 14,009)
Segment 5 --> 14,010 (14,010 to 16,009)

زمانی که لایه کاربرد، داده‌ای را به داخل TCP ارسال می‌کند، پروتکل TCP ممکن است بسته به شرایط مختلف آن را فوراً ارسال کند و یا آن را بافر کند. اما به هر حال گاهی اوقات، کاربرد می‌خواهد تا داده هر چه سریعتر ارسال شود. برای اِعمال اجبار برای خروج داده، کاربرد می‌تواند از **پرچم (فلگ) PUSH** استفاده کند که به TCP می‌گوید که در ارسال تاخیری ایجاد نکند.

یکی از آخرین ویژگی‌های سرویس TCP که باید در اینجا به آن اشاره کرد، **داده اضطراری (Urgent data)** است. زمانی که کاربر تعاملی (interactive)، دکمه‌های DEL یا CTRL-C را برای قطع محاسبه‌ای که از قبل آغاز شده فشار می‌دهد، اپلیکیشن فرستنده، تعدادی اطلاعات کنترلی را در جریان داده قرار داده و آن را از طریق فلگ URGENT تحویل TCP می‌دهد. این امر باعث می‌شود تا TCP جمع آوری داده را قطع کرده و هر آنچه را که برای آن ارتباط جمع آوری کرده است، سریعاً ارسال کند. زمانی که داده اضطراری در گیرنده دریافت شود، یک وقفه (interrupt) در اپلیکیشن گیرنده رخ می‌دهد.

مثال۱۴: در مورد TCP کدام گزینه صحیح است؟

۱) برای اِعمال اجبار برای خروج داده، کاربرد می‌تواند از پرچم (فلگ) URGENT استفاده کند.
۲) تعدادی از اتصالات TCP، full-duplex و نقطه به نقطه هستند.
۳) TCP از multicasting و یا پخش همگانی (broadcasting)، پشتیبانی مناسبی می‌کند.
۴) اتصال TCP، به شکل جریان بایتی است و نه جریان پیامی.

☑ **پاسخ:** گزینه «۴» برای اِعمال اجبار برای خروج داده، کاربرد می‌تواند از پرچم (فلگ) PUSH استفاده کند که به TCP می‌گوید در ارسال تاخیری ایجاد نکند(نادرستی گزینه ۱). کلیه اتصالات TCP، full-duplex و نقطه به نقطه هستند (نادرستی گزینه ۲).TCP از multicasting و یا پخش همگانی(broadcasting)، پشتیبانی نمی‌کند(نادرستی گزینه ۳).

در این قسمت به بررسی اجمالی پروتکل TCP می‌پردازیم. هر بایت روی ارتباط TCP، **شماره ترتیب 32 بیتی** خود را دارد (sequence number). شماره‌های ترتیب، هم در پیام‌های تصدیق و هم در روش پنجره کاربرد دارند.

موجودیت‌های TCP گیرنده و فرستنده، داده را با یکدیگر به شکل سگمنت مبادله می‌نمایند. هر سگمنت شامل سرآیندی به طول ثابت 20 بایت است که به دنبال آن ممکن است صفر یا چندین نوع داده قرار گرفته باشد. اندازه سگمنت توسط نرم‌افزار TCP تعیین می‌شود. نرم افزار TCP می‌تواند داده‌های مختلفی را در یک سگمنت جای دهد و یا آن‌ها را در قالب چندین سگمنت در آورد. دو نکته، اندازه سگمنت را محدود می‌کند:

۱ـ هر سگمنت که حاوی سرآیند TCP است، باید در قالب داده(payload) IP که 65,535 است، گنجانده شود.

۲- هر شبکه برای خود یک حداکثر اندازه واحد انتقال یا **MTU (maximum transfer unit)** دارد. هر سگمنت باید حداکثر به اندازه MTU باشد. هر سگمنت که اندازه آن از حد مجاز تعریف شده در شبکه بیش‌تر باشد، توسط مسیریاب به چندین سگمنت کوچکتر تقسیم می‌شود. هر سگمنت تولید شده به این شکل، برای خود سرآیند TCP و IP اختیار می‌کند. به این ترتیب عمل کوچک کردن سگمنت‌ها یا همان **fragmentation** که توسط مسیریاب‌ها انجام می‌شود، باعث افزایش حجم کلی سربار در شبکه می‌شود.

پروتکل پایه‌ای که توسط موجودیت‌های TCP استفاده می‌شود، پروتکل پنجره لغزان با طول پنجره متغیر است. پنجره لغزان(Sliding Window)، برای کارایی بیشتر ارسال و کنترل جریان داده بکار می‌رود تا مقصد در داده‌ها «غوطه‌ور» نشود.

مثال۱۵: کدام گزینه صحیح است؟

۱) شماره ترتیب سگمنت‌های TCP هر بار یک واحد، یک واحد لزوماً افزایش پیدا نمی‌کند.

۲) حداکثر اندازه فایلی که قبل از Reset شدن شماره ترتیب TCP می‌توان ارسال کرد، حدوداً برابر 4 گیگابایت است.

۳) هر دو

۴) هیچ‌کدام

☑ **پاسخ:** گزینه «۳» از آن‌جا که مقدار شماره ترتیب در هر سگمنت، شماره بایت اول موجود در آن است لذا شماره ترتیب برای سگمنت‌های TCP یکی یکی اضافه نمی‌شود (دلیل درستی گزینه ۱).

از طرف دیگر گفته شده که شماره ترتیب هر بایت در TCP، 32 بیتی است. با این توصیف حداکثر شماره ترتیب که می‌توان تولید کرد برابر $2^{32}-1$ خواهد بود. بنابراین حداکثر $2^{32}-1$ بایت می‌توان در نظر گرفت. اگر از عدد 1 در مقابل 2^{32} صرف‌نظر کنیم، داریم:

$$2^{32}=2^2\times 2^{30}=4\times 2^{30}=4GB$$

بخاطر دارید که اگر بخواهیم دقیق صحبت کنیم، داریم: $1K=2^{10}$ و $1M=2^{20}$ و $1G=2^{30}$

مثال۱۶: اگر از پروتکل TCP بر روی شبکه‌ای با نرخ ارسال 100Mbps استفاده کنیم، تقریباً چند ثانیه باید بگذرد تا شماره ترتیب TCP، ریست (Reset) شود؟

۱) 300　　۲) 328　　۳) 400　　۴) 428

☑ **پاسخ:** گزینه «۲» همان‌طور که در مثال قبل توضیح داده شد پس از ارسال 2^{32} بایت، شماره ترتیب TCP، ریست می‌شود. باید مدت زمان لازم برای ارسال چنین حجمی از داده را بر روی شبکه داده شده بدست آوریم.

$$\frac{\text{بیت}}{\text{ثانیه}}:\frac{100\times 2^{20}}{1}=\frac{2^{32}\times 8}{t}\quad\Rightarrow t=\frac{2^{32}\times 8}{100\times 2^{20}}=327.6\simeq 328\,sec$$

مثال۱۷: شبکه‌ای را با حداکثر نرخ ارسال 512bps در نظر بگیرید. اگر شماره ترتیب 5 بیتی و مقدار فیلد TTL برابر 16 ثانیه باشد، اندازه بسته‌ها کدام گزینه می‌تواند باشد؟

۱) 256b　　۲) 512b　　۳) 128b　　۴) 646 b

☑ **پاسخ:** گزینه «۱» باید حساب کنیم که قبل از آن که مدت زمان اعتبار داده‌ها منقضی شود (یعنی در عرض 16 ثانیه) چه میزان داده ارسال می‌شود.

$$\frac{\text{بیت}}{\text{ثانیه}}:\frac{512}{1}=\frac{L}{16}\Rightarrow L=512\times 16=2^9\times 2^4=2^{13}\,bit$$

از آن‌جا که با یک شماره ترتیب 5 بیتی، 2^5 عدد شماره ترتیب مختلف می‌توان برای بسته‌ها در نظر گرفت. اگر اندازه هر بسته برابر x بیت باشد، باید داشته باشیم:

$$2^{13}=2^5\times x\quad\Rightarrow x=2^8=256\,b$$

نکته۹: پنجره لغزان TCP بایت گرا است و نه سگمنت گرا. اندازه پنجره لغزان بوسیله پنجره دریافت کننده معین می‌گردد؛ اگرچه اندازه واقعی پنجره می‌تواند بدلیل تراکم در شبکه کوچکتر شود.

دقت نمایید که اندازه پنجره لغزان ممکن است افزایش یا کاهش یابد.

در مورد پنجره لغزان TCP توجه به نکات زیر حائز اهمیت است:

۱ـ مبداء مجبور نیست تا تمام داده‌های ممکن در اندازه کامل پنجره را ارسال کند.

۲ـ اندازه پنجره می‌تواند بوسیله مقصد افزایش و یا کاهش یابد.

۳ـ مقصد می‌تواند یک Acknowledgment را در هر زمانی ارسال کند.

۴ـ اگر به هر دلیلی همچون معیوب بودن و یا گم شدن سگمنت ارسالی، در فاصله زمانی مشخص (به اندازه Time out) Ack دریافت نشود، فرستنده مجدداً داده‌های ارسالی پس از آخرین Ack دریافتی را ارسال می‌نماید.

مثال۱۸: کدام گزینه در مورد TCP صحیح نیست؟

۱) مقصد می‌تواند یک Acknowledgment را در هر زمانی ارسال کند.

۲) پنجره لغزان TCP بایت‌گرا است.

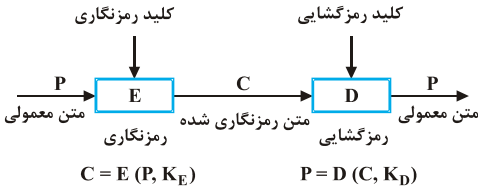
۳) هر سگمنت شامل سرآیندی به طول ثابت 20 بیت است.

۴) عمل fragmentation که توسط مسیریاب‌ها انجام می‌شود، باعث افزایش حجم کلی سربار در شبکه می‌شود.

☑ **پاسخ:** گزینه « ۳» هر سگمنت شامل سرآیندی به طول ثابت 20 بایت یا 160 بیت است که به دنبال آن ممکن است صفر یا چندین نوع داده قرار گرفته باشد. سایر گزینه‌ها مشکلی ندارند.

رمزنگاری (Cryptography)

رمزنگاری تاریخچه جذاب و طولانی دارد. در طول تاریخ، طبقات مختلفی از افراد، عادت به رمزنگاری داشته‌اند: نظامی، دیپلمات، خاطره نویس و ... . شکل ۱ یک سیستم رمزنگاری را نشان داده است.



شکل ۱: یک سیستم رمزنگاری

پیام P، که پیام ساده و بدون رمز (plaintext) است با پارامتری شدن به وسیله یک کلید (K_E)، توسط یک تابع (E)، رمزنگاری می‌شود. نتیجه‌ی حاصل، یک پیام رمزنگاری شده است که Cryptogram یا Cipher text نامیده می‌شود. پیام رمزنگاری شده روی لینک ارتباطی، به طرف گیرنده ارسال می‌شود. گیرنده با استفاده از تابع (D) و کلیدی دیگر (K_D)، پیام را رمزگشایی می‌کند. اگر کلید رمزنگاری و رمزگشایی مشابه باشند، سیستم را **تک کلیده یا متقارن (Symmetric)** می‌گوییم یعنی: $K = K_E = K_D$. اما اگر این دو با هم متفاوت باشند، سیستم، **دو کلیده یا غیرمتقارن** (Asymmetric) خواهد بود یعنی: $K_E \# K_D$.

به عمل رمزنگاری Cryptography و به عمل رمزگشایی Cryptoanalyses گفته می‌شود. این دو با یکدیگر، بخشی از علم Cryptology را شکل می‌دهند.

جانشانی نشانه‌ها (Substitution Ciphers)

در روش جانشانی نشانه‌ها حرف یا گروهی از حروف، جانشین یک حرف یا گروهی از حروف دیگر می‌شود. به مثال زیر دقت کنید:

متن اصلی: 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

متن رمزنگاری شده: Q WE R T Y U I O P A S D F G H J K L Z X C V B N M

رمزنگاری قیصری (Caesar Cipher)

در این روش، هر حرف در متن با حرف دیگری که k حرف بعد از آن در ترتیب آلفا بت قرار دارد جایگزین می‌شود. نسخه اصلی رمزنگاری قیصری، از $K = 3$ استفاده می‌کند. بنابراین: $a \to d, b \to e, \ldots$. در این صورت اگر حروف a تا z را از 1 تا 26 شماره گذاری نماییم (تعداد حالات یا $N = 26$)، آنگاه تابع رمزنگاری با استفاده از کلید k برابر خواهد بود با: $E = (P + K) \bmod N$

جانشانی تک آلفا بتی (Mono Alphabetic Substitution)

در یک روش مرسوم‌تر، کلیه حروف با توجه به یک جدول تبدیل می‌شوند. در این حالت کلید آن چیزی است که توسط تبدیل کاراکتر(character map) نشان داده می‌شود. مثلاً طبق یک جدول مفروض، a تبدیل به t می‌شود. در این حالت برای به دست آوردن کلید، باید $26! = 4 \times 1026$ حالت مختلف را مورد بررسی قرار داد. حتی کامپیوتری که در هر میکرو ثانیه یک حالت را می‌تواند چک کند، برای بررسی کامل حالات، به 1013 سال نیاز خواهد داشت!!!

شکستن رمزنگاری تک آلفا بتی (Breaking Mono-Alphabetic Ciphers)

تکرار حروف، ترکیب دو حرف و یا ترکیب‌های سه‌تایی در انگلیسی و اکثر زبان‌های دنیا معمول است. با محاسبه تکرار کل حروف و ترکیبات در متن رمزنگاری شده، متخصص رمزنگاری می‌تواند متن اصلی را بدون زحمت زیاد، بازیابی نماید.

رمزنگاری به روش جابجایی (Transposition Ciphers)

مکان حروف در رمزنگاری به روش جابجایی، جا به جا شده و شیفت داده می‌شوند و به همین دلیل، ترتیب آن‌ها به هم می‌خورد. از دید ریاضیات، برای رمزنگاری محل و موقعیت کاراکترها، از یک تابع دوسویه(Bijective Function) استفاده می‌شود. از تابع معکوس نیز برای رمزگشایی استفاده می‌شود. در اینجا اطلاعات تکراری کمک چندانی برای کشف رمز نخواهند کرد. به مثال زیر دقت کنید:

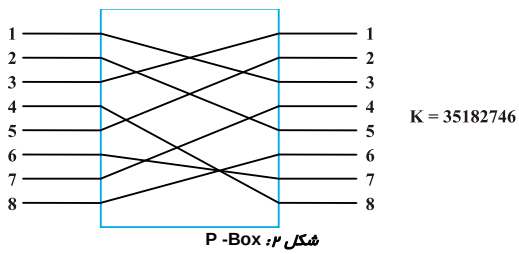
متن اصلی:Communication .is. easy

متن رمزنگاری شده: Cuan, yont.emiiiamcoss

لازم به ذکر است که پیاده‌سازی‌های مختلفی تا به حال برای رمزنگاری به روش جابجایی پیشنهاد شده است.

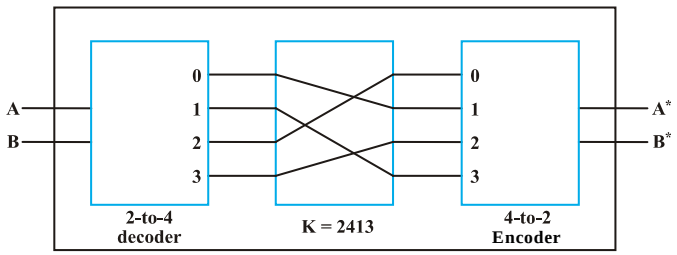
تولید رمزها (Product Ciphers)

برای تعویض حروف از جعبه P (P-Box) استفاده می‌شود (p اول permutation به معنای جایگشت است). یک جعبه P در شکل ۲ نشان داده شده است. این شکل، n ورودی و n خروجی دارد. متون کاراکتری معمولاً توسط کدهای 8 بیتی (ASCII) نشان داده می‌شوند. کلید K، موقعیت جدید را برای هر بیت ورودی تعیین می‌کند.



شکل ۲: P -Box

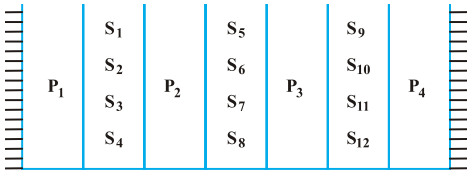
اگر یک رمزنگار بخواهد از هر جانشانی برای هر کاراکتر 8 بیتی به کاراکتر 8 بیتی دیگری پشتیبانی کند، نیاز به جدولی با 256 مدخل کاراکتری خواهد داشت. طول کل کلید برای توصیف جدول، به $256 * 8 = 2048$ می‌رسد. برای کاهش این حجم، همان‌طور که شکل ۳ نشان داده است، یک P-Box مابین یک رمزگشا (decoder) و رمزنگار (encoder)، محصور (encapsulated) می‌شود (جعبه S).



شکل ۳: جعبه‌های S

از واحدهای پشت سر هم جعبه‌های Product Ciphers P و Product Ciphers S ، برای افزایش قدرت الگوهای رمزنگاری استفاده می‌شود. مثالی از یک Product Ciphers، DES (Data Encryption Standard) است که توسط موسسه ملی علم و فناوری National Institute of Science and Technology (NIST) در سال 1977 ایجاد شد.

اصل رمزنگاری به‌صورت زیر است: کلیه پیام‌ها باید شامل افزونگی (redundancy) باشند تا از فریب گیرنده‌ها توسط نفوذگرها برای عمل بر روی پیام‌های اشتباه جلوگیری به عمل آید. شکل ۴ یک Product Ciphers را نمایش می‌دهد.



شکل ۴: یک Product Ciphers

الگوریتم‌های کلید امنیت (Secret Key Algorithms)

رمزنگاری‌های جدید از اصول مشابهی که در روش‌های قدیمی استفاده می‌شد، مانند جابجایی و جانشانی بهره می‌برند اما با این حال تفاوت‌هایی نیز با نیاکان خود دارند. در گذشته، رمزنگارها برای امنیت خود باید از الگوریتم‌های ساده‌ای که بر مبنای کلیدهای بسیار طولانی قرار داشت، استفاده می‌کردند. اما امروزه قضیه برعکس شده است به گونه‌ای که الگوریتم رمزنگاری باید به گونه‌ای پیچیده باشد که حتی خبره‌ترین افراد هم قادر به رمزگشایی آن نباشند. جابجایی و جانشانی می‌توانند توسط مدار ساده‌ای پیاده‌سازی شوند. شکل ۲ وسیله‌ای را که P-box نام دارد، نشان می‌دهد که برای جابجایی 8 بیت ورودی مورد استفاده قرار می‌گیرد. اگر این هشت بیت ورودی از بالا به پایین 01234567 در نظر گرفته شوند آنگاه ۸ بیت خروجی این P-box برابر می‌شود با 36071245. با سیم‌بندی مناسب داخلی، یک P-box می‌تواند برای هر نوع جابجایی مورد استفاده قرار گیرد و آن را با سرعت بالایی انجام دهد.

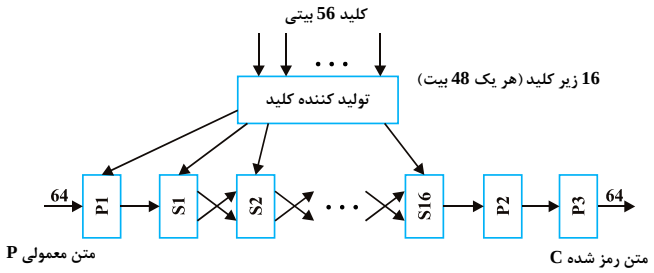
جانشانی می‌تواند توسط جعبه‌های S (S-boxes) که در شکل ۳ نشان داده شده است، انجام شوند. در این مثال، ورودی، یک متن ساده 2 بیتی است و متن رمزنگاری شده دو بیتی دیگر، خروجی را تشکیل می‌دهد. دو بیت ورودی، یکی از 4 خطوط موجود را از اولین مرحله انتخاب کرده و آن را به مقدار 1 تنظیم (set) می‌کنند. کلیه خطوط دیگر برابر 0 می‌شود. دومین مرحله یک P-box است. در سومین مرحله ورودی انتخاب شده، مجدداً به شکل دودویی رمزنگاری می‌شود. مجدداً با سیم‌بندی مناسب P-box داخل S-box، هر نوع جانشانی قابل اجرا خواهد بود.

توان واقعی این اجرا تنها زمانی آشکار می‌شود که تعدادی جعبه را به شکل 4 به‌صورت سری به یکدیگر متصل کنیم. در این مثال، 12 خط ورودی توسط مرحله اول، جابه جا شده‌اند. از نظر تئوری، این امکان وجود دارد که مرحله دومی را داشت که یک S-box باشد که عدد 12 بیتی را به عدد دیگر 12 بیتی نگاشت کند. اما چنین تجهیزاتی نیاز به $2^{12} = 4096$ سیم متقاطع در مرحله میانی خود خواهد داشت. به جای این، ورودی به 4 گروه 3 بیتی تقسیم می‌شود که هر یک مستقل از دیگری جانشانی می‌شود. هر چند کاربرد این روش گسترده نیست اما روش قدرتمندی به شمار می‌رود.

استاندارد رمزنگاری داده یا DES(Data Encryption Standard)

الگوریتم DES در اوایل دهه هفتاد توسط IBM توسعه یافت. DES یک مثال از Product Ciphers است که عمل رمزنگاری را به‌صورت بلوک‌گرا (block-oriented) انجام داده و بلوک‌های 64 بیتی را رمزنگاری می‌نماید. یک کلید 56 بیتی، رمزنگاری را کنترل می‌کند. کلید 56 بیتی، به 16 زیر کلید 48 بیتی برای کنترل 16 جانشانی در یک تراشه DES تبدیل می‌شود (شکل ۵). پروسه رمزنگاری، شامل 19 مرحله در تراشه DES است. اولین مرحله، جابجایی با استفاده از قانون جابجایی ثابتی است که با 16 جایگزینی و در نهایت 2 جابجایی نهایی دنبال می‌شود. در هر مرحله جانشانی، پرارزش‌ترین و کم‌ارزش‌ترین بیت‌های بلوک‌های 32 بیتی با هم تعویض می‌شوند. 32 بیت پرارزش قبلی تحت کنترل زیر کلید (subkey)، جانشانی و جابجا می‌شوند و نتیجه، به مرحله بعد ارسال می‌شود. اشکال این روش این است که کلید رمزنگاری و رمزگشایی مشترک است.

لازم به ذکر است که طول کلید اولیه پیشنهاد شده توسط IBM ، 128 بیت بوده است. با این حال آژانس امنیت ملی ایالات متحده، طول 56 بیتی را پیشنهاد داد.



شکل ۵: تولید متن رمزنگاری شده DES

حالات کاری DES

حالات کاری DES عبارتند از:

- **Electronic Code Book (ECB)** : هر بلوک از متن رمزنگاری شده، به طور مستقل از سایر بلوک‌ها رمزنگاری می‌شود. به همین خاطر، هر بلوک متن رمز شده با یک بلوک متن ساده مطابقت دارد؛ دقیقاً مانند یک کتاب رمزی (codebook).
- **Chain Block Cipher (CBC)** : ECB از درج بلوک‌های تکراری جلوگیری نمی‌کند چراکه نسبت به هر بلوک به طور مستقل رفتار می‌کند. ضعف دیگر آن این است که الگوهای بلوک‌های متنی مشابه، بلوک‌های متنی رمز شده‌ی مشابهی را تولید می‌کنند.

دقت کنید که رشته 32 بیتی، ابتدا باید به داده 48 بیتی تبدیل شود. سپس عمل XOR با کلید 48 بیتی انجام می‌شود و نتیجه باید به 32 بیت تبدیل شود. در تمام مراحل، 16 کلید 48 بیتی از روی کلید 56 بیتی ساخته می‌شوند.

در جهت بهبود DES برای جریان‌های ارتباطی، هر بلوک 64 بیتی با متن رمز شده 64 بیتی قبلی، XOR شده و وارد تراشه DES می‌شود. علاوه بر یک کلید امنیت مشترک، فرستنده و گیرنده باید در مورد بردار اولیه (initial vector) که باید با اولین بلوک جریان پیام XOR شود، با هم به توافق برسند.

- **Cipher Feedback Mode (CFM)** : CFM حالت جایگزین DES، برای کاراکترهای 8 بیتی محسوب می‌شود. کاراکتر ورودی با کم ارزش‌ترین بایت خروجی DES، XOR شده و سپس روی لینک ارتباطی ارسال می‌شود.

برای جمع آوری بیت‌های لازم برای بلوک رمزنگاری 64 بیتی، کاراکترهای خروجی توسط شیفت رجیستر گردآوری می‌شوند. هر کاراکترِ ورودی در شیفت رجیستر پیش رفته و یک رمزنگاری DES جدید را تحریک (trigger) می‌کند. به همین دلیل کاراکتر ورودی بعدی با خروجی جدید DES، XOR می‌شود. روش CFM برای استفاده در خطوط سریال مناسب است.

رمزنگاری با کلیدهای عمومی و خصوصی

همانطور که اشاره شد، اشکال DES این است که فرستنده و گیرنده از کلید مشترکی استفاده می‌نمایند. سوالی که پیش می‌آید، چگونگی توزیع کلیدها مابین زوج‌های ارتباطاتی است.

در رمزنگاری و به شکل کلی آن، هر کاربر باید دارای دو کلید باشد:

۱ـ کلید عمومی: از کلید عمومی برای رمزنگاری پیامی که قرار است به کاربر ارسال شود، استفاده می‌شود.

۲ـ کلید خصوصی: از این کلید، برای رمزگشایی پیام استفاده می‌شود.

سیستم‌های کلید خصوصی

در سال 1976، Diffie و Hellman پیشنهاد استفاده از الگوریتم‌های رمزنگاری E و رمزگشایی D را دادند، به طوری که:

۱ـ $D(E(P)) = P$

۲ـ استخراج D از E بسیار سخت است.

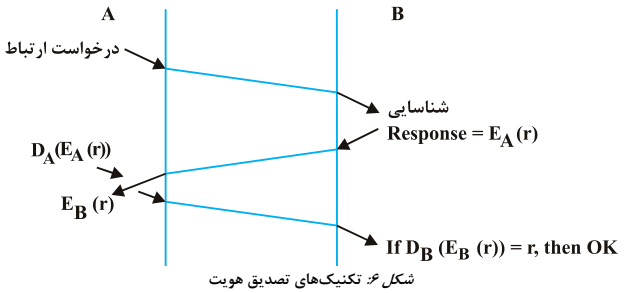
۳ـ E نمی‌تواند توسط حمله افراد نخبه شکسته شود. تحت این شرایط، E می‌تواند عمومی باشد در حالی که D به طور محرمانه نگهداری می‌شود.

فرستنده A را در نظر بگیرید که می‌تواند با استفاده از رمزنگاری کلید عمومی، یک کلید محرمانه را برای گیرنده B ارسال کند. ابتدا A ، کلید عمومی B (E_B) را از یک بانک اطلاعاتی عمومی بازیابی می‌کند. با داشتن E_B ، اکنون A می‌تواند پیامی که شامل کلید K برای B است را رمزنگاری کند. تنها ایستگاه B کلید رمزگشایی محرمانه (D_B) خود را در اختیار دارد که توسط آن می‌تواند متن اصلی و بوسیله آن، K را بازیابی کند.

از آنجا که سیستم‌های رمزنگاری متقارن (symmetric) با کلید مشترک K، سریعتر از الگوهای کلید عمومی نامتقارن (asymmetric) اجرا می‌شوند، A و B تصمیم می‌گیرند که به جای تکنیک کلید عمومی، از یک الگوریتم کلید محرمانه مشترک برای باقیمانده ارتباط خود استفاده نمایند.

تصدیق هویت (Authentication)

منظور از تصدیق هویت، شناسایی یا تعیین هویت(identification) و تصدیق (verification) است. شناسایی یا تعیین هویت، پروسه‌ای است که توسط آن هر فرد، مشخصه‌ی به خصوصی را برای خود مدعی می‌شود. در حالی که تصدیق، پروسه‌ای است که توسط آن صحت و سقم این ادعا مورد بررسی قرار می‌گیرد.



شکل ۶: تکنیک‌های تصدیق هویت

پروسه تصدیق هویت با توجه به شکل ۶ و با استفاده از کلید عمومی به شکل زیر انجام می‌شود. ایستگاه A یک درخواست ارتباط (connection request) را به B ارسال می‌کند. B، نتیجه بررسی صلاحیت (challenge) A را با رمزنگاری یک عدد تصادفی r با کلید عمومی E_A به A برمی‌گرداند. فقط A قادر است تا پیام بررسی صلاحیت را با کلید خصوصی خودش(D_A)، رمزگشایی نماید. A نتیجه را یا به‌صورت باز یا به صورت رمزنگاری شده با کلید عمومی B، ارسال می‌کند. اکنون B می‌تواند بررسی کند که آیا عدد تصادفی به طرز صحیحی رمزگشایی شده است یا خیر.

الگوریتم RSA

بر پایه ایده Diffie و Hellman یک الگوی رمزنگاری عمومی توسط Rivest، Shamir و Adleman در سال 1978 ابداع شد که RSA نام گرفت. در این الگو کلیدهای متفاوتی که از یکدیگر قابل اشتقاق نیستند، الگوی رمزنگاری E و رمزگشایی D را انجام می‌دهند. RSA یکی از انواع الگوریتم‌های نامتقارن است. در الگوریتم نامتقارن (بر خلاف متقارن)، از کلیدهای متفاوتی برای رمزنگاری و رمزگشایی استفاده می‌شود. اشاره داشتیم که کلید رمزنگاری را کلید عمومی و کلید رمزگشایی را کلید خصوصی می‌نامند.

در الگوریتم RSA ابتدا داده‌ها را به قسمت‌های دو کاراکتری تقسیم می‌کنیم. هر کاراکتر را نیز به عدد تبدیل می‌کنیم. به‌عنوان مثال A به 01، B به 02، C به 03 و... تبدیل می‌شود. در این روش انتخاب کلید عمومی و خصوصی به‌صورت زیر است:

۱ـ دو عدد اول p و q انتخاب می‌شوند (تا جایی که می‌توانند باید بزرگ باشند(دویست رقمی))

۲ـ اعداد n و z به‌صورت زیر محاسبه می‌شوند:

$$n = p \times q$$
$$z = (p-1) \times (q-1)$$

۳ـ عدد d را به گونه‌ای انتخاب می‌کنیم که نسبت به z اول باشد.

۲ـ بر اساس d، عدد e به گونه‌ای انتخاب می‌شود که رابطه روبه‌رو برقرار باشد: $(e \times d) \bmod (z) = 1$

مثال۱: فرض کنید که suzanne را بخواهیم رمز کنیم:

پاسخ : p و q را انتخاب می‌کنیم: $p = 3 \ , \ q = 11$

$n = 33 \ , \ z = 20$ یک عدد اول که نسبت به z اول است، انتخاب می‌شود: $d = 7$ و e را بدست می‌آوریم به گونه‌ای که $(e \times 7) \bmod 20 = 1$. بنابراین e را می‌توان برابر 3 یا 23 در نظر گرفت.

نتیجه به‌صورت زیر است:

Pain text (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
Sender's computation			Receiver's computation			

عملکرد و سرویس‌های لایه نشست (session)

وظیفه لایه نشست (شکل ۷)، درخواست برای ایجاد یک ارتباط منطقی برای پردازش ارتباطی است. این ارتباط به اشتراک گذاشته شده، نشست نامیده می‌شود. این لایه، همزمانی داده بین کارهای کاربر (user tasks) را با ایجاد **نقاط بررسی (checkpoints)** تامین می‌کند. به این طریق در صورت رخداد خطای شبکه، تنها داده‌هایی که پس از نقطه بررسی قرار دارند، نیاز به ارسال مجدد خواهند داشت.

شکل ۷: لایه نشست

وظایف لایه نشست عبارتند از:

۱ـ ایجاد ارتباط نشست‌ها با لایه انتقال

۲ـ کنترل جریان نشست

۳ـ تبادل داده بین کارها (tasks)

۴ـ ایجاد، اتمام و ایجاد مجدد ارتباط‌ها

۵ـ مدیریت لایه نشست و ارتباط با لایه‌های مجاور

۶ـ کنترل محاوره (Dialogue) به‌عنوان مثال محاوره با چه کسی، چه زمانی، چه مدتی، half -duplex یا full-duplex صورت می‌پذیرد.

۷ـ بازیابی (recovery) از مشکلات ارتباطی در طول نشست بدون از دست رفتن داده.

سرویس‌هایی که توسط لایه نشست ارائه می‌شوند عبارتند از:

۱ـ آغاز و اتمام نشست ۲ـ ارسال داده ۳ـ کنترل محاوره ۴ـ همگام سازی ارتباط نشست (Synchronization) ۵ـ اعلام خطاهای غیر قابل بازیابی

مثال۲: کدام یک، از جمله وظایف لایه نشست به شمار نمی‌رود؟

۱) تبادل داده بین کار(task)ها ۲) کنترل محاوره ۳) بازیابی ۴) کنترل جریان

☑ **پاسخ :** گزینه «۴» کنترل جریان از جمله وظایف لایه نشست نمی‌باشد.

Abstract syntax notation 1(ASN.1)

قرارداد ASN.1، قرارداد استاندارد ISO است که برای مشخص کردن مستقل انواع داده در قوانین کدگذاری توسط SNMP استفاده می‌شود. در عمل ASN.1 از نظر بزرگی، پیچیدگی و ناکارایی به OSI شبیه است. فردی که بخواهد با مفهوم SNMP آشنا شود باید با ASN.1 نیز آشنا باشد. سینتَکس انتزاعی ASN.1، اساساً زبانی است برای اعلام داده. ASN.1 به کاربر اجازه تعریف اشیای اولیه و ترکیب آن‌ها به موارد پیچیده‌تر را می‌دهد. تعاریف موجود در ASN.1 از نظر عملکرد، به فایل‌های سرآیند برنامه‌های زبان C شباهت دارد. انواع داده پایه اصلی ASN.1 در SNMP ، در جدول ۲ لیست شده است. استفاده از کد، در ادامه توضیح داده خواهد شد.

جدول ۲: انواع داده پایه‌ای اصلی ASN. 1 در SNMP

نوع اصلی	توضیح	کد
INTEGER	عدد صحیح با طول دلخواه	2
BIT STRING	رشته‌ای از صفر یا بیت‌های دیگر	3
OCTET STRING	رشته‌ای از صفر یا بایت‌های بدون علامت دیگر	4
NULL	نگهدارنده مکان (place holder)	5
OBJECT IDENTIFIER	تعریف رسمی نوع داده	6

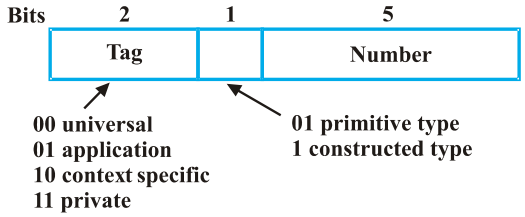
سینتکس انتقال ASN-1(ASN-1 Transfer Syntax)

یک سینتکس انتقال ASN-1 تعیین می‌کند که چگونه مقادیر انواع ASN.1 بدون هیچ گونه ابهامی به دنباله‌ای از بایت‌های مناسب برای ارسال تبدیل می‌شوند. سینتکسِ انتقالِ استفاده شده توسط ASN.1، BER (Basic Encoding Rules) نام دارد. البته ASN-1 سینتکس دیگری هم دارد که SNMP از آن استفاده نمی‌کند. از آنجا که قوانین بازگشتی هستند، رمزنگاری یک شی ساختار یافته (structured object)، صرفاً برابر زنجیرهای (concatenation) از رمزنگاری‌های اشیای تشکیل دهنده (component objects) می‌باشد. بدین صورت کل اشیای رمزنگاری شده می‌توانند به یک دنباله معروفِ شناخته شده از اشیای اصلی رمزنگاری شده، کاهش یابند. رمزنگاری‌های این اشیا به نوبت توسط BER تعیین می‌شوند. قاعده اصلی قوانین رمزنگاری این است که هر مقدار ارسال شده، چه اصلی چه ساختنی (constructed)، شامل چهار فیلد زیر باشد:

۱ـ مشخصه (identifier) یعنی نوع یا نشانه (type or tag) ۲ـ طول فیلد داده بر حسب بایت

۳ـ فیلد داده ۴ـ فلگ پایان محتویات؛ اگر طول داده مشخص نباشد.

مورد آخر در ASN-1 مجاز است در حالی که SNMP اجازه آن را نمی‌دهد. بنابراین ما فرض می‌کنیم که طول داده همیشه مشخص است.



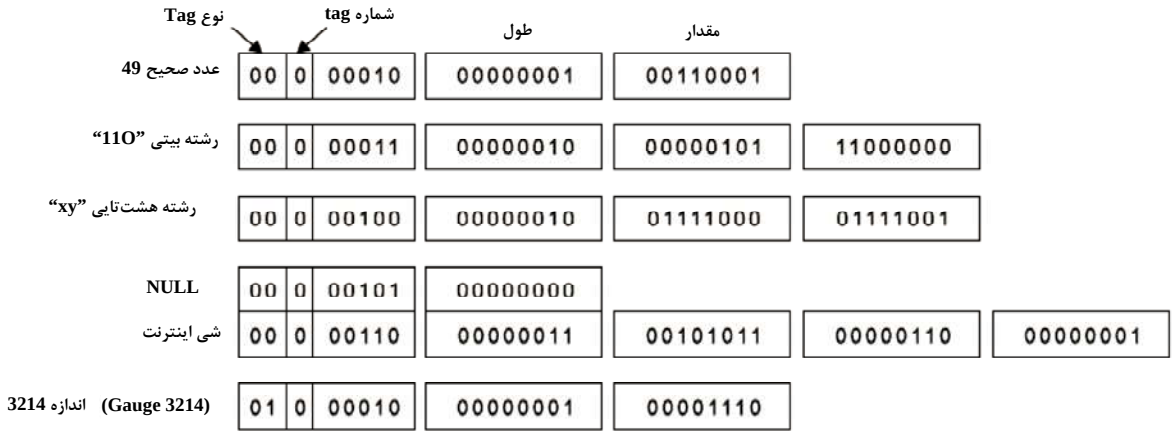
شکل ۸: اولین بایت از هر آیتم داده ارسالی در ASN. 1

همان‌طور که شکل ۸ نشان می‌دهد، اولین فیلد، خود از سه زیر فیلد تشکیل شده است. دو بیت بالایی نوع نشانه (tag type) را مشخص می‌کنند. بیت بعدی نشان دهنده اصلی(0) یا غیر اصلی(1) بودن مقدار است. بیت‌های نشانه، 00، 01، 10 و 11 هستند که به ترتیب نمایشگر همگانی (UNIVERSAL)، کاربرد (APPLICATION)، متن مخصوص (Context-Specific) و خصوصی (private) هستند.

اگر مقدار نشانه از 0 تا 30 باشد، 5 بیت باقیمانده می‌تواند برای رمزنگاری مقدار به کار رود. اگر مقدار نشانه 31 یا بیشتر از آن باشد، 5 بیت پایینی، حاوی 11111 با مقدار درست (true value) برای بایت یا بایت‌های بعدی خواهد بود.

نکته۱: نحوه رمزنگاری فیلد داده، بستگی به نوع داده دارد.

اعداد صحیح به‌صورت مکمل دو رمزنگاری می‌شوند. اما رشته‌ها به طریق دیگری رمزنگاری می‌شوند. تنها مشکلی که وجود دارد چگونگی نشان دادن طول است. فیلد طول، تعداد بایت‌های مقدار را نشان می‌دهد نه تعداد بیت‌های آن را. یک راه حل، ارسال یک بایت قبل از رشته بیت حقیقی است که تعداد بیت‌های آخرین بایت (از 0 تا 7) را که استفاده نشده‌اند، نشان می‌دهد. بنابراین رمزشده رشته 9 بیتی ’010011111‘، برابر با 80 ,4F ,07 در مبنای 16 می‌شود.



شکل ۹: چند مثال از نحوه رمزنگاری ASN-1

نحوه کار با رشته‌های هشت‌تایی(Octet strings) ساده است. مقدار NULL با تنظیم فیلدِ طول به صفر نشان داده می‌شود. در عمل هیچ مقدار عددی ارسال نمی‌شود. یک مشخصه شیء نیز بر حسب توالی اعداد صحیح خود رمزنگاری می‌شود. شکل ۹ چند مثال از مقادیر رمز شده را نشان می‌دهد. لازم به ذکر است که کدهای اسکی حروف x و y به ترتیب برابر 120 و 121 است (ر.ک به ضمیمه د).

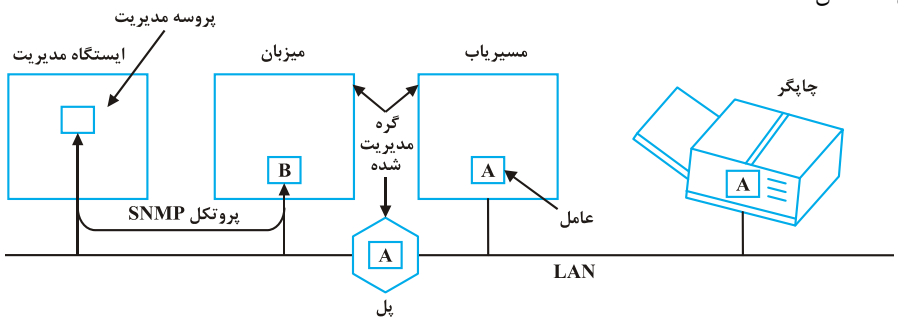
پروتکل مدیریت شبکه ساده SNMP (Simple Network Management Protocol)

زمانی که ARPANET وارد دنیای اینترنت با چندین backbone و اپراتور شد، راه حل ذکر شده کفایت نمی‌داد. به همین خاطر روش‌های بهتری برای مدیریت شبکه مورد نیاز شد. اولین تلاش‌ها در RFC 1028 و RFC 1067 نمایان شد اما عمر چندانی نکرد. در ماه مه سال 1990 میلادی، RFC 1057 منتشر شد که نسخه یک SNMP را تعریف کرد. به همراه RFC 1155، SNMP روش سیستماتیکی را برای کنترل و مدیریت شبکه‌های کامپیوتری فراهم می‌آورد. این چهارچوب (framework) و پروتکل به شکل گسترده‌ای در محصولات تجاری پیاده‌سازی شده و به استاندارد عملی برای مدیریت شبکه تبدیل شده است.

اگر چه یکی از مهم‌ترین اهداف SNMP، سادگی است اما با این حال مدل SNMP برای مدیریت شبکه از 4 جزء تشکیل شده است:

۱ـ گره‌های مدیریتی ۲ـ ایستگاه‌های مدیریتی ۳ـ اطلاعات مدیریتی ۴ـ یک پروتکل مدیریتی

مدل SNMP در شکل ۱۰ نشان داده شده است.



شکل ۱۰: اجزای مدل مدیریتی SNMP

گره‌ی مدیریتی می‌تواند مسریاب، هاب، پل، چاپگر و یا هر وسیله دیگری باشد که بتواند وضعیت ارتباطی را به دنیای بیرون منتقل کند. برای کنترل مستقیم توسط SNMP، گره باید قادر باشد تا پروسه مدیریت SNMP را که عامل SNMP (SNMP agent) نام دارد، اجرا نماید. کلیه کامپیوترها با افزایش پل‌ها، مسیریاب‌ها و وسایل جانبی که برای استفاده شبکه طراحی شده‌اند، به چنین نیازمندی‌هایی احتیاج پیدا خواهند نمود. هر عامل، پایگاه داده‌ی محلی از متغیرهایی را نگهداری می‌کند که وضعیت، تاریخچه (history) و عملکرد آن را تشریح می‌نمایند.

مدیریت شبکه از ایستگاه‌های مدیریتی انجام می‌شود که در حقیقت، کامپیوترهای چندمنظوره هستند که نرم‌افزارهای مدیریتی خاصی را اجرا می‌کنند. ایستگاه‌های مدیریتی شامل یک یا چند پروسس هستند که با سایر عامل‌ها در کل شبکه در ارتباط هستند و به این وسیله فرامین را صادر و پاسخ‌ها را دریافت می‌کنند.

بسیاری از ایستگاه‌های مدیریتی دارای واسط کاربر گرافیکی هستند که به مدیر شبکه این اجازه را می‌دهند تا از وضعیت شبکه مطلع شده و در زمان مناسب، عملیات مقتضی را انجام دهد. برای آنکه ایستگاه‌های مدیریتی با تمام این اجزای گوناگون صحبت کنند، باید ماهیت اطلاعاتی که در کلیه این تجهیزات ذخیره می‌شود، کاملاً مشخص باشد. SNMP اطلاعات دقیقی را که هر عامل باید نگهداری کند و قالبی (format) که آن را پشتیبانی می‌کند، تشریح می‌کند.

به طور ساده هر وسیله، یک یا تعداد بیشتری متغیر را که حالت و وضعیت آن را تشریح می‌کند، نگهداری می‌کند. به تعبیر SNMP، این متغیرها، **شئ(objects)** نامیده می‌شوند. البته این نامگذاری ممکن است باعث اشتباه شود چراکه آن‌ها در حقیقت از نظر سیستم‌های شیءگرایی، شئ به شمار نمی‌روند زیرا آن‌ها تنها حالت(state) هستند و نه روش. مجموعه‌ای از کل اشیا در ساختار داده، **MIB (Management Information Base)** نامیده می‌شود. ایستگاه‌های مدیریتی توسط پروتکل SNMP با عامل‌ها در تعامل هستند. این پروتکل به ایستگاه مدیریتی این امکان را می‌دهد که حالات یک عامل را مورد پرسش خود قرار دهد و در صورت لزوم آن‌ها را تغییر دهد. اغلب SNMP ها، شامل این نوع ارتباط پرسش-پاسخ می‌باشند. تجهیزات قدیمی یا آن‌هایی که اساساً برای کار در شبکه طراحی نشده‌اند ممکن است فاقد این توانایی باشند. SNMP برای کنترل آن‌ها ، مفهومی به نام **عامل پروکسی(proxy agent)** را تعریف کرده است. عامل پروکسی، عاملی است که یک یا تعداد بیشتری از دستگاه‌های غیر SNMP را که ممکن است از پروتکل‌های غیراستاندارد استفاده کنند، مراقبت کرده و ارتباط آن‌ها را با یک ایستگاه مدیریت برقرار می‌سازند.

نکته۲: امنیت و تصدیقِ هویت نقش مهمی در SNMP ایفا می‌کنند.

سیستم نام دامنه یا DNS (Domain Name System)

زمانی که هزاران ایستگاه کاری به شبکه متصل می‌شوند کاملاً مشخص است که این روند به دلایل مختلف نمی‌تواند برای همیشه ادامه پیدا کند. یکی از مهمترین مشکلات به وجود آمده، می‌تواند تصادم نام باشد که ممکن است مرتباً رخ دهد مگر آنکه نام‌ها به طور متمرکز کنترل شوند. برای کنترل نام و حوزه‌های اینترنتی، DNS اختراع شده است. اساس و فلسفه DNS، ایجاد ساختار سلسله مراتبی برای نام حوزه‌ها و سیستم پخش شده برای **پیاده سازی** این الگوی نامی است. از DNS به طور گسترده برای تبدیل اسامی میزبان‌ها و مقاصد ایمیل به آدرس‌های IP استفاده می‌شود اما با این حال، برای اهداف دیگر نیز کاربرد دارد. DNS در RFC 1034 و RFC 1035 تشریح شده است.

به طور خلاصه می‌توان گفت که از DNS برای تبدیل نام به آدرس IP استفاده می‌شود. یک برنامه کاربردی، یک رویه کتابخانه‌ای را که resolver نام دارد، فراخوانی می‌کند و نام حوزه را به‌عنوان پارامتر داخل آن ارسال می‌کند. resolver، بسته UDP را به سرور DNS محلی ارسال می‌کند. نام مورد نظر جستجو شده و آدرس IP متناظر آن به resolver و در نهایت به فراخوانی کننده، برگشت داده می‌شود. با داشتن آدرس IP، برنامه می‌تواند یک ارتباط TCP را با مقصد تشکیل داده یا به ارسال بسته UDP مشغول شود.

مثال۳: از ..... برای تبدیل نام به آدرس استفاده می‌شود.

۱) EBS ۲) DNS ۳) BSS ۴) EBSS

☑ **پاسخ:** گزینه «۲» از DNS برای تبدیل نام به آدرس استفاده می‌شود.

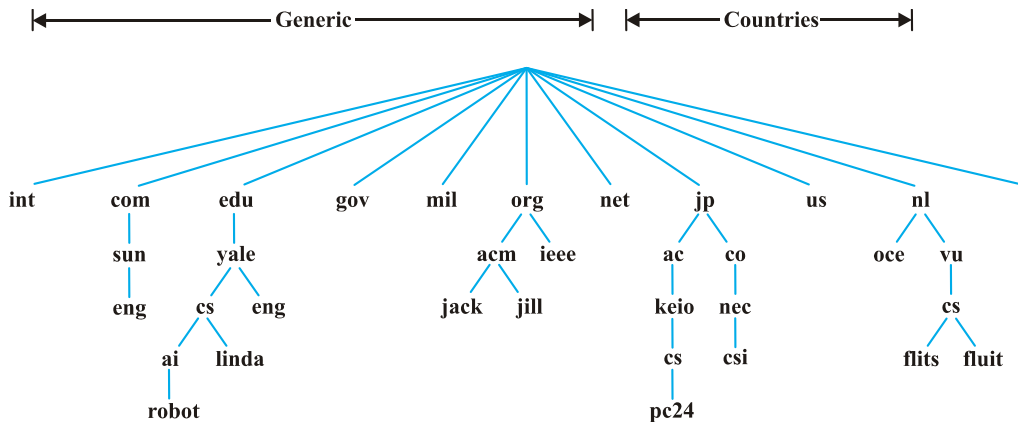
فضای نام DNS (DNS Name Space)

مدیریت مجموعه بزرگی از نام‌ها که مدام نیز در حال تغییر هستند کار کوچکی نیست. از لحاظ مفهومی، اینترنت به چند صد **دامنه (Domains)** تقسیم می‌شود که هر یک از آن‌ها تعداد زیادی میزبان را پوشش می‌دهند. هر دامنه به زیردامنه‌هایی تقسیم می‌شود که خود این زیردامنه‌ها نیز به نوبه خود به قسمت‌های کوچکتری تقسیم می‌شوند. همانطور که شکل ۱۱ نشان داده است، کل این دامنه‌ها را می‌توان با استفاده از درخت نشان داد. برگ‌های درخت، دامنه‌هایی را نشان می‌دهد که خود دارای زیردامنه دیگری نیستند. دامنه برگی ممکن است یک میزبان تنها باشد یا شرکتی باشد که شامل هزاران میزبان است.

دامنه‌هایی که در بالاترین سطح قرار دارند (top-level domains)، به دو نوع عمومی (generic) و کشوری (country) تقسیم می‌شوند. دامنه‌های عمومی عبارتند از: com (تجاری)، edu (موسسات آموزشی)، gov (دولت فدرال ایالات متحده) و int (سازمان‌های بین‌المللی خاص). دامنه کشور، شامل مداخلی است که برای هر کشور متفاوت می‌باشد (ر.ک به ISO 3166).

نام‌گذاری هر دامنه از مسیری که از بالا تا ریشه به وجود آمده است، صورت می‌گیرد. هر جزء و قسمت با نقطه (که dot خوانده می‌شود) از دیگری جدا می‌شود. به‌عنوان مثال به یکی از آدرس‌های ایمیل مولف توجه کنید: o_mohabati@iust.ac.ir به دامنه iust.ac.ir دقت کنید. با توجه به این دامنه می‌توان متوجه شد که دامنه مذکور، مربوط به دانشگاه علم و صنعت ایران (iust.) است. دامنه ac. نشان می‌دهد که دامنه، مربوط به یک مرکز آکادمیک و تحصیلات عالی است. دامنه ir. نیز کشور ایران را مشخص می‌کند. (توضیح مشابهی را برای o.mohabati@yahoo.com بیان کنید)

نام‌های دامنه‌ها می‌توانند **مطلق (absolute)** یا **نسبی (relative)** باشند. یک دامنه مطلق با یک نقطه خاتمه پیدا می‌کند (مانند eng.sun.com) در حالی که دامنه‌های نسبی این چنین نیستند. دامنه‌های نسبی باید به شکل واحد به گونه‌ای تفسیر شوند تا مفهوم دقیق آن‌ها مشخص شود. در هر دو حالت، یک نام حوزه در درخت، به یک گره مشخص و کلیه گره‌های زیر آن اشاره می‌کند.



شکل ۱ / ۱: بخشی از فضاهای نام دامنه اینترنت

نام‌های دامنه، نسبت به کوچک یا بزرگ بودن حروف حساس نیستند؛ بنابراین edu و EDU یکسان قلمداد می‌شوند. نام اجزا می‌تواند طولی برابر با 63 کاراکتر داشته باشد. نام مسیر کامل(full path name)، نباید از 255 کاراکتر فراتر رود. دامنه‌ها را می‌توان به دو طریق داخل درخت درج کرد. برای مثال، cs.yale.edu می‌تواند تحت دامنه کشوری US به شکل cs.yale.ct.us درج شود. هر دامنه، نحوه تخصیص فضاهای زیردست خود را کنترل می‌کند. برای مثال کشور ژاپن ممکن است دامنه‌های ac.jp و co.jp را برای خود در نظر گیرد در حالی که شاید کشور دیگری مانند هلند، سازمان‌های خود را تحت nl قرار دهد.

برای ایجاد دامنه‌های جدید نیاز به اخذ مجوز از دامنه‌های بالاتر می‌باشد. به‌عنوان مثال اگر گروه VLSI در Yale شروع به کار کرده باشد و خواستار فضای VLSI.cs.yale.edu, باشد، ابتدا باید از فردی که مدیریت cs.yale.edu را برعهده دارد، مجوز کسب کند.

رکوردهای منابع (Resource records)

هر دامنه؛ چه یک میزبان منفرد و چه دامنه‌ی سطح بالا، می‌تواند مجموعه‌ای از رکوردهای منابع مختص به خود را داشته باشد. برای یک میزبان منفرد، معمول‌ترین رکورد منبع، فقط آدرس IP می‌باشد اما برای انواع دیگر، رکوردهای منابع بسیاری وجود دارد. زمانی که resolver، نام دامنه را به DNS تحویل می‌دهد، آنچه که برگشت داده می‌شود، رکوردهای منابع مربوط به آن نام هستند. به همین دلیل، مهم‌ترین وظیفه DNS تبدیل نام‌های دامنه به رکوردهای منابع است.

رکورد منبع شامل پنج قسمت به قرار زیر است: نام دامنه (Domain-name)، Time-to-live، نوع (Type)، کلاس (Class)، ارزش (Value)

نام دامنه (Domain-name)

نام دامنه، دامنه مربوط به رکورد را نشان می‌دهد. معمولاً رکوردهای بسیاری برای هر دامنه وجود دارند که هر کپی از پایگاه داده، حاوی اطلاعاتی درباره چند دامنه است. به همین خاطر، این فیلد دارای نقشی اساسی در کلید جستجو برای پاسخ به پرسش‌ها می‌باشد.

Time-to-live

این فیلد بر چگونگی ایستایی رکورد دلالت می‌کند. اطلاعاتی که به شدت ثابت هستند، مقدار بزرگی مانند 86400 (تعداد ثانیه‌های یک روز) می‌گیرند. از آن طرف، اطلاعاتی که به شدت فرار هستند، مقادیر کوچکتری را می‌پذیرند مانند 60 (یک دقیقه).

نوع (Type)

فیلد نوع، نوع رکورد را نشان می‌دهد. مهمترین انواع در جدول ۳ درج شده‌اند.

جدول ۳: مهمترین انواع منابع DNS

نوع	مفهوم	ارزش
SOA	آغاز تصدیق هویت (Start of Authority)	پارامترهای مربوط به این ناحیه (zone)
A	آدرس IP میزبان	عدد صحیح ۳۲ بیتی
MX	تبادل ایمیل (Mail exchange)	تمایل دامنه برای دریافت ایمیل
NS(Name serve)	سرور نام (Name serve)	نام سرور مربوط به این دامنه
CNAME	نام رسمی (Canonical name)	نام دامنه
PTR	اشاره‌گر (Pointer)	نام مستعار (Alias) آدرس IP
HINFO	توصیف میزبان (Host description)	متن ASCII متوالی و غیر منقطع
TXT	متن	متن ASCII

کلاس (Class)

چهارمین فیلد از هر رکورد منبع، فیلد کلاس می‌باشد. برای اطلاعات اینترنتی این فیلد همیشه IN است. برای اطلاعات غیر اینترنتی مقادیر دیگری می‌تواند مورد استفاده قرار گیرد.

ارزش (Value)

در نهایت به فیلد ارزش می‌رسیم. این فیلد می‌تواند یک عدد، نام دامنه یا رشته ASCII باشد که این بستگی به نوع رکورد دارد. شرح مختصری از فیلد ارزش برای هر نوع داده اصلی در جدول ۳ آورده شده است.

سرویس رکوردها (Records' Service)

۱ـ SOA: یک رکورد SOA، نام منبع اصلی اطلاعاتی همچون نام ناحیه سرور (name server's zone)، آدرس ایمیل مدیر آن، یک عدد سریال، فلگهای مختلف و timeout تامین می‌کند.

۲ـ A: بر آدرس رکورد دلالت می‌کند. حاوی آدرس IP 32 بیتی برای برخی از میزبان‌ها می‌باشد.

۳ـ MX: نام دامنه‌ای را که برای دریافت ایمیل آماده است، مشخص می‌کند.

۴ـ NS: رکورد NS، نام سرور را مشخص می‌کند.

مثال۴: دامنه‌هایی که در بالاترین سطح قرار دارند(top-level domains)، به چند دسته تقسیم می‌شوند؟

۱) 4 ۲) 3 ۳) 2 ۴) 1

☑ **پاسخ:** گزینه «۳» دامنه‌هایی که در بالاترین سطح قرار دارند (top-level domains)، به دو نوع عمومی(generic) و کشوری(country) تقسیم می‌شوند.

مثال۵: رکوردهای منبع(Resource records) شامل چند قسمت هستند؟

۱) 5 ۲) 3 ۳) 4 ۴) 2

☑ **پاسخ:** گزینه «۱» رکورد منبع شامل پنج قسمت به قرار زیر است:

نام دامنه (Domain-name)، Time-to-live، نوع(Type)، کلاس (Class)، ارزش (Value)

سرویس‌های لایه کاربرد (Application Layer Services)

لایه کاربرد وظیفه تامین این سرویس‌ها را بر عهده دارد:

۱ـ شناسایی طرفین ارتباط و ایجاد دسترس پذیری
۲ـ تصدیق هویت
۳ـ اختصاص هزینه‌ها
۴ـ به توافق رساندن منابع در دسترس
۵ـ پذیرش سرویس‌ها
۶ـ همزمان‌سازی برنامه‌ها
۷ـ تصحیح خطا
۸ـ انتخاب محاوره
۹ـ بررسی صحت و یکپارچگی داده
۱۰ـ درخواست فایل و ارسال آن
۱۱ـ رویه‌های گرافیکی
۱۲ـ درخواست‌های بانک اطلاعاتی، درج و حذف
۱۳ـ سرویس پایانه مجازی
۱۴ـ ...

مطالعه ابزار اینترنت (Internet Tools)

اکنون به بررسی برخی ابزار اینترنت مانند ایمیل، WWW، HTTP و HTML می‌پردازیم.

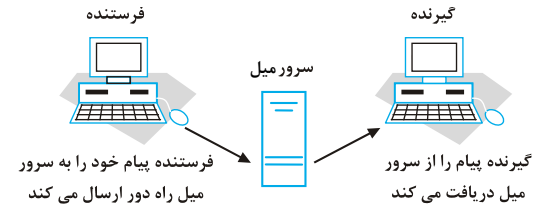
ایمیل (E-mail)

شاید بی‌راه نباشد اگر بگوییم که رایج‌ترین استفاده از اینترنت به پست الکترونیکی که ایمیل نامیده می‌شود، بر می‌گردد. با استفاده از ایمیل، کاربر می‌تواند به ارسال متن، تصویر، فایل‌های صوتی، فایل‌های برنامه‌ای، فیلم و... به هر فرد دیگری که در هر جای جهان قرار دارد، بپردازد. استفاده از ایمیل دو امتیاز دارد:

۱ـ سرعتی که در آن اطلاعات ارسال می‌شوند (تقریباً فوری).

۲ـ هزینه آن، صرف نظر از فاصله و مسافت جغرافیایی است.

هر کاربر ایمیل شبکه باید دارای آدرس ایمیل منحصر به فردی باشد که دامنه‌ی نامش با علامت @ جدا می‌شود. شکل ۱۲، انتقال پیام ایمیل در اینترنت را نشان می‌دهد.



شکل ۱۲: انتقال پیام ایمیل در میان شبکه اینترنت

برای ارسال یک ایمیل، مراحل زیر طی می‌شود:

۱ـ فرستنده توسط نرم‌افزاری که مِیل کلاینت (mail client) نام دارد، پیام ایمیل را تشکیل می‌دهد. میل کلاینت، به کاربر اجازه می‌دهد تا پیام ایمیل خود را ایجاد، ویرایش و ارسال کند. برخی از نرم‌افزارهای میل کلاینت موجود عبارتند از: Netscape Mail، Outlook express، Eudora، Pine و غیره.

۲ـ پس از تشکیل و ایجاد ایمیل، کاربر آن را به آدرس ایمیل گیرنده ارسال می‌کند. پیام در میان اینترنت پیش می‌رود تا به مِیل سرور گیرنده برسد.

۳ـ گیرنده به حساب ایمیل خود بر روی میل سرور متصل شده و پیامی که به او ارسال شده را می‌خواند. گیرنده از میل کلاینت برای دریافت، ذخیره و چاپ پیام ایمیل استفاده می‌کند.

پروتکل‌های ایمیل اینترنت

ایمیل اینترنتی بر پایه استانداردهایی همچون SMTP، POP، IMAP و MIME قرار دارد. این استانداردها در حقیقت همان پروتکل‌های ایمیل را تشکیل می‌دهند که در ادامه مورد بررسی قرار می‌گیرند.

پروتکل انتقال ایمیل ساده یاSMTP (Simple Mail Transfer Protocol)

پروتکل استانداردی که برای انتقال ایمیل مابین کامپیوترها استفاده می‌شود، SMTP است. SMTP قالبی که مطابق آن پیام ایمیل ایجاد می‌شود را مشخص می‌کند. یک پیام ایمیل اینترنتی شامل دو بخش است: سرآیند و بدنه. سرآیند پیام، حاوی آدرس گیرنده، آدرس فرستنده، موضوع و سایر اطلاعاتی درباره پیام مانند تاریخ، زمان ارسال، نوع مِیل کلاینتی که فرستنده از آن استفاده کرده و ... می‌باشد. بدنه پیام نیز حاوی اصل پیام است.

POP (Post Office Protocol)

پروتکل POP مشخص می‌کند که چگونه میل کلاینت‌ها می‌توانند پیام‌ها را از میل سرورها بازیابی نمایند. POP برای کامپیوترهای تک کاربره ایجاد شده است. سه نسخه از این پروتکل وجود دارد: POP، POP2 و POP3.

IMAP (Internet Message Access Protocol)

هدف این پروتکل که در سال 1986 در دانشگاه Stanford ابداع شد، بازیابی پیام‌های ایمیل است. نسخه چهار این پروتکل یعنی IMAP4، شبیه POP3 است اما از ویژگی‌های بیشتری پشتیبانی می‌کند. برای مثال توسط IMAP4 شما می‌توانید در میان پیام ایمیل خود در حالیکه همچنان در میل سرور است، عمل جستجو و سرچ کلمات کلیدی را انجام دهید. سپس شما می‌توانید انتخاب کنید که کدام پیام در ماشین شما دانلود شود. مانند POP، IMAP از SMTP برای ارتباط بین میل کلاینت و میل سرور استفاده می‌کند.

MIME (Multipurpose Internet Mail Extensions)

از SMTP تنها می‌توان برای ارسال پیام‌هایی که با کاراکترهای ASCII ایجاد شده‌اند استفاده کرد. این امر، استفاده از ایمیل را محدود می‌کند. پروتکل دیگری به نام MIME وجود دارد که می‌تواند به مبادله پیام‌های غیر متنی مانند گرافیک، صوت و سایر فایل‌های چندرسانه‌ای بپردازد. هر زمان که شما بخواهید یک فایل غیرمتنی مانند صفحه گسترده (spreadsheet)، فایل برنامه، فایل گرافیکی یا فایل صوتی را ارسال کنید، می‌توانید این فایل را با استفاده از MIME رمزنگاری نمایید. MIME این فایل‌ها را به شکل و قالب متنی رمزنگاری کرده و در نتیجه می‌توانند با استفاده از SMTP ارسال شوند. گیرنده نیز می‌تواند این داده رمزنگاری شده MIME را به فایل اصلی غیر متنی برگرداند.

اغلب ایمیل کلاینت‌ها مانند Netscape به طور خودکار پیام ایمیل را که حاوی داده‌های غیر متنی است، رمزگذاری و رمزگشایی می‌کنند. با این حال برخی ایمیل کلاینت‌ها هم وجود دارند که خود شما باید داده را با استفاده از ابزار رمزنگاری، رمزنگاری کنید(مانند Infoxfer) .

WWW (World Wide Web)

واژه WWW، خلاصه شده کلمات «وب جهان گسترده» (World Wide Web) می‌باشد. تعریف رسمی وب جهان گسترده به این صورت است: اطلاعات فرارسانه‌ای (hypermedia) قابل بازیابی که به هدف دسترسی عمومی به حجم وسیعی از مستندات می‌باشد.

واژه فرارسانه‌ای از واژه فرامتنی (hypertext) مشتق شده است. فرامتن، متنی است که ماهیت غیر ترتیبی (nonsequential) و غیر خطی(non-linear) دارد. توسط متون hypertext، میان موضوعات مشابه و مرتبط با هم که می‌توانند تنها حاوی متن نباشند (مثلاً صوت و تصویر)، اتصال ایجاد می‌شود. می‌توان گفت که WWW روش یکدستی را برای دسترسی به حجم عظیمی از اطلاعات در بستر شبکه فراهم می‌آورد.

Tim Berners-lee در سال 1989 مفهوم WWW را مطرح کرد. طرح اولیه WWW به دو هدف انجام گرفت: پیشرفت علم و امور آموزشی. اما WWW علاوه بر آنکه باعث پیشرفت در این دو حوزه شد، تاثیر عمیقی نیز بر سایر موارد همچون تجارت، سیاست و .. گذاشت. استاندارد جهانی یکسانی در مورد WWW وجود ندارد. اما با این حال اکثر ابزارها به شکل مشابهی فعالیت می‌کنند. جستجوگرهای اینترنتی گوناگونی مانند Microsoft Internet Explorer، Opera، Netscape Navigator، Mosaic، Google Chrome و ... وجود دارند که می‌توانند برای دسترسی به اطلاعات موجود در روی اینترنت از آن‌ها استفاده کرد. کلیه این جستجوگرها، ساختار گرافیکی دارند. علاوه بر این، اغلب جستجوگرهای جدید می‌توانند اطلاعات را به شکل صوت و تصویر ارائه دهند.

مستنداتی که در صفحه مانیتور نمایش داده می‌شود معمولاً **صفحه یا page** نامیده می‌شود. برخی از مستندات خاص را طراحِ صفحه ترجیح می‌دهد به گونه‌ای تنظیم کند که کاربر در اولین نگاه آن‌ها را مشاهده کند. نام این اطلاعات که **صفحه خانگی (home page)** نامیده می‌شود معمولاً با سایت، شخص یا مورد مشابهی در ارتباط است. محلی که صفحات وب در آن ذخیره می‌شوند، **وب سایت (web site)** نامیده می‌شود. دسترسی به وب سایت از طریق تایپ نام آن امکان پذیر است.

سیستم WWW بر پایه معماری کلاینت-سرور بنا شده است. از یک وب کلاینت برای ارسال درخواست اطلاعات به وب سروری که اطلاعات خواسته شده را در خود ذخیره کرده است، استفاده می‌شود. وب سرور برنامه‌ای است که به محض دریافت درخواست، سند خواسته شده را به کلاینت برمی‌گرداند.

وب کلاینت‌ها و وب‌سرورها توسط پروتکلی به نام Hypertext Transfer Protocol که HTTP نیز نامیده می‌شود می‌توانند با یکدیگر در ارتباط باشند. کلیه وب‌کلاینت‌ها و وب سرورها باید از HTTP برای ارسال و دریافت مستندات hypermedia استفاده کنند. به همین دلیل وب‌سرورها اغلب اوقات **HTTP سرور(HTTP server)** نیز نامیده می‌شوند.

صفحات وب و HTML (Hypertext Markup Language)

صفحه وب یک سند hypermedia می‌باشد. HTML زبان استانداردی است که از آن برای ایجاد صفحات وب استفاده می‌شود.HTML تعدادی دستور و فرمان را برای ایجاد مکان و قالب متن، عکس و صوت در روی صفحات وب فراهم می‌آورد.

مستندات وب معمولاً به زبان HTML نوشته می‌شوند که کاربرد ساده‌ای دارند به طوری که هر ویرایشگر متنی مانند Notpad ، گزینه مناسبی برای نوشتن آن‌ها به شمار می‌رود. مستندات وب معمولاً با پسوند html. یا htm. نامگذاری می‌شوند. لذا مستندات HTM ، فایل‌های ASCII با قالب بندی خاصی هستند که شامل اطلاعاتی در باره طرح واره (layout) و لینک‌ها می‌باشند. پس از وارد کردن کد HTML در یک فایل، شما می‌توانید آن را در یک جستجوگر ملاحظه کنید. جستجوگرها به طور خودکار کد HTML مستندات را به شکل مناسبی تفسیر می‌کنند.

URL (Universal Resource Locators)

وبِ جهان گستر (WWW) از URLها برای نشان دادن لینک‌های hypermedia و ایجاد ارتباط به سرویس‌های درون مستندات HTML استفاده می‌کند. تقریباً هر فایل یا سرویسی بر روی اینترنت را می‌توان با استفاده از یک URL، مانند نمونه زیر نمایش داد:

http://www.microsoft.com

قسمت اولِ URL (قبل از دو اِسلش(//))، روش و پروتکل دسترسی را نشان می‌دهد. دومین قسمت معمولاً آدرس کامپیوتری است که بر روی آن اطلاعات یا سرور قرار دارند. قسمت‌های دیگر ممکن است به اسامی فایل‌ها، پورتی که به آن اتصال وجود دارد و ... اشاره کنند. هر URL تنها یک خط است و هیچ فاصله‌ای (space) بین آن وجود ندارد.

مثال۶: رکوردهای منبع(Resource records) شامل چند قسمت هستند؟

۱) 5 ۲) 3 ۳) 4 ۴) 2

☑ **پاسخ:** گزینه «۱» رکورد منبع شامل پنج قسمت به قرار زیر است:

نام دامنه(Domain-name)، Time-to-live، نوع(Type)، کلاس(Class)، ارزش(Value)

پروتکل انتقال فرامتن یا HTTP(HyperText Transfer Protocol)

پروتکل انتقال فرامتن یا همان HTTP(HyperText Transfer Protocol)، به عنوان پروتکل لایه کاربرد وب، وظیفه مهمی در وب ایفا می‌کند به طوری که می‌توان آن را «قلب وب» درنظر گرفت. HTTP که RFC 1945 و RFC 2616 تعریف شده است، در دو برنامه پیاده‌سازی می‌شود: برنامه کلاینت و برنامه سرور. برنامه کلاینت و برنامه سرور در دو سیستم انتهایی متفاوت اجرا می‌شوند و توسط تبادل پیام‌های HTTP با یکدیگر «صحبت» می‌کنند. ساختار این پیام‌ها و همچنین نحوه تبادل پیام‌ها توسط کلاینت و سرور توسط HTTP معین می‌شود. قبل از آنکه جزئیات بیشتری را در رابطه با HTTP توضیح دهیم، بهتر است با چند اصطلاح در وب آشنا شویم.

یک **صفحه وب (Web page)یا سند (document)**، از اشیای مختلفی تشکیل شده است. به طور ساده یک **شئ (object)** یک فایلی-مانند یک فایل HTML، یک تصویر JPEG، یک Java applet یا یک کلیپ ویدئویی-است که توسط یک URL مورد آدرس‌دهی قرار می‌گیرد. اغلب صفحات وب از یک **فایل HTML پایه (base HTML file)** و چندین شئ ارجاع داده شده تشکیل شده‌اند. به عنوان مثال اگر یک صفحه وب شامل متن HTML و پنج تصویر JPEG باشد آنگاه صفحه وب حاوی شش شئ است: فایل پایه HTML و پنج عدد تصویر. فایل پایه HTML به سایر اشیای موجود در صفحه، با URL مربوط به آن شئ اشاره می‌کند. هر URL از دو جزء تشکیل شده است: **نام میزبان(hostname)** سروری که دربرگیرنده آن اشیا است و **نام مسیر (path name)**مربوط به شئ. به عنوان مثال، یک URL همانند:

http://www.someSchool.edu/someDepartment/picture.gif