شبکه های کامپیوتری

استاد فیروزبخت

پاییز ۹۲

فهرست

شبکه

به مجموعه ای از کامپیوتر های مستقل و متصل به هم شبکه گفته می شود.
در سیستم های توزیع شده کامپیوتر ها متصل هستند ولی مستقل نیستند ولی در شبکه علاوه بر اینکه کامپیوتر ها متصل به یکدیگرند، از همدیگر مستقل نیز می باشند.

۱. اشتراک منابع، برنامه ها، داده ها و تجهیزات جانبی

(Resource, Programs, Data and Device Sharing)

۲. قابلیت اطمینان (به دلیل تعداد منابع)

(High Reliability)

۳. هزینه کم

(Saving Money)

بعد های مسافتی از بین می رود.

۴. Scalability

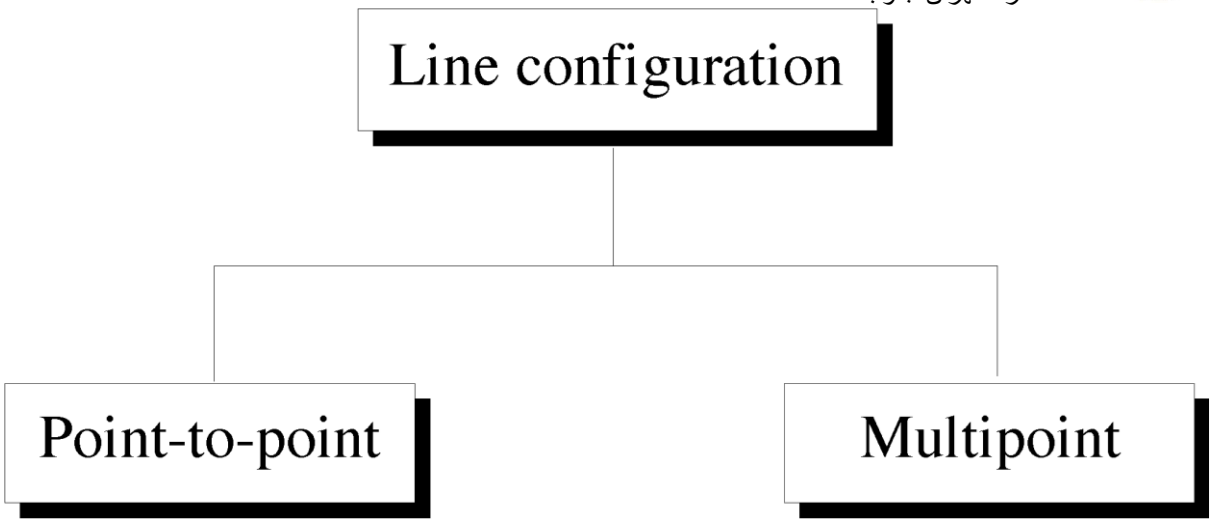
نیاز ها و سرویس ها متناسب با سخت افزار رسد پیدا کرده و تغییر می کند. در شبکه نیز وقتی یک بسته نرم افزاری ارتقاء پیدا می کند، نیاز ها و سرویس ها تغییر می کند.

۵.Communication Medium

افراد مختلف در نقاط مختلف به هزینه کم و سرویس خوب می توانند از طریق شبکه با یکدیگر ارتباط برقرار نمایند.

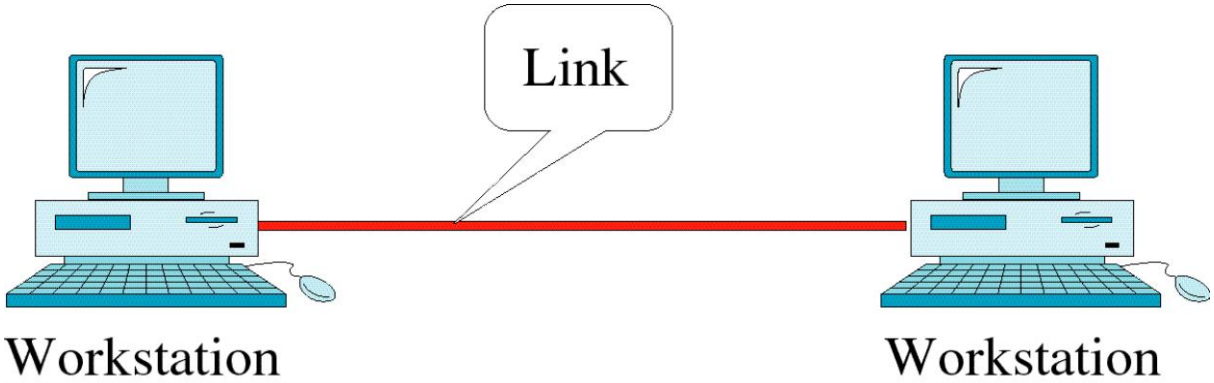
تعاریف پایه

انواع اتصال



شکل ۱: انواع پیکربندی خط

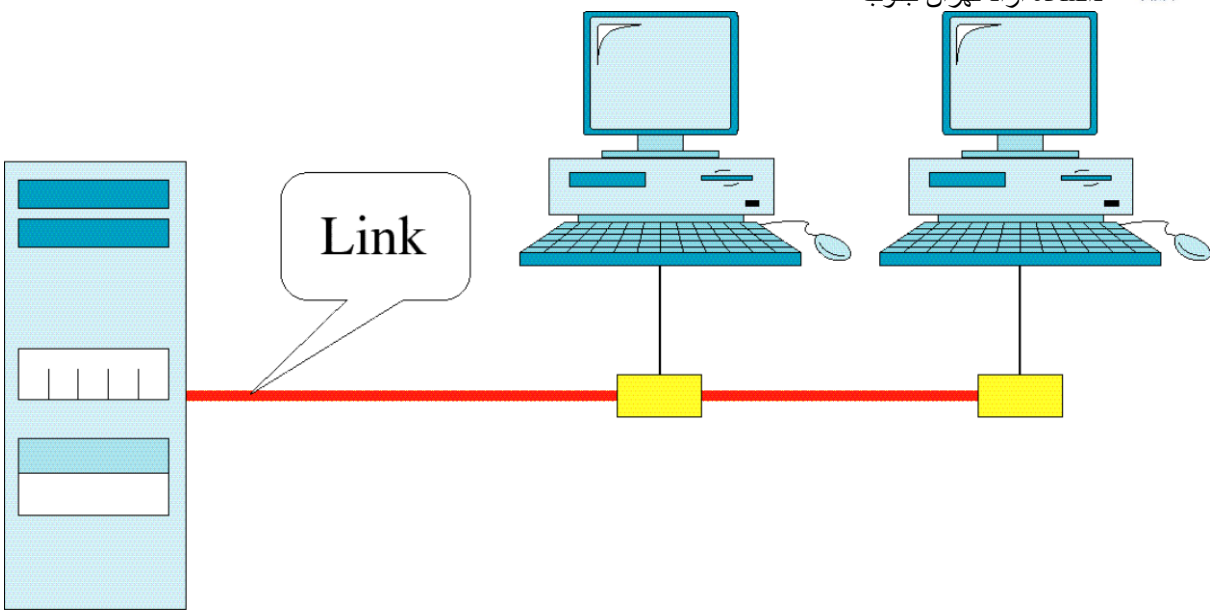
پیکربندی خطوط به دو صورت نقطه به نقطه و یا چند نقطه ای امکان پذیر است



شکل ۲: اتصال نقطه به نقطه

شکل ۳: اتصال نقطه به نقطه

در حالت چند نقطه ای چون محیط اشتراکی است بحث آدرس دهی مطرح است و اینکه چه کسی باید اطلاعات را بگیرد، بحث دیگری که مطرح است کنترل خط است و اینکه وقتی همه با هم صحبت می کنند، تداخل اطلاعاتی پیش نیاید.



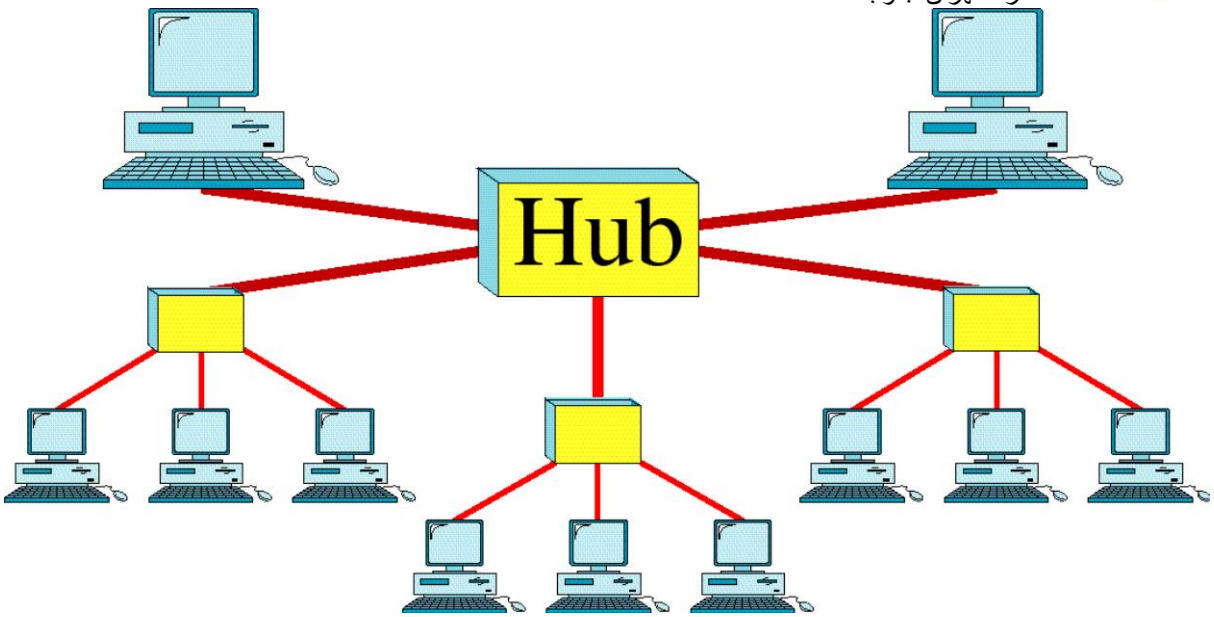
شکل ۴: اتصال چند نقطه ای

Topology

به نحوه قرار گیری کامپیوتر ها کنار یکدیگر توپولوژی گفته می شود

انواع مختلف توپولوژی

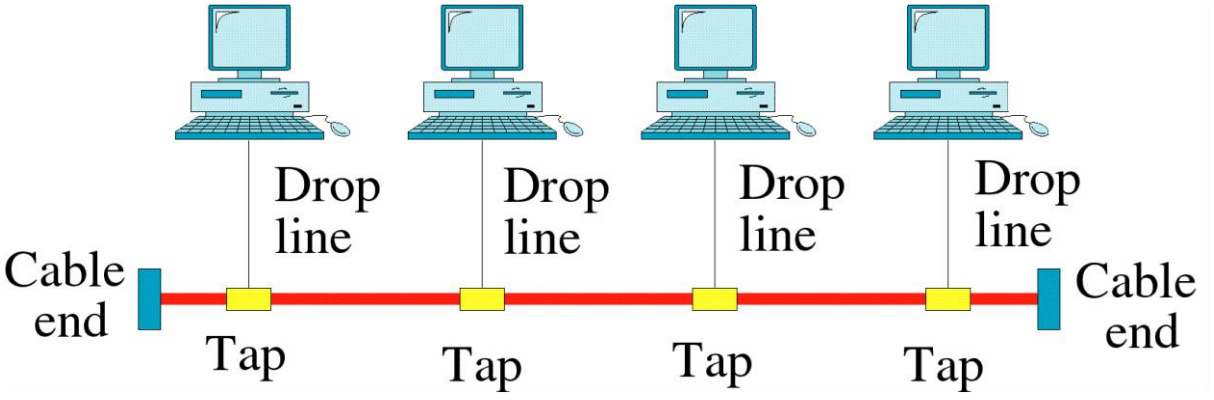
صفحه ۱۷



شکل ۷: توپولوژی درخت

توپولوژی باس

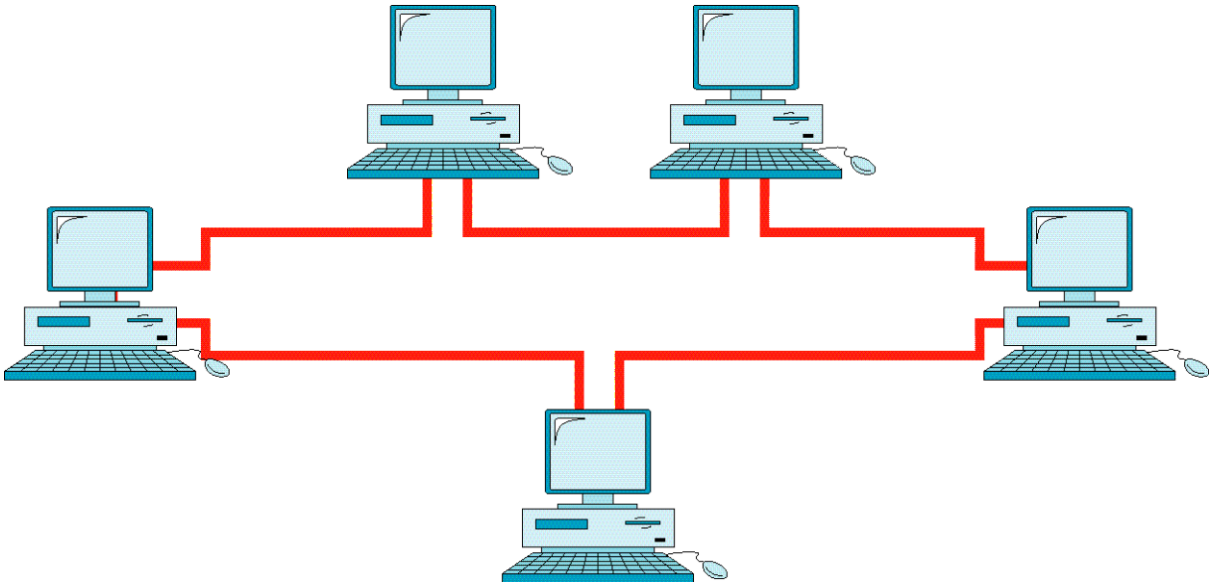
این توپولوژی مسنوخ شده است به دلیل اینکه در اینجا اگر یکی از ارتباطات قطع شود، کل شبکه از کار می افتد.



شکل ۸: توپولژی باس

توپولوژی حلقه

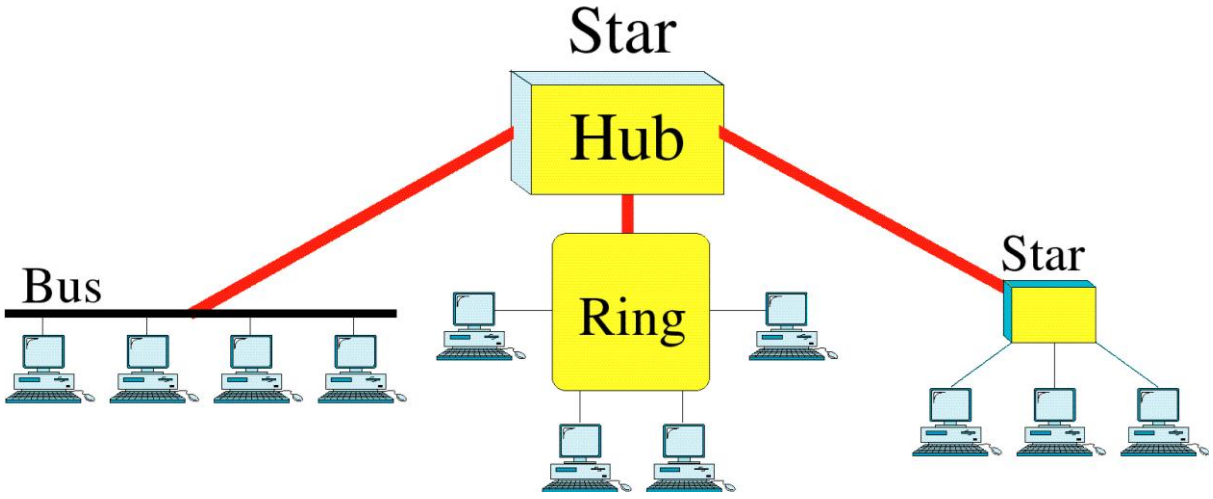
ارتباط کامپیوتر ها در این توپولوژی به صورت یک حلقه می باشد.



شکل ۹: توپولوژی حلقه

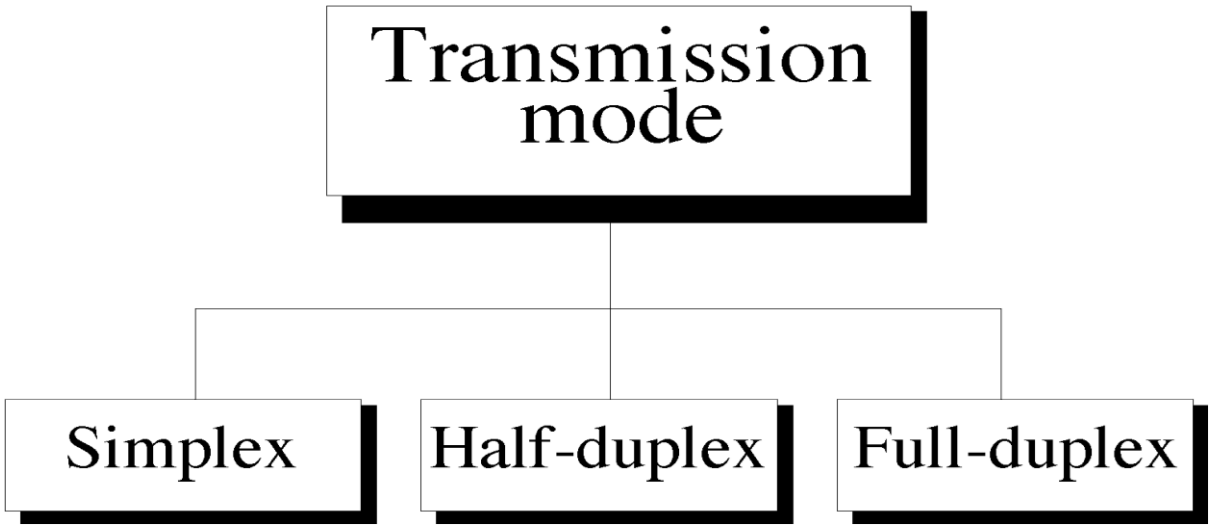
توپولوژی پیوندی

این توپولوژی ترکیبی از توپولوژی های مختلف می باشد.



شکل ۱۰: توپولوژی پیوندی یا هایبرید

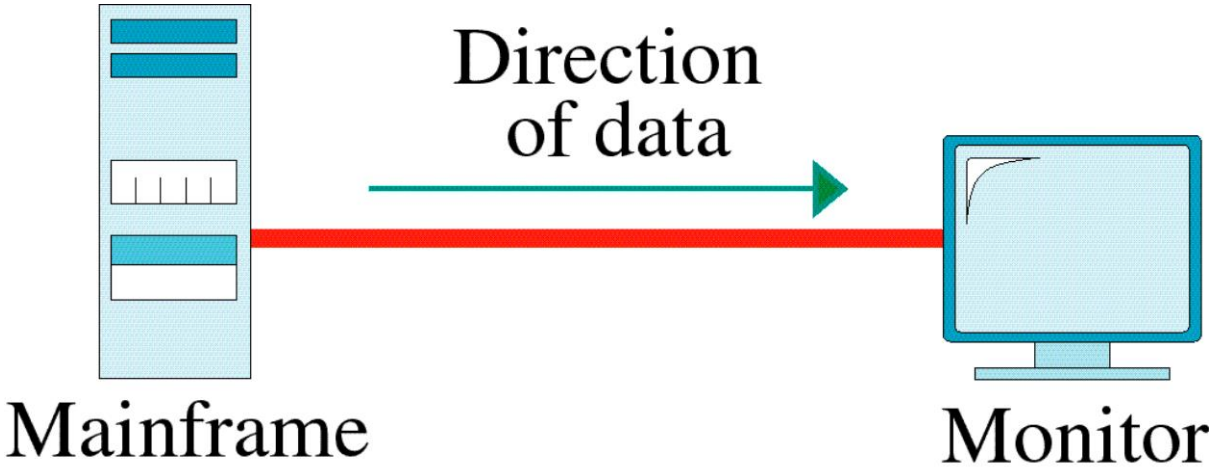
انواع حالت های انتقال



شکل ۱۱: حالت های انتقال

۱.ساده (Simplex)

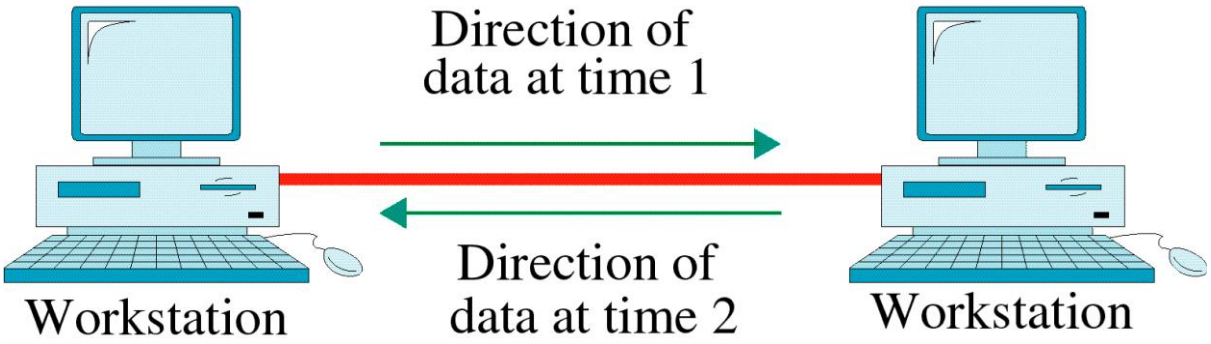
در این حالت ارتباط یک طرفه می باشد مانند ارسال اطلاعات از کامپیوتر به مانیتور



شکل ۱۲: حالت انتقال ساده

۲. یکطرفه (Half Duplex)

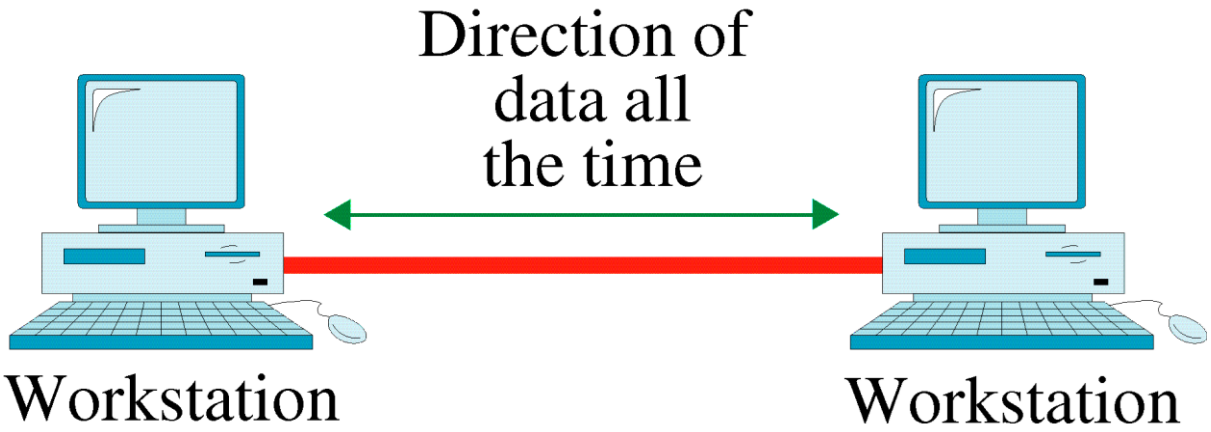
در این حالت ارتباط دو طرفه است ولی نه در یک زمان بلکه در زمان های متفاوت مانند بیسیم.



شکل ۱۳: حالت انتقال یکطرفه

۳. دو طرفه (Full Duplex)

این حالت ارتباط دو طرفه را در آن واحد امکان پذیر می سازد مانند تلفن.



شکل ۱۴: حالت انتقال دو طرفه

تقسیم بندی شبکه ها

۱. از دیدگاه تکنولوژی انتقال داده ها

۲. از دیدگاه جغرافیایی (فاصله ایستگاه ها)

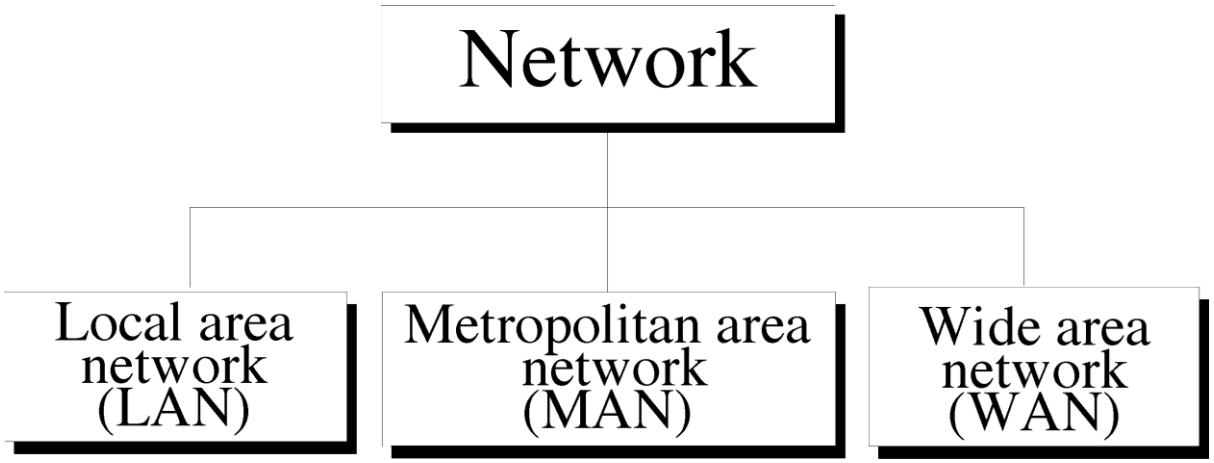
تقسیم بندی شبکه ها از دیدگاه تکنولوژی انتقال داده ها

۱. Point to Point (نقطه به نقطه)

۲. Broad Cast(پخش همگانی)

در حالت Broad Cast وقتی اطلاعات را ارسال می کنیم، همه دریافت می کنند.

تقسیم بندی شبکه ها از دیدگاه جغرافیایی

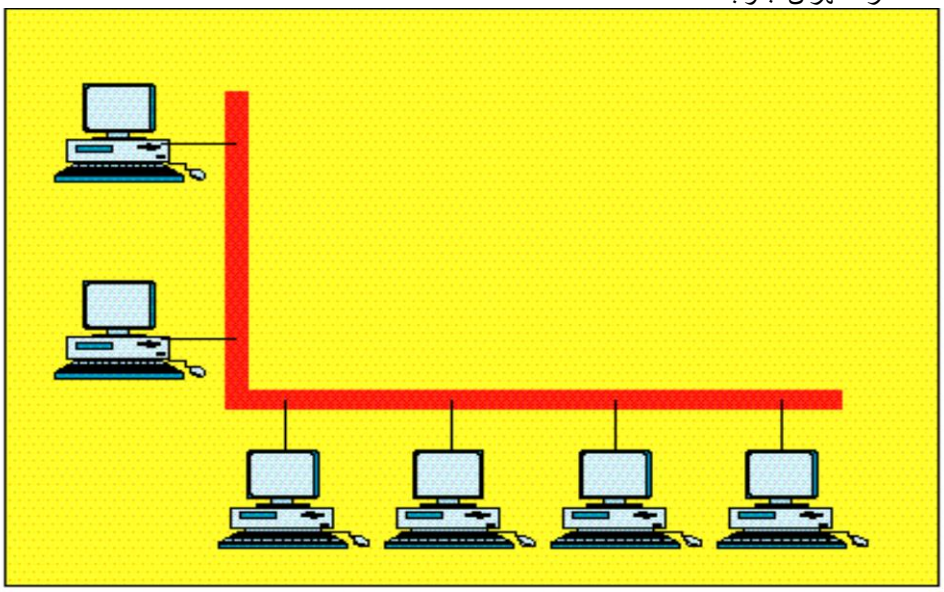


شکل ۱۵: تقسیم بندی شبکه ها از دیدگاه جغرافیایی

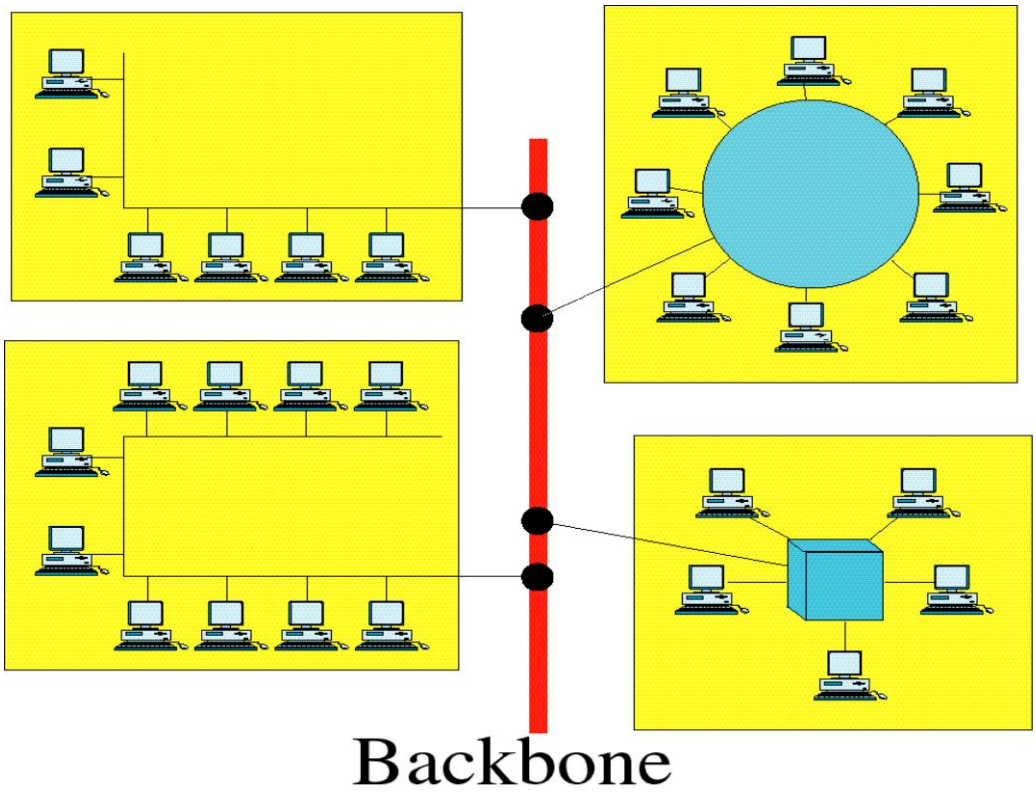
۱. Local Area Network (LAN)
۲. Metropolitan Area Network (MAN)
۳. Wide Area Network (WAN)

شبکه های LAN

شبکه هایی که با فاصله زیر یک کیلومتر (در یک یا چند ساختمان) می باشند. تکنولوژی این شبکه ها Broad Castاست که باعث می شود شبکه های LAN از نظر فاصله محدود شوند.



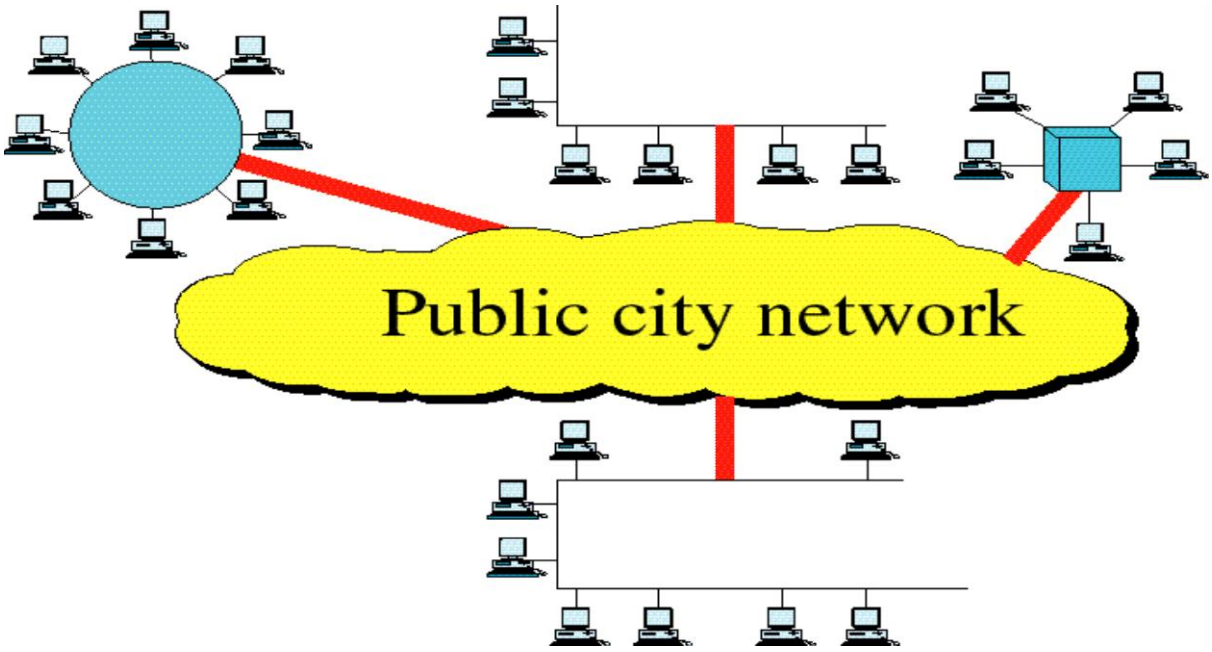
شکل ۱۶: یک نمونه شبکه LANساده



شکل ۱۷: یک نمونه شبکه LAN چندگانه

شبکه های MAN

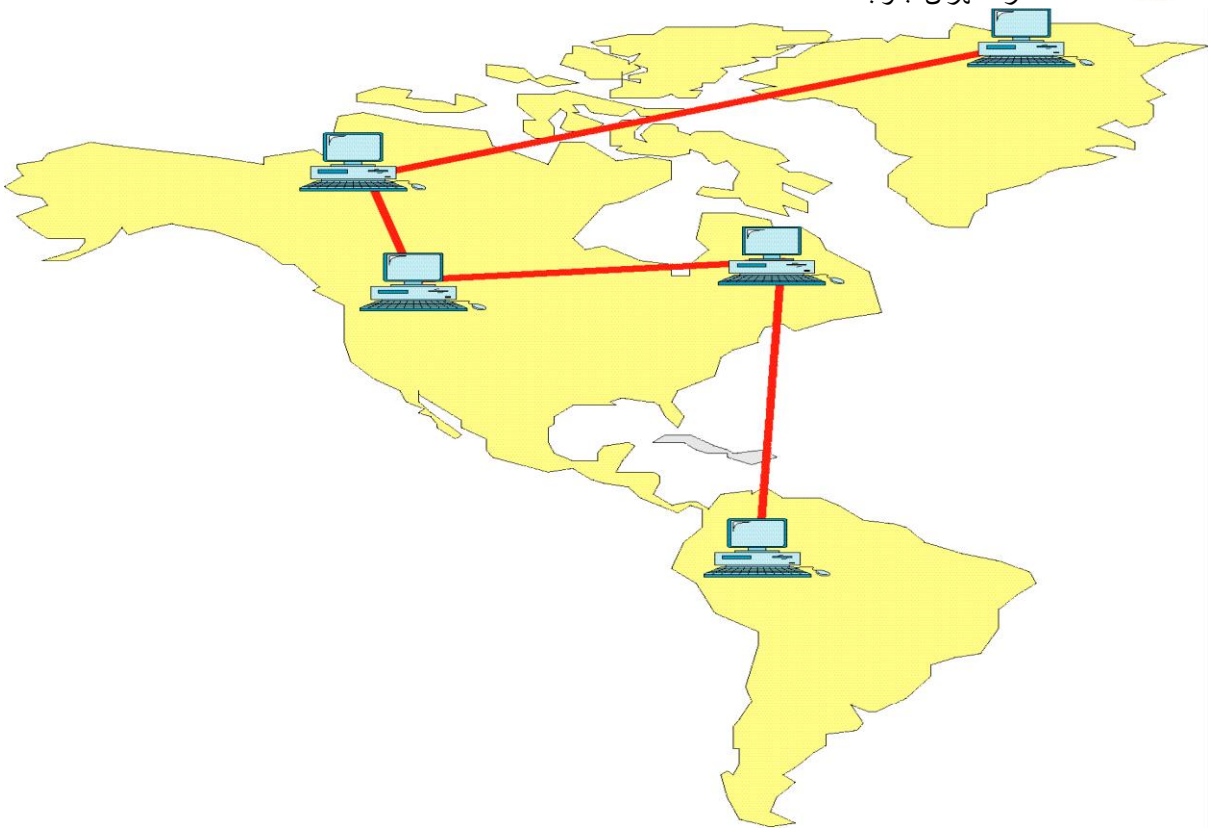
اگر فاصله از حدی بیشتر شود، دیگر شبکه های LAN جوابگو نخواهند بود. در شـبکه هـای MAN ابـری (یک چیز یکپارچه مثل ابر مخابرات) وجود دارد که ارتباط بین شبکه ها را برقرار می کند که این شبکه ها، شبکه های LAN می باشند. این کار هزینه بسیار بالایی خواهد داشت.



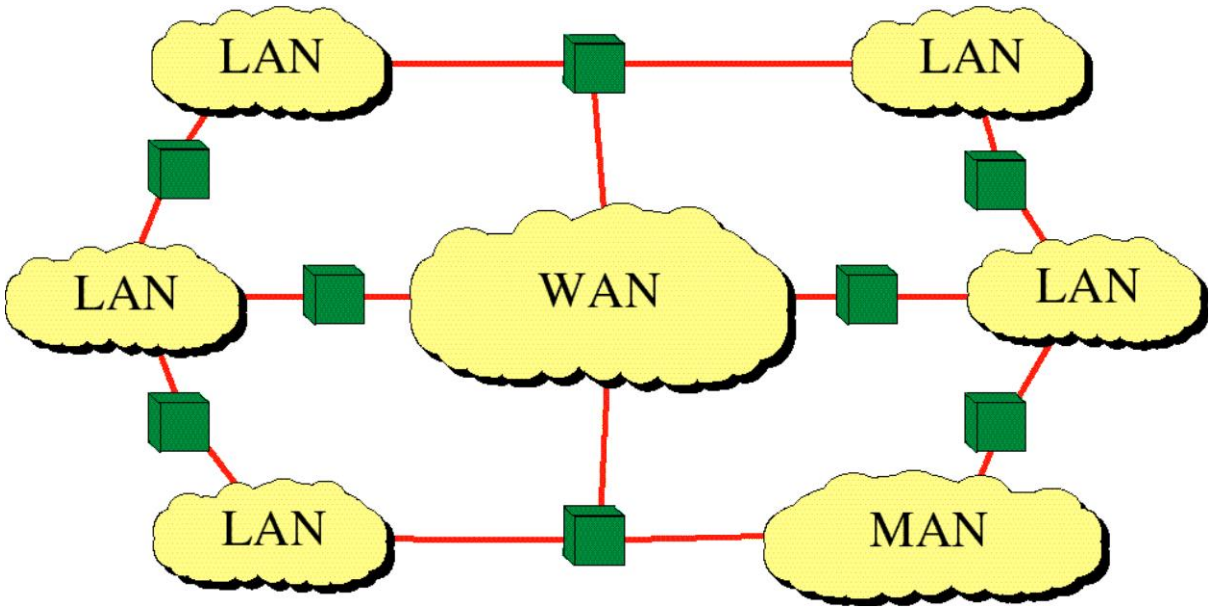
شکل ۱۸: یک نمونه شبکه MAN

شبکه های WAN

در شبکه های WAN، از تکنولوژی Point to Point استفاده می شود که این اطلاعات نقطه به نقطـه ارسال می شوند تا به مقصد برسد. بین شبکه ها، مسیر یابی وجود دارد که وظیفه مسیریابی را بر عهده دارد که اطلاعات از کدام مسیر به مقصد برسد.



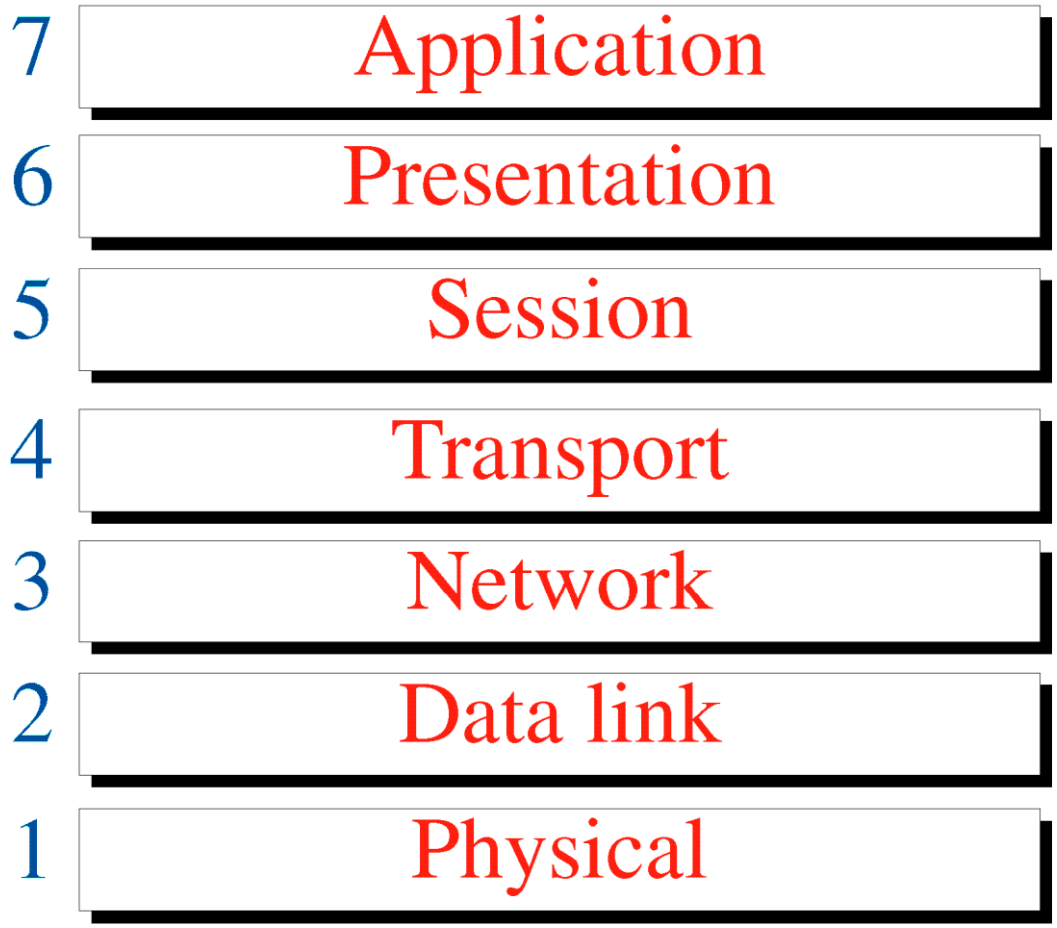
شکل ۱۹: یک نمونه شبکه WAN



شکل ۲۰: آرایش از شبکه

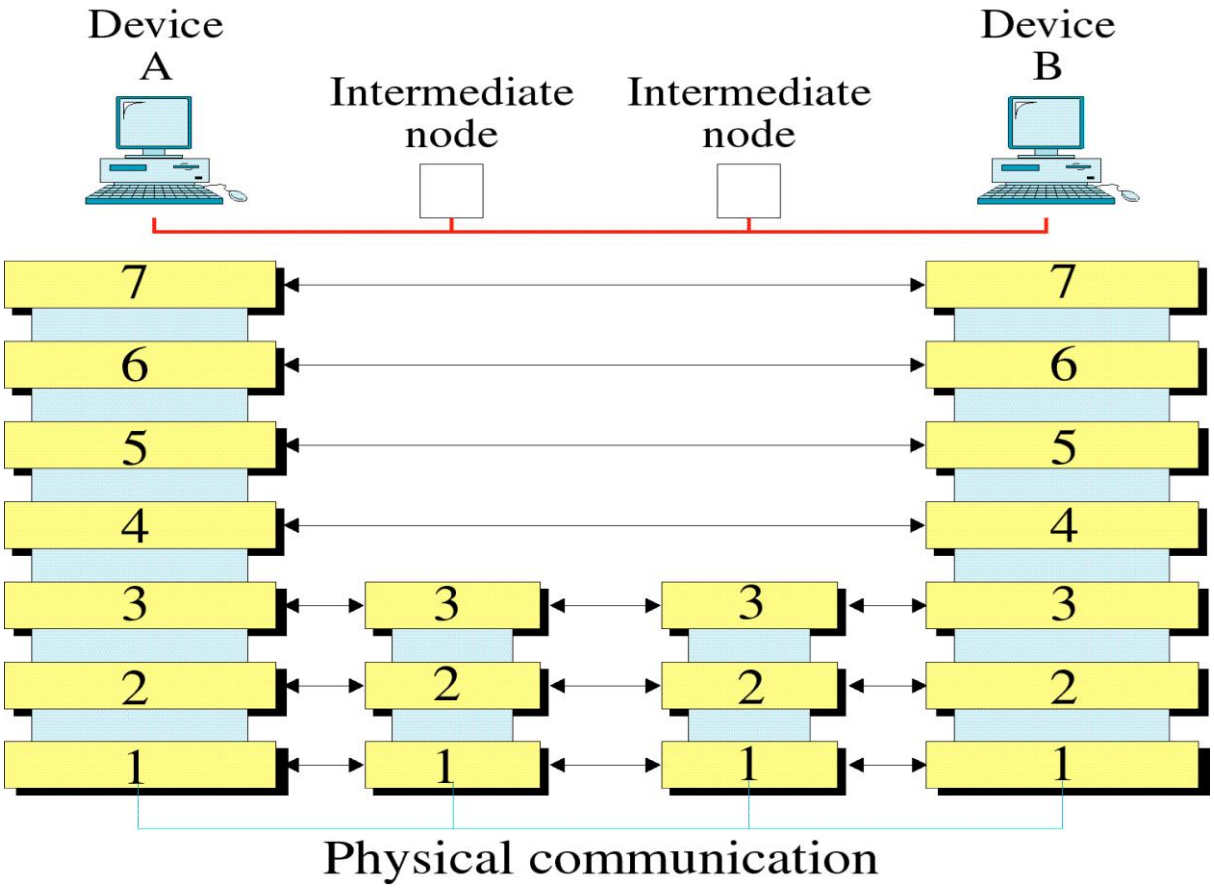
OSI Model

این مدل، مدلی هفت لایه است از وظایف شبکه و هر لایه کاری انجام می دهد و لایه ها کاملا مستقل از یکدیگر می باشند و ارتباط بین لایه ها از طریق Interfaceبرقرار می شود. هر لایه وظیفه دارد که به لایه های بالاتر سرویس بدهد و جزئیات لایه های زیرین را برای لایه های بالایی پنهان نگهدارد.



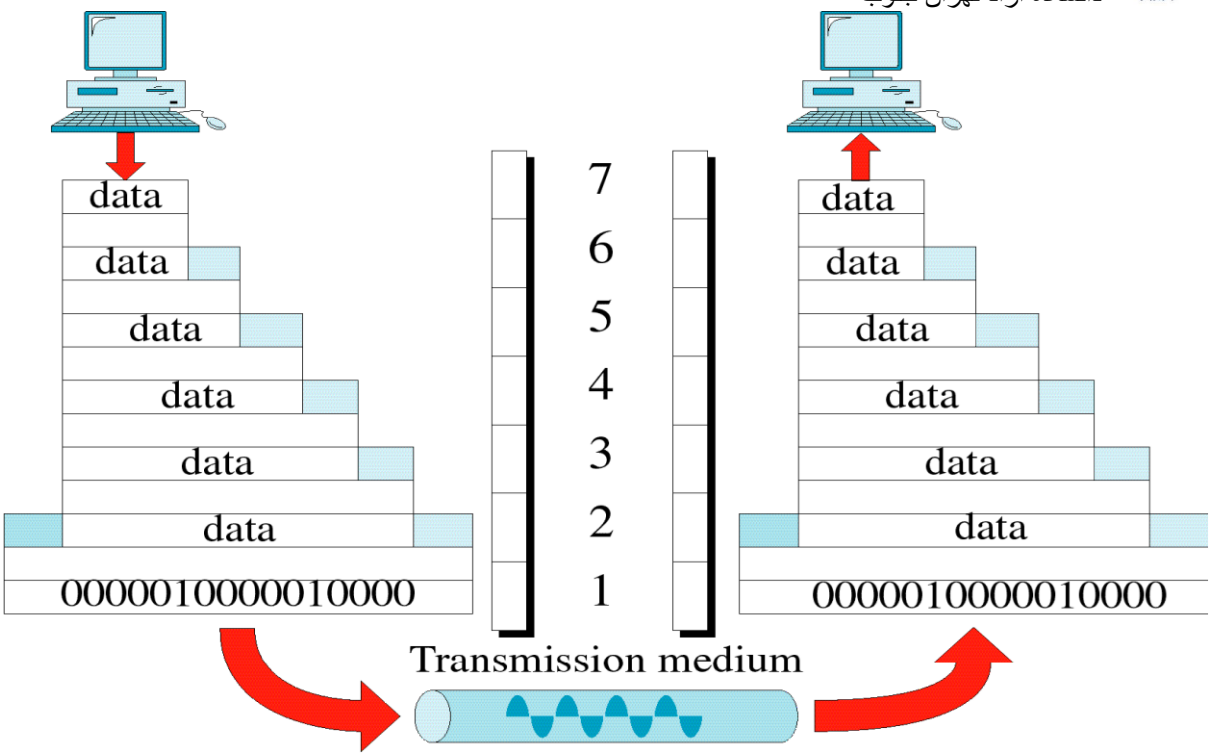
شکل ۲۱: مدل OSI

لایه های متناظر ماشین Aو ماشین B می توانند با هم ارتباط برقرار کنند. پس هر لایه با لایه متناظر یک پروتکل یکسان دارد. هیچ لایه ای نمی تواند مستقیما اطلاعات را روی محیط ارتباطی قرار دهد و برای انتقال اطلاعات، ابتدا لایه ۷ به لایه ۶، لایه ۶ به لایه ۵ و به همین ترتیب انتقال می دهند و اطلاعات را روی محیط ارتباطی قرار می دهند.



شکل ۲۲: ارتباط لایه های منتاظر در دو سیستم

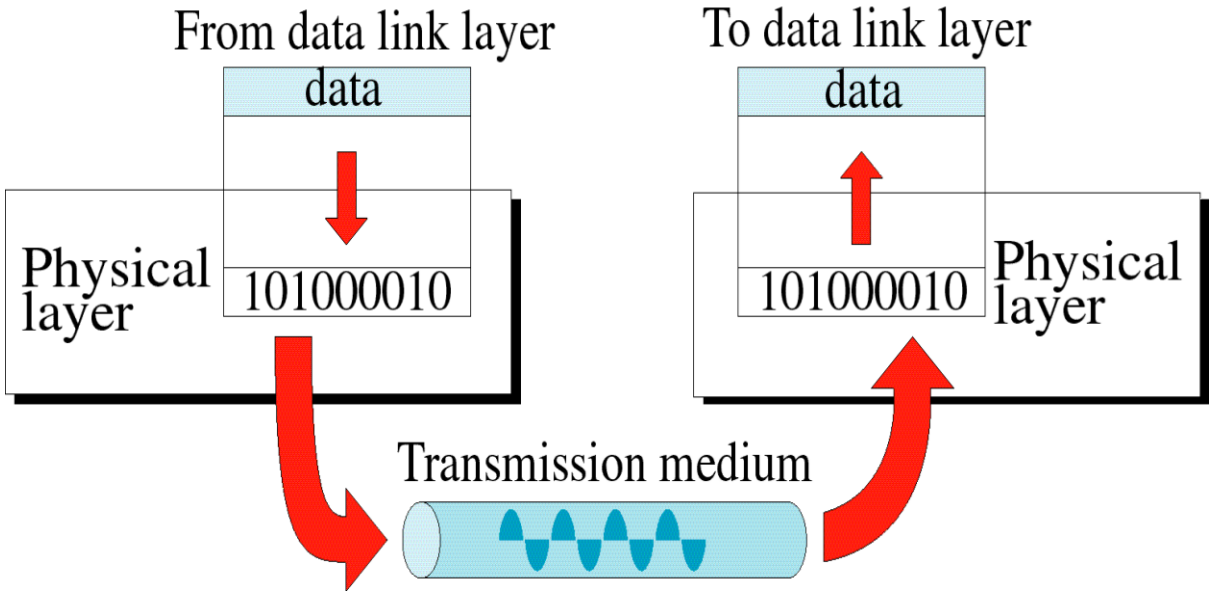
هر لایه برای انجام دادن کار باید یک سری اطلاعات کنترلی داشته باشد تا گیرند بتواند بر حسب آن اطلاعات کنترلی کار را انجام دهد. که هر لایه یک سری اطلاعات کنترلی به داده ها اضافه می کند و به لایه بعدی می فرستد.



شکل ۲۳: محیط OSI

Physical Layer

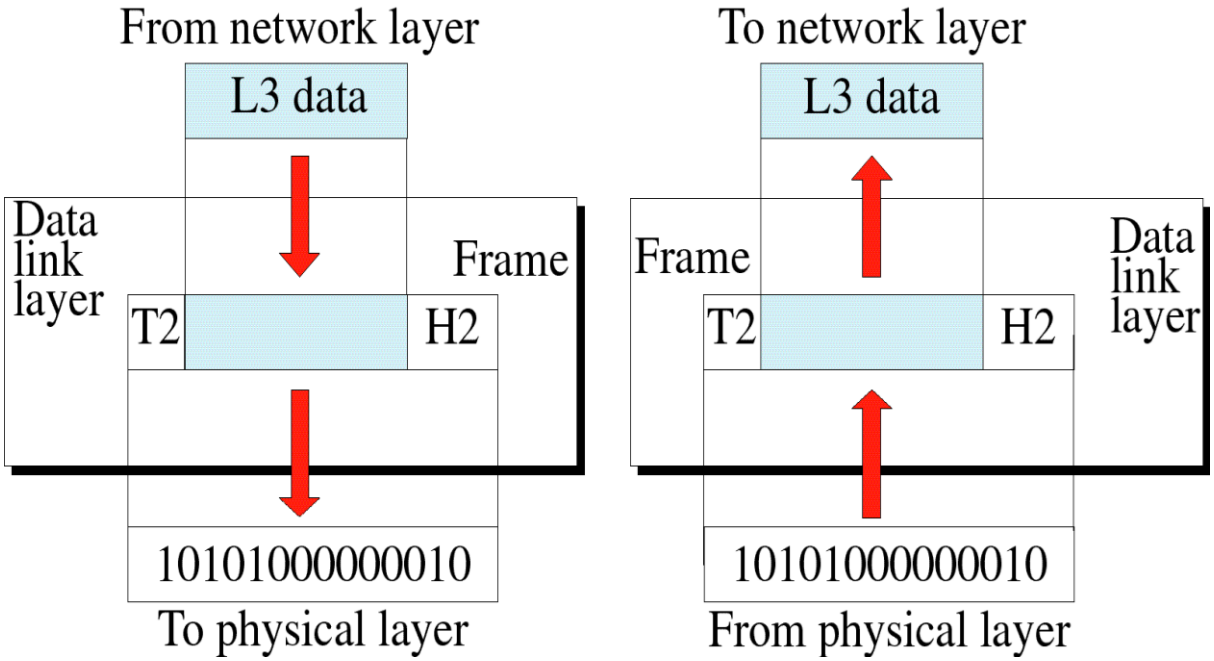
این لایه وظیفه گرفتن اطلاعات از لایه بالاتر (صفر و یک) و تبدیل آن به سیگنال متناسب بـا محـیط را بـر عهده دارد. در لایه فزیکی مستقیما با صفر و یک سروکار داریم. عرض بیت، انواع Coding و سیگنال های آنالوگ و دیجیتال در این لایه مطرح می شود.



شکل ۲۴: لایه فیزیکی

DataLink Layer

این لایه وظیفه کشف خطا را بر عهده دارد. اگر فرستنده اطلاعات را سریع بفرستد و گیرنده نتواند دریافت کند(با همان سرعت)، بخشی از اطلاعات از بین می رود که این لایه وظیفه کشف و تصحیح آن را بر عهده دارد. چون لایه فیزیکی هیچ عکس العملی نسبت به خطا نشان نمی دهد.



شکل ۲۵: لایه DataLink

واحد تبادل اطلاعات فریم است یکی از اطلاعات کنترلی، آدرس کارت شبکه فرستنده است و همچنین آدرس کارت شبکه گیرنده و فریم های مربوط به کشف خطا و فریم Data

آدرس کارت شبکه گیرنده باید به نوعی مشخص گردد که این عمل به چند طریق می تواند صورت پذیرد.

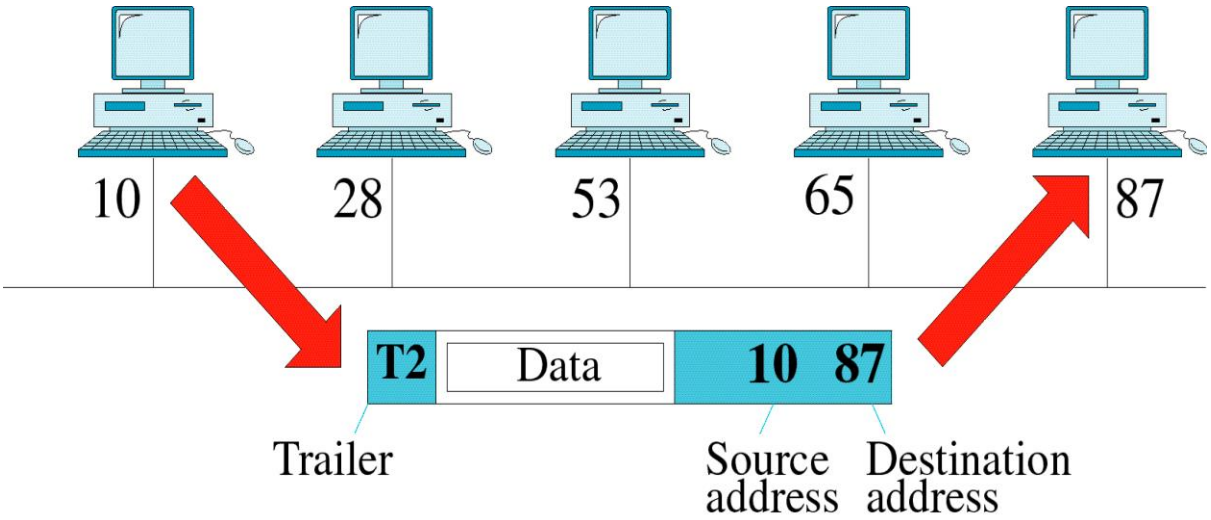
روش اول اینکه هر کسی که وارد می شود آدرس خود را اعلام نماید و روش دوم اینکه خودمان آدرس طرف مقابل را بپرسیم. که لایه فیزیکی این آدرس را دریافت می کند و به لایه های بعدی (DataLink) می دهد. مثلا آدرس مورد نظر ۸۷ است (شکل زیر)

DataLink تمام آدرس ها را چک می کند و چون مورد قبول نیست آن ها را Discard می کند تا به آدرس موردنظر رسیده و این آدرس انتخاب گردد.

در اینجا دو مشکل رخ می دهد. اول اینکه به ازای سوختن هر کارت شبکه، آدرس آن تغییر می کند و دوم اینکه در ارتباطات بین شبکه ها از توپولوژی های مختلفی استفاده می شود. یعنی به دلیل تنوع شبکه های

LAN ، برای وصل کردن آن ها در لایه DataLink به مشکل بر می خوریم به همین دلیل، لایه Networkمطرح می شود که با طرح کردن آدرس لاجیک در این لایه، مشکل آدرس ها حل خواهد شد. در لایه Network ، شبکه های LAN با آدرس های متفاوت قالب یک آدرس لاجیکی وجود دارند.

مثالی از لایه DataLink

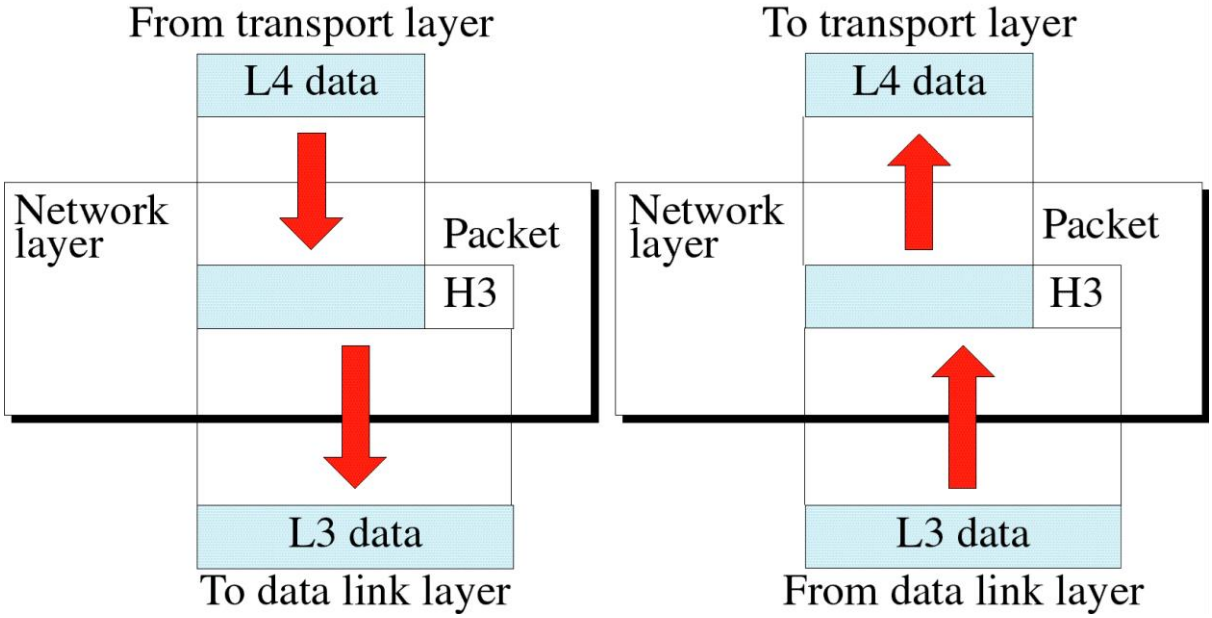


شکل ۲۶: مثال لایه DataLink

این لایه برای کشف خطا یک سری اطلاعات کنترلی اضافه می کند. شبکه های مختلف، تفاوتشان در لایه های فیزیکی و DataLink است که این دو لایه از لایه های سخت افزاری می باشند و از لایه ۳ به بالا اکثرا به صورت نرم افزاری پیاده سازی می شوند. وقتی یک Packetوارد می شود، لایه فیزیکی بدون در نظر گرفتن آدرس آن بسته را عبور می دهد این لایه DataLink است که وظیفه پیدا کردن آدرس صحیح را بر عهده دارد. اگر آدرس موجود در Packet مربوط به کامپیوتر نباشد، لایه DataLink ، Reject می کند.در Packet ای که فرستاده می شود آدرس لاجیکی فرستنده و گیرنده ثبت می شود از دید لایه DataLink آدرس های لاجیکی هیچ مفهومی ندارند و برای شناسایی آن را به لایه Network می دهد و لایه Network آدرس لاجیکی را به آدرس کارت شبکه تبدیل کرده به لایه DataLink می دهد تا آدرس را تشخیص دهد.

لایه Network

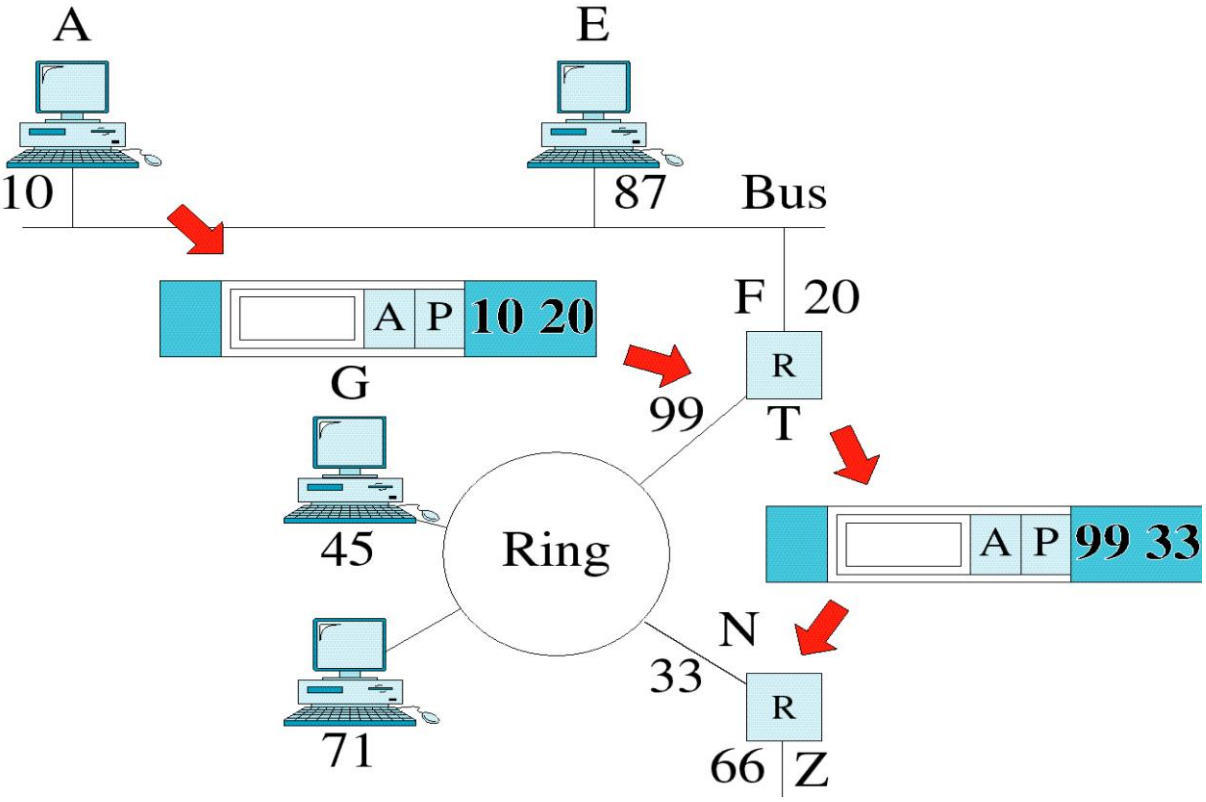
ارتباطات بین شبکه ای نیز در این لایه مطرح می شود، همچنین مسیریاب ها و آدرس های لاجیکی به این لایه مربوط می شود.



شکل ۲۷: لایه Network

مثالی از لایه Network

در این مثال فریم هایی با آدرس لاجیکی و آدرس کارت شبکه گیرنده دیده می شود. بعنوان مثال A و P آدرس های لاجیکی و ۱۰ و ۲۰ آدرس های اصلی می باشد که پس از رسیدن به مسیریاب ها آدرس ها چک شده و در صورت لزوم تغییر می کنند و آدرس مسیریاب های بعدی در فریم ها قرار می گیرد تا جایی که به آدرس های منطقی داده شده، دست پیدا کنیم.



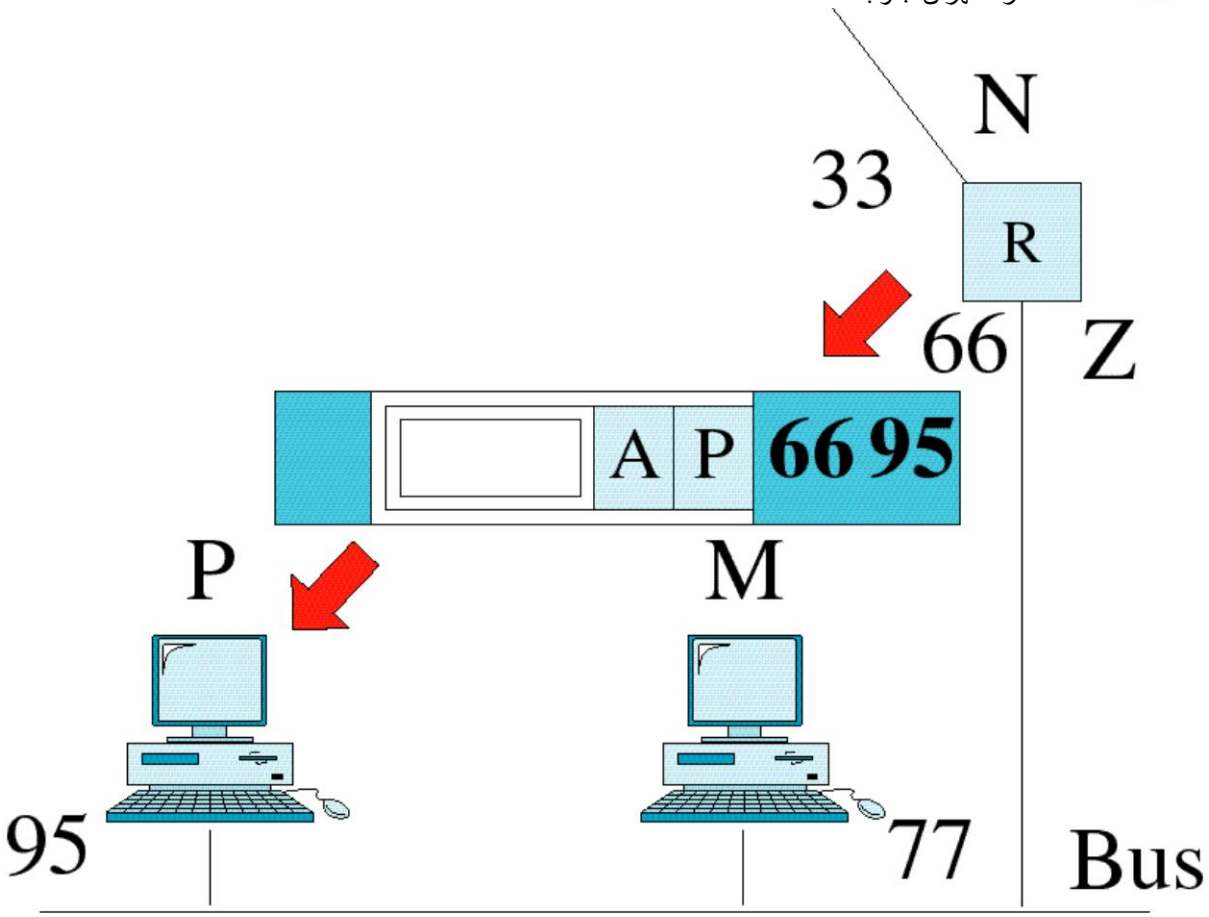
شکل ۲۸: مثال لایه Network

شکل ۲۹: تغییر آدرس لاجیکی در مسیریاب ها

در شبکه ها، مسیریاب ها هستند که وظیفه مسیریابی و پیدا کردن مقصد را بر عهده دارند.

لایه Transport

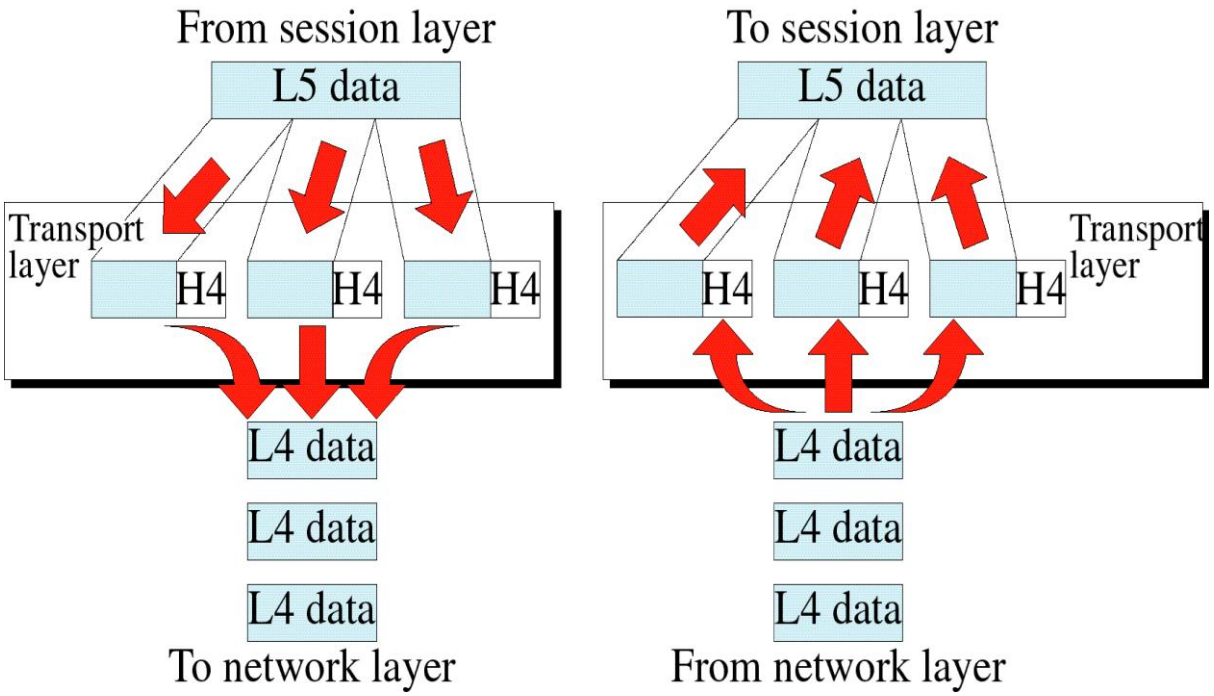
امکان خطا در لایه Network وجود دارد. هیچ لایه ای، لایه بالاتر را کنترل نمی کند پس لایه Network لایه امنی نیست و به همین دلیل، لایه Transport مطرح می شود تا به لایه Network ، امنیت بیشتری دهد.



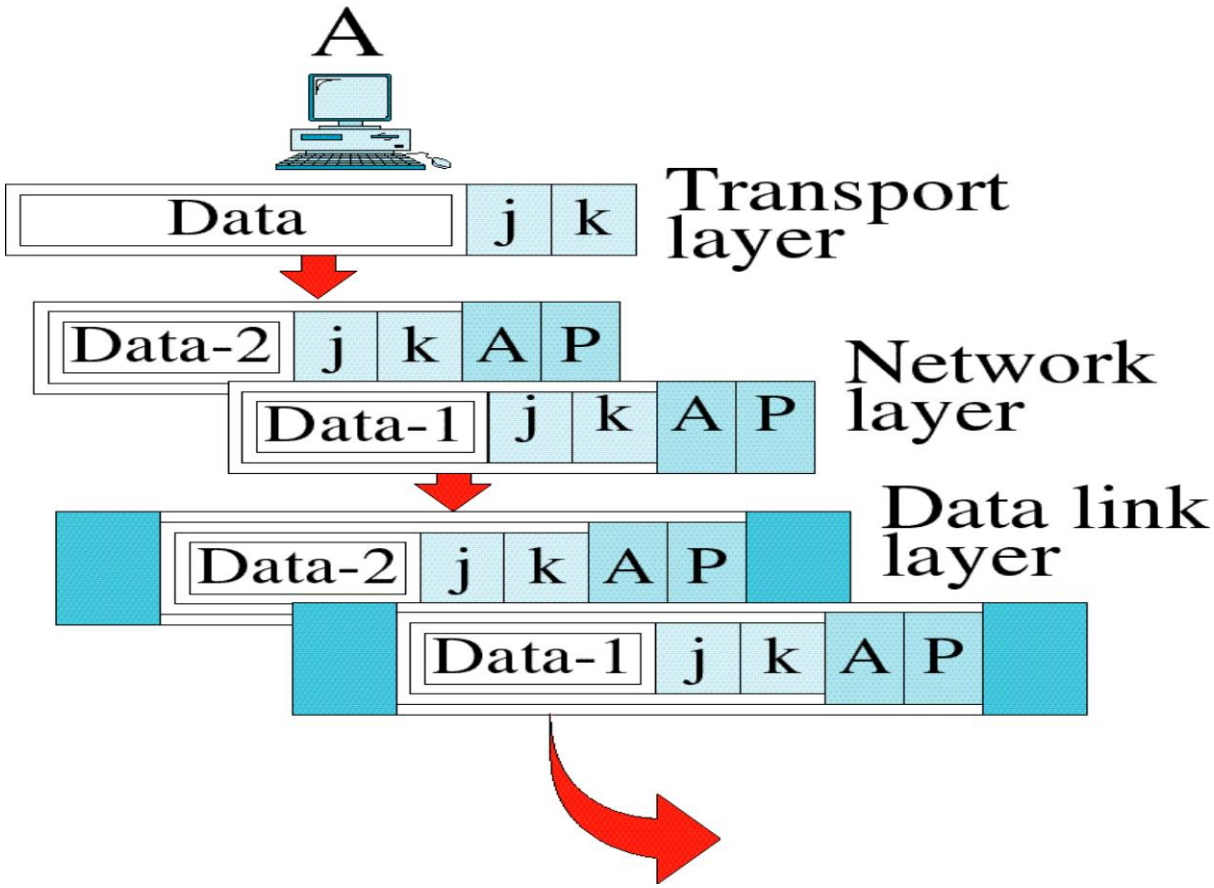
شکل ۳۰: لایه Transport

به واحد تبادل اطلاعات در لایه Transport ، Data Unit گفته می شود که اطلاعات را میگیرد. حال اگر این اطلاعات خیلی زیاد باشد، به دلیل اینکه نمی تواند اندازه Packet را بزرگ کرده و رشد دهد، مجبور است که اطلاعات را شکسته و بسته بندی کند.

از لایه Transport به بالا در قالب یک لایه است که به آن Application گفته می شود. لایه Transport احتیاج به آدرس Applicationدارد و اگر لایه Transportبه راحتی شخیص دهد که این Data Unit مربوط به کدام Applicationاست، مشکل حل می شود. در این دو شکل، چگونگی دادن اطلاعات از لایه Transport به لایه زیرین و چگونگی گرفتن اطلاعات لایه Transportاز لایه های زیرین به نمایش در آمده است و همانطور که دیده می شود، هر لایه یک سری اطلاعات کنترلی مربوط به خود را اضافه نموده و به لایه پایینی می دهد و در برگشت آن اطلاعات کنترلی را برداشته و به لایه بالایی می دهد.



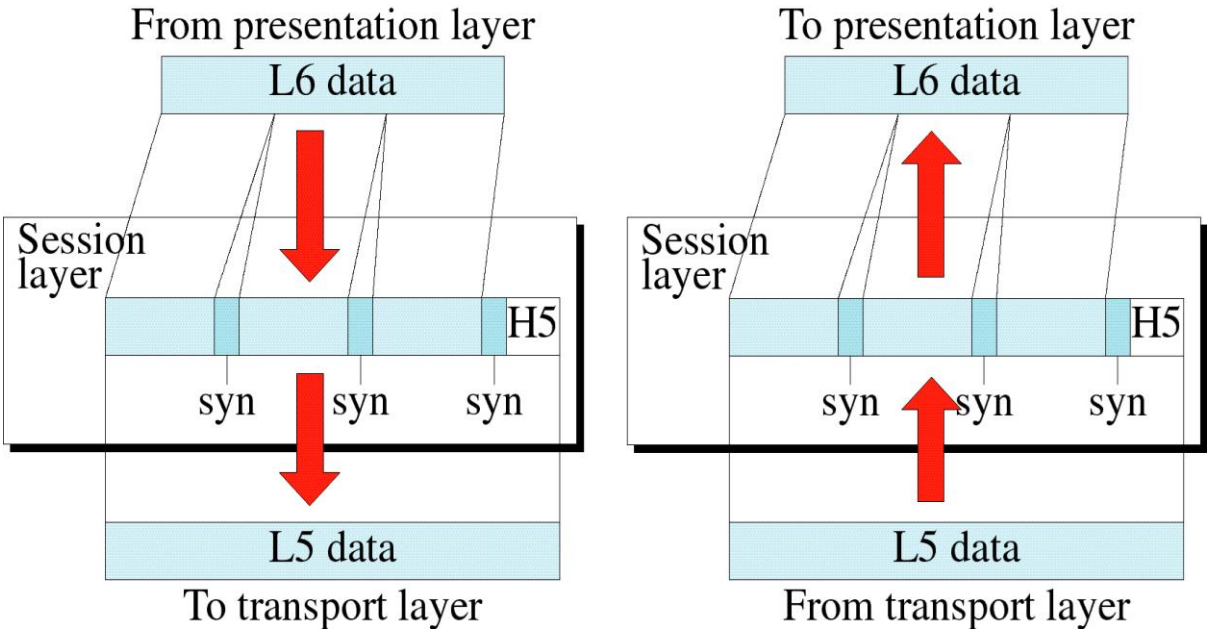
شکل ۳۱: بسته بندی داده در فرستنده از لایه Transport تا DataLink



شکل۳۲: باز کردن داده در گیرنده از لایه DataLink تا Transport

لایه Session

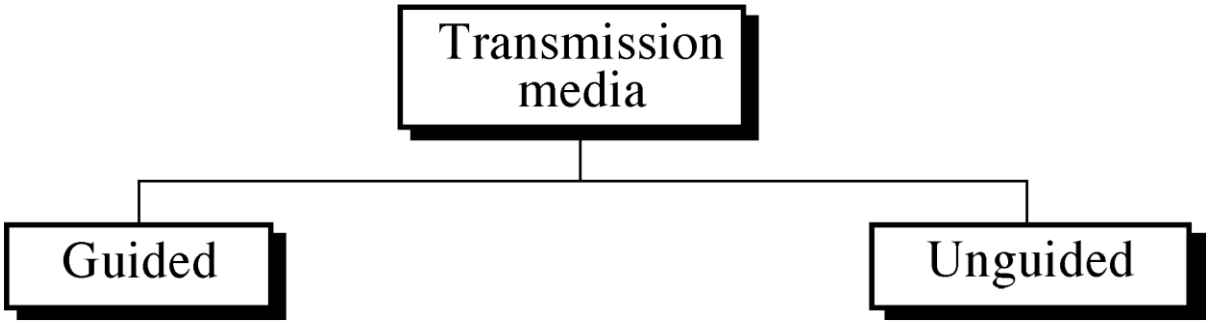
این لایه که به آن جلسه گفته می شود، وظیفه مدیریت تبادل اطلاعات را بر عهده دارد.



شکل ۳۳: لایه Session

صفحه ۲۹

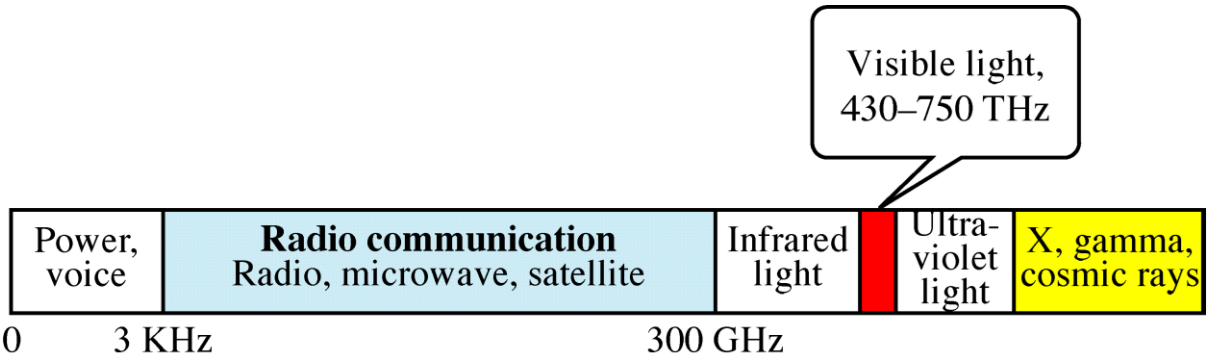
انواع محیط های ارتباطی(Transmission Media)



۱ .هادی(Guided Media)

۲ .غیر هادی(Unguided Media)

* یک طیف فرکانس خاص متناسب با محیط های ارتباطی مختلف وجود دارد .به عنوان مثال پهنـای بانـد صدای شما روی تلفن، ۴ کیلوهرتز است و صدای استریو، امواج رادیویی، ماکرویو، ۲۰ کیلوهرتز به بالا می باشد.

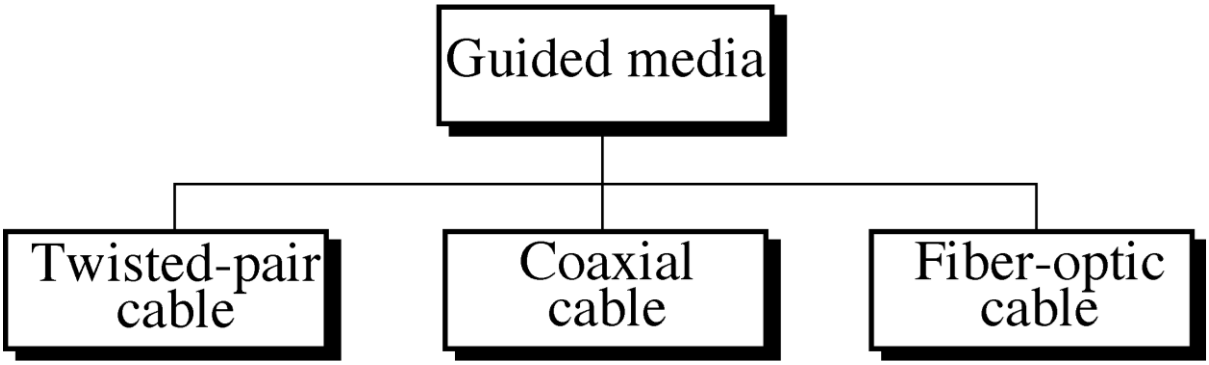


*پهنای باند :حداقل و حداکثر فرکانس که محیط عبور می دهد.

به طور کلی امواج ماهواره ها بالای یک گیگاهرتز کار می کننـد .وسـایلی مثـل Remote Controls & Laptopامواجی از ۳۰۰GHz به بالا دارند .پس از آن رده امواج ماوراء بفش هستند.

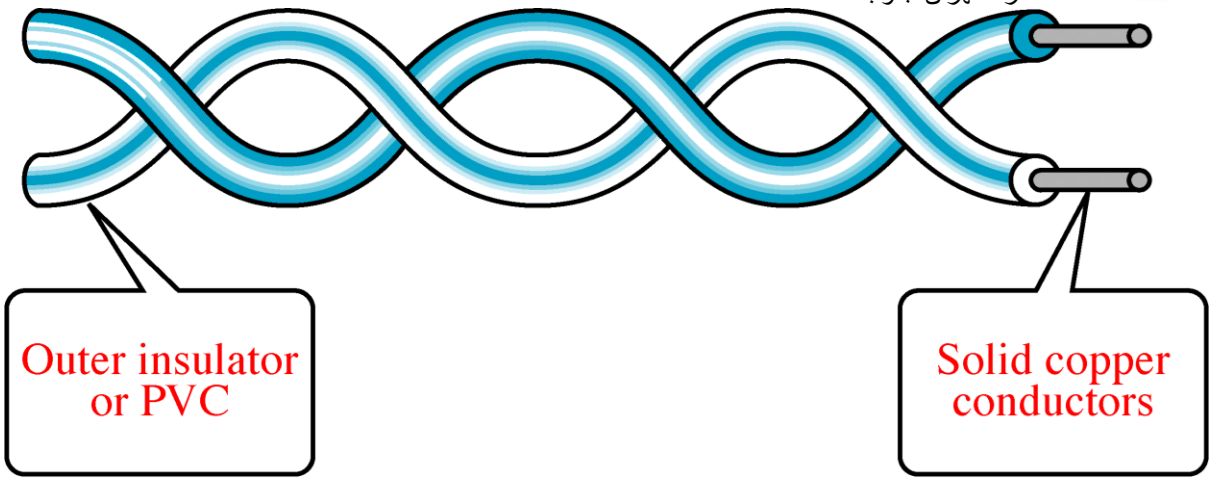
Visible Liqhtبه امواج مرئی، مثل نوری که دیده می شود گفته می شود و پس از آنها امـواج ایکـس، گاما و ... که هر یک کاربرد و ویژگی های خود را دارند، می باشند.

محیط های رایج انتقال اطلاعات(Guided Media)

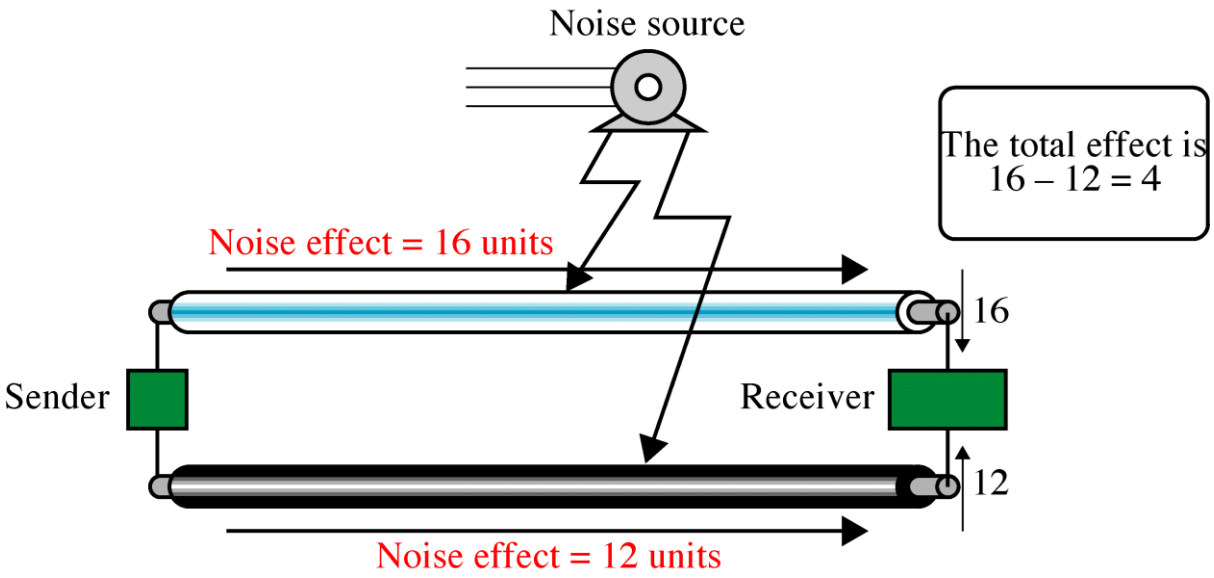


کابل هایTwisted-Pair

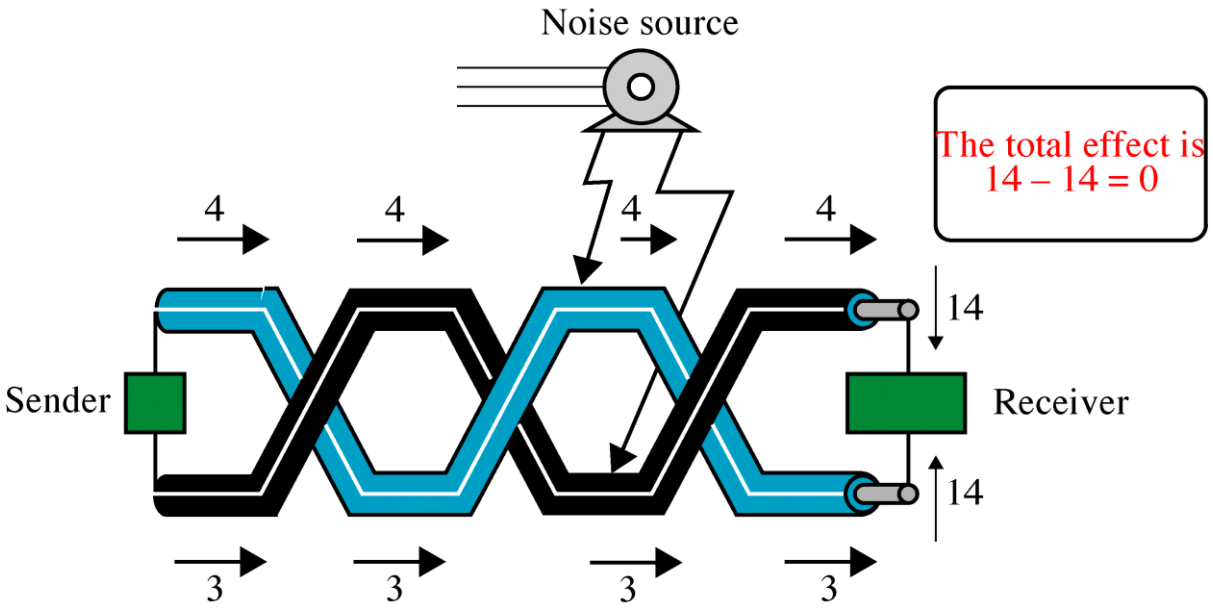
کابل های Twisted-Pair ، کابل های رایجی هستند که در منازل از آنها استفاده می شود .این کابل هـا داری یک مفتول فلزی و یک پوشش هستند که به یکدیگر تاب خورده اند و این تاب خـوردگی سـبب مـی شود که اثر نویز ها از بین برود .فرکانس آن از ۱۰۰Hz تا ۵MHz می باشد.



منبع نویز روی کابل نزدیک تر اثر بیشتری دارد، هنگام ارسال اطلاعات برای امنیت بیشتر، هم خود اطلاعات ارسال می شود هم not اطلاعات.



در شکل دیده می شود که اثر نویز روی سیمی که به منبع نویز نزدیک تر است، ۱۶ واحد می باشد در صورتی که روی سیمی که رورتر است ۱۲ واحد می باشد وقتی که این دو مقدار را از هم کم می کنیم، یاز هم ۴ واحد نویز خواهیم داشت.



حال اگر کابل ها، مطابق این شکل به هم تابیده شوند، اث نویز در آن ها از بین خواهد رفت .چون همانطور که دیده می شود اثر نویز روی کابل اول و دوم، هر دو ۱۴ واحد است که اگر ایـن ذو را از هـم کـم کنـیم وقدار نویز به صفر می رسد.

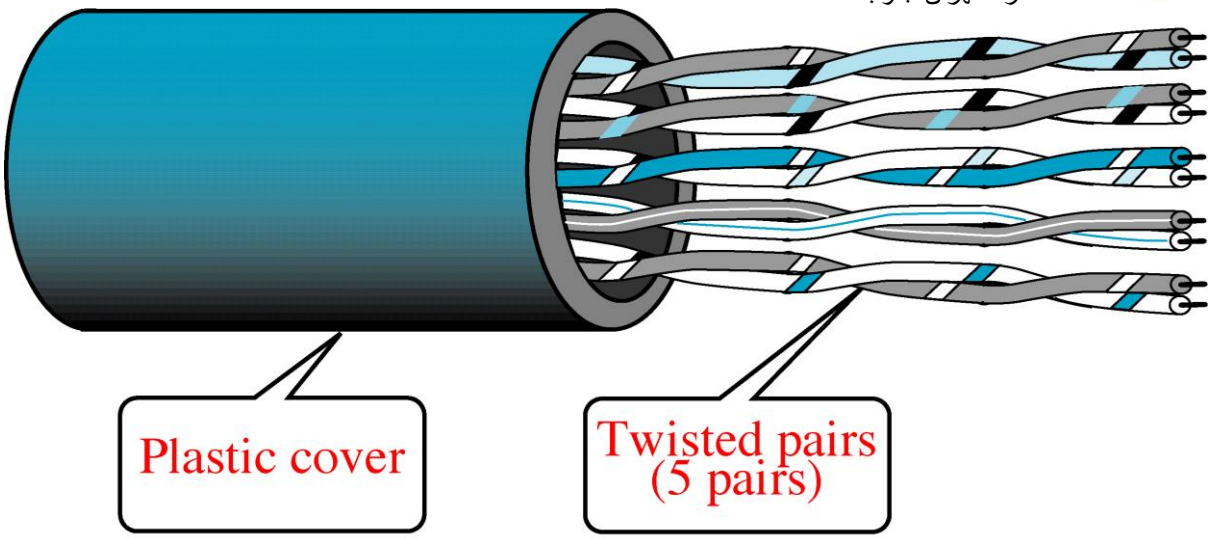
بنابراین، هر چه پیچش در کابل ها بیشتر باشد، اثر نویز کمتر می شود.

تقسیم بندی کابل هایTwisted-Pair

۱.Shielded Twisted-Pair

۲.Un Shielded Twisted-Pair

در نوع اول، یک پوشش روی کابل ها وجود دارد که اثر نویز را کم می کند و در کار های صنعتی از این نوع کابل استفاده می گردد.



در نوع دوم، باز هم دو نوع کابل وجود دارد Cat۳ :و .Cat۵ که این دو از نظر جنسی هیچ فرقی با هم ندارند و تنها فرق آن ها از نظر پیچش کابل است.

در کابل Cat۳ ، نرخ نرخ اطلاعات ۱۰ مگابیت است .در حالی که در کابل Cat۵ نرخ اطلاعات ۱۰۰ مگابیت می باشد.

این دو نوع کابل، ۴ زوج سیم تاب خورده اند که درون یک پوشش پلاستیکی قرار گرفته اند و عموماً در شبکه های عادی از دو زوج آن بیشتر استفاده نمی شود.

کاربردهای گابلTwisted-Pair

۱ .سیم کشی عادی

۲ .شبکه تلفن

۳ .بین ساختمان ها

۴ .شبکه هایLAN

کابل Coaxial

این کابل، از یک مفتول فلزی که روی آن را یک عایق پلاستیکی و سپس یک بفافت فلـزی و دوبـاره یـک عایق پلاستیکی پوشانده است، تشکیل شده و فرکانس آن بین ۱۰۰ تا ۵۰۰ مگاهرتز می باشد .پهنای بانـد این کابل، ۱۰۰ برابر کابل Twisted-Pair است.

انواع کابل Coaxial

۱ .کابل های ۵۰ اهمی :انتقال اطلاعات در آن های به صورت Digital است.

۲ .کابل های ۷۵ اهمی :انتقال اطلاعات در آن های به صورت Analog است.

کاربرد کابل های Coaxial

۱ .تلویزیون(Television Distribution)

به دلیل پهنای باند بالا.

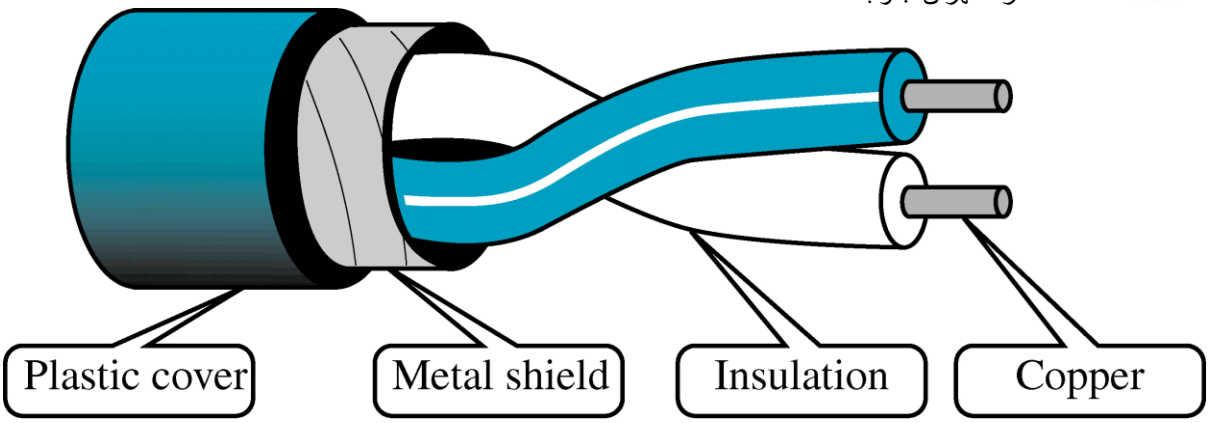
۲ .بین مراکز تلفن(Long distance Telephone Transmission)

۳ .شبکه های:LAN (Local Area Network)

با سرعت ۱۰ مگابیت که از کابل ۵۰ اهمی دیجیتال است و محدودیت کابل و پهنای باند ندارد.

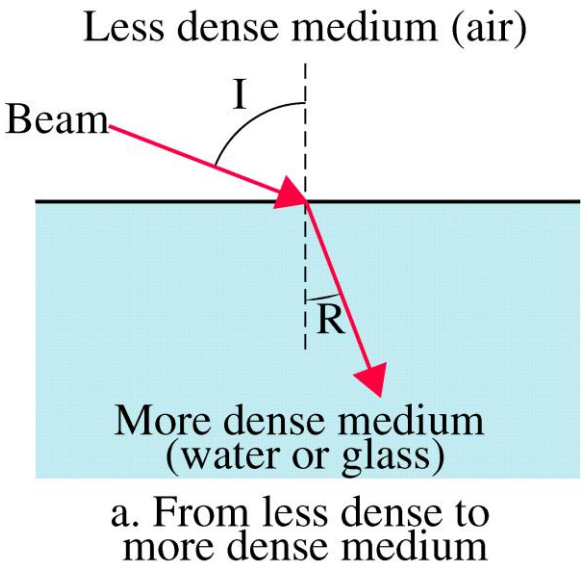
کابل های فیبر نوری(Fiber Optic Cable)

در فیبر نوری هسته ای از جنش شیشه است که با یک پوشش شیشه ای پوشانده شـده اسـت .(Buffer) که توان شکست آن کمتر از هسته است تا کـل نـور را در هسـته نگهـداری کننـد .سـپس پوشـش نـازک پلاستیکی قرار دارد تا از پوشش شیشه ای محافظت کنند .فیبر ها در بسته هایی با یکدیگر دسته بندی می شوند و توسط پوشش خارجی محافظت می شوند.

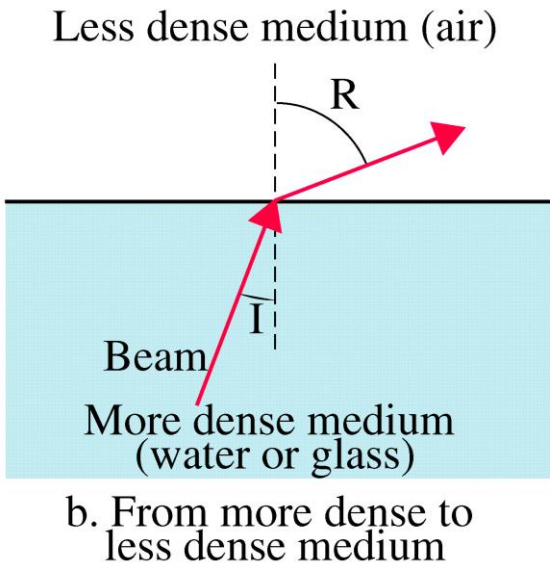


در فیبر نوری از نور برای انتقال اطلاعات استفاده می گردد که در اینجا پدیده ای به نـام Refraction یـا شکست به وجود می آید .زمانیکه نور از هوا وارد محیط آبی یا شیشه ای می شود، شکست پیدا می کنـد . پس برای زوایای بیش از یک مقدار بحرانی معین، نور درون سیلیکا بازگشت می کند .هیچ کدام آن ها بـه درون هوا نمی رود .لذا هر پرتو با زاویه ی بحرانی یا بالاتر از آن به مرز برخورد می کند، از محیط سـیلیکا خارج نمی شود و بدون آن که از بین برود، می تواند کیلومتر ها انتشار یابد.

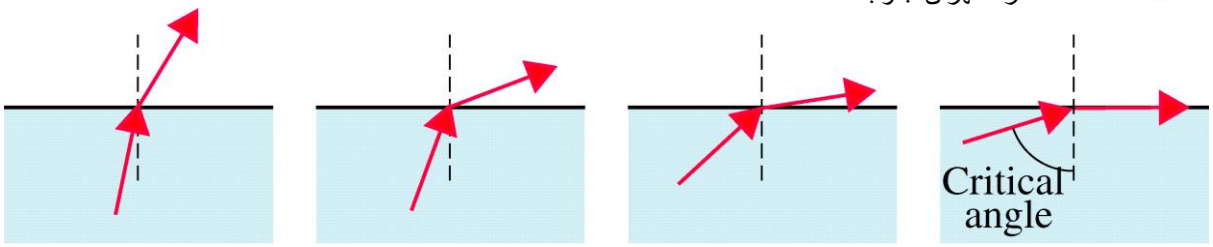
پس زاویه ی شکست نور باید طوری تنظیم شود که از محیط خارج نشده و در مسیر خود ادامه یابد.

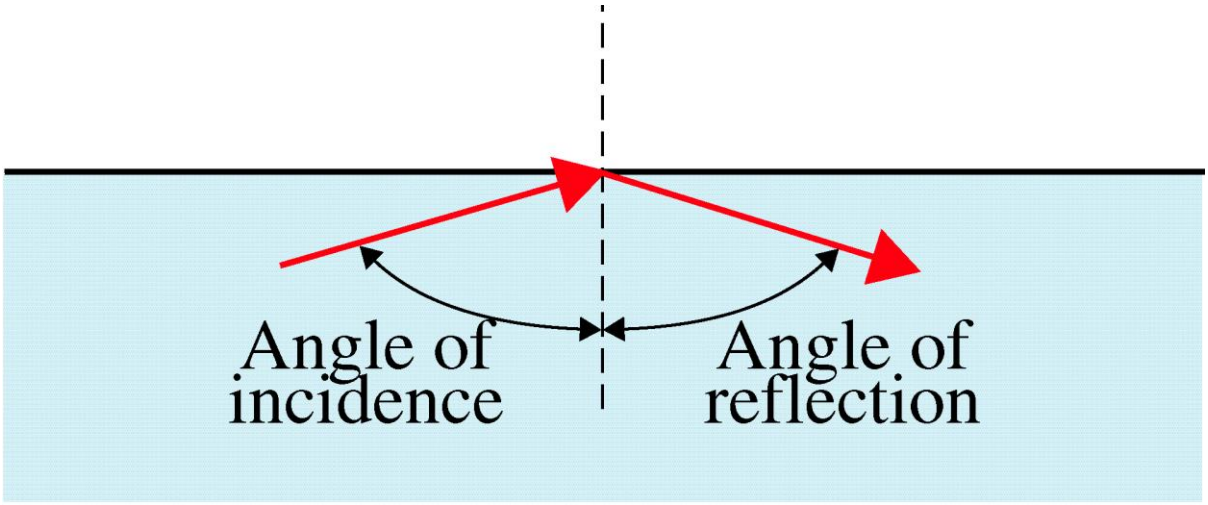


a. From less dense to more dense medium



b. From more dense to less dense medium





ویژگی های فیبر نوری

۱ .پهنای باند بالا یا ظرفیت بیشتر(صد ها گیگابیت در ثانیه)

۲.سایز و وزن کم.

۳ .تضعیف کم.

۴ .ایزوله بودن در برابر امواج الکترومغناطیسی(.از نور استفاده می شود و نور به دلیل اینکه باری ندارد، میدانی بوجود نمی آورد)

برای افزایش برد انتقال اطلاعات در صورت تضعیف، از تکرار کننده یا تقویت کننده استفاده می شود.

مشکلات فیبر نوری

۱ .تجهیزات آن گران است.

۲ .کابل کشی آن یک کار تخصصی است) اگر زاویه خمش تغییر کند، به مشکل بر می خوریم.(

کابرد های فیبر نوری

۱) Telephone Network .از محیطی با پهنای باند بیشتر استفاده می شود.

۲) Metropolitan Trunks .تلفن های شهری

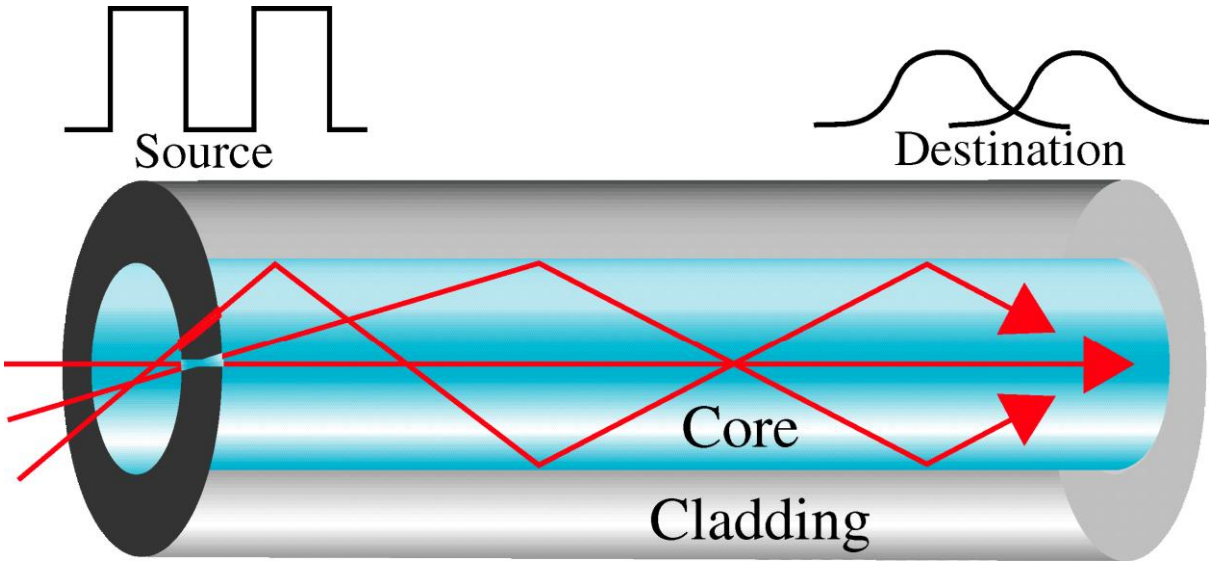
۳ .شبکه هایLAN

انواع فیبر های نوری

دو نوع رایج فیبر نوری عبارتند از:

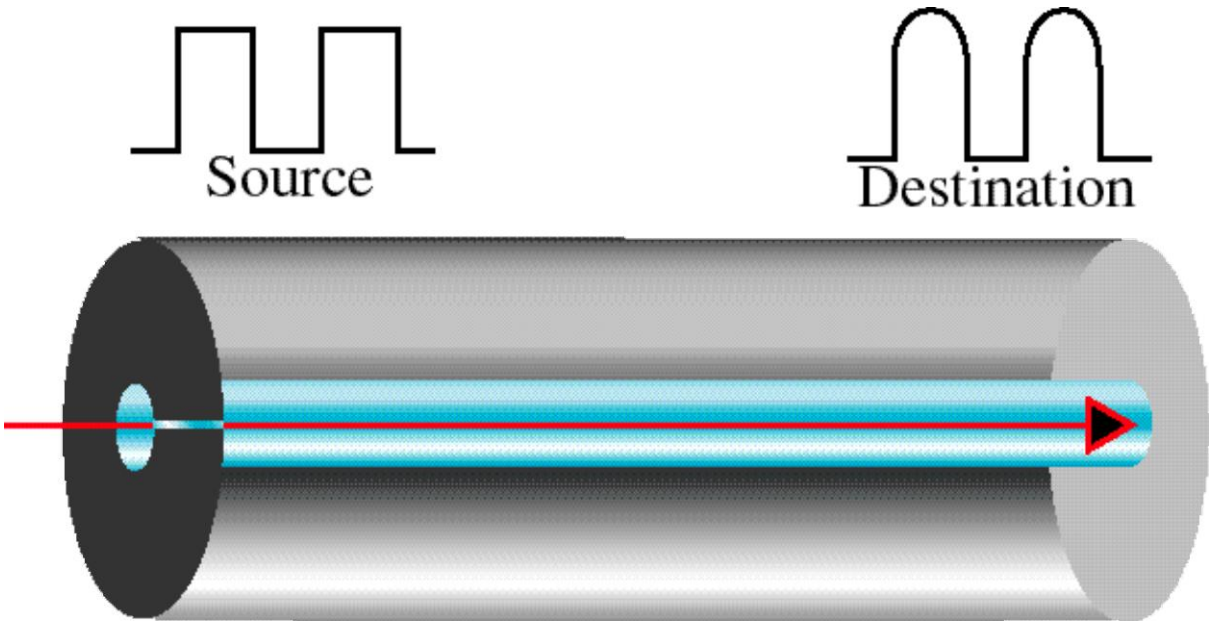
۱. Multi Mode

۲. Single Mode



در Multi Mode قطر Core زیاد است و چون هر پرتوی نوری که با زاویه ای بیش از مقدار بحرانی به مرز برخورد کند به محیط اول برمی گردد، بسیاری از پرتو ها با زاویه های متفاوت نوسان می کنند و هر

پرتو حالت متفاوتی دارد .در این حالت قطر Core نسبت به Cladding زیاد است و نرخ انتقال اطلاعـات در آن ها کاهش می یابد.



در حالت Single Mode ، قطر Core فقط در حد تابیدن یک پرتو است و نور بدون نوسان کردن در یک خط مستقیم انتشار می یابد .در این حالت تقدم و تأخر در فرستادن سیگنال ها نداریمو ورودی بـه همـان شکل در خروجی دریافت می شود.

قیمت فیبر های نوری Single Mode خیلی بیشتر از Multi Mode است .در شبکه های LAN همه فیبر ها از نوع Multi Mode هستند.

لایه Physical

کد گذاری اطلاعات(Encoding)

در لایه Pysical ورودی به صورت صفر و یک است و خروجی سیگنال متناسب با محیط می باشد.

تکنیک های مختلف کد گذاری

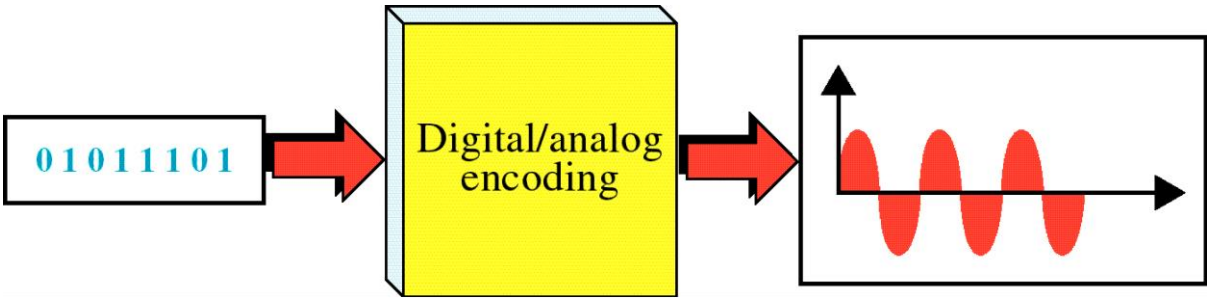
۱ .ورودی داده دیجیتال و خروجی سیگنال آنالوگ باشد.(مانند مودم)

۲ .ورودی داده دیجیتال و خروجی سیگنال دیجیتال باشد .(مانند تلویزیون)

۳ .ورودی داده آنالوگ و خروجی سیگنال آنالوگ باشد.(مانند تلفن)

Digital to Analog Encoding

در این حالت همانطور که گفته شد ورودی صفر و یک و خروجی سیگنال متناسب با محیط می باشد.



۳ پارامتر کلیدی برای کد گذاری اطلاعات در محیط آنالوگ عبارتند از:

۱ .دامنه A .

۲ .تعداد تکرار ها در واحد زمان(فرکانس).

۳.اختلاف نسبی نسبت به زمان P (فاز)

از این ویژگی ها و یا ترکیبی از آن ها می توان برای ساختن سمبل های آنالوگ استفاده نمود.

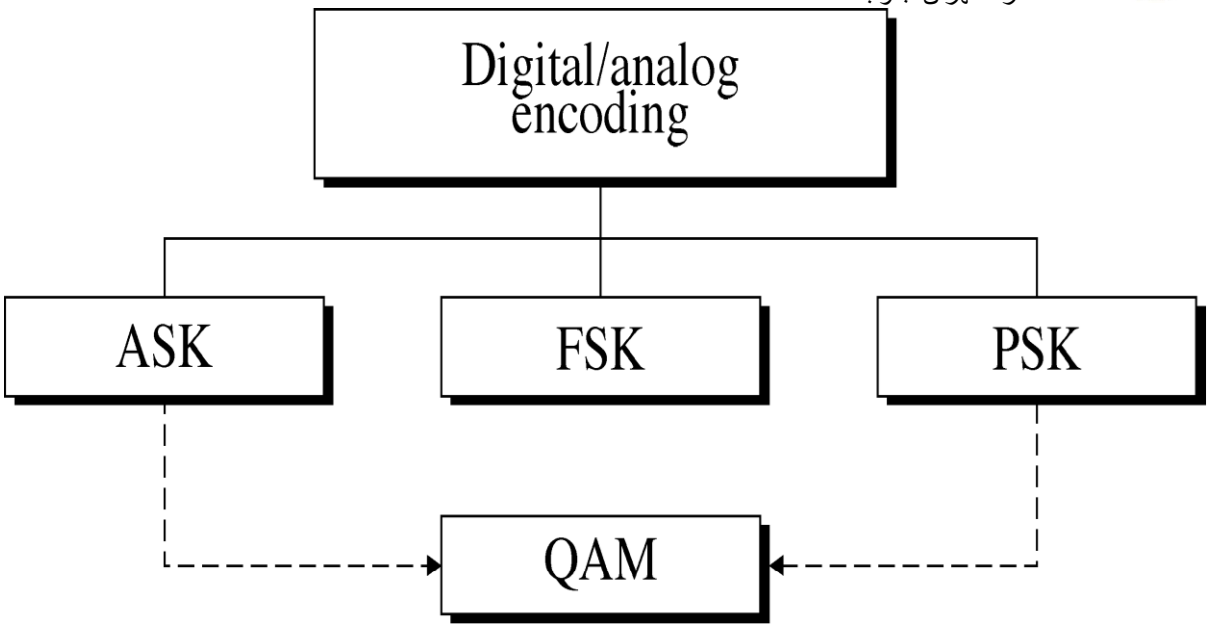
روش های تبدیل سمبل های صفر و یک به سمبل های آنالوگ:

۱ : ASK .از ویژگی دامنه برای ساختن سمبل های آنالوگ استفاده می کند.

۲ : FSK .از ویژگی فرکانس برای ساختن سمبل های آنالوگ استفاده می کند.

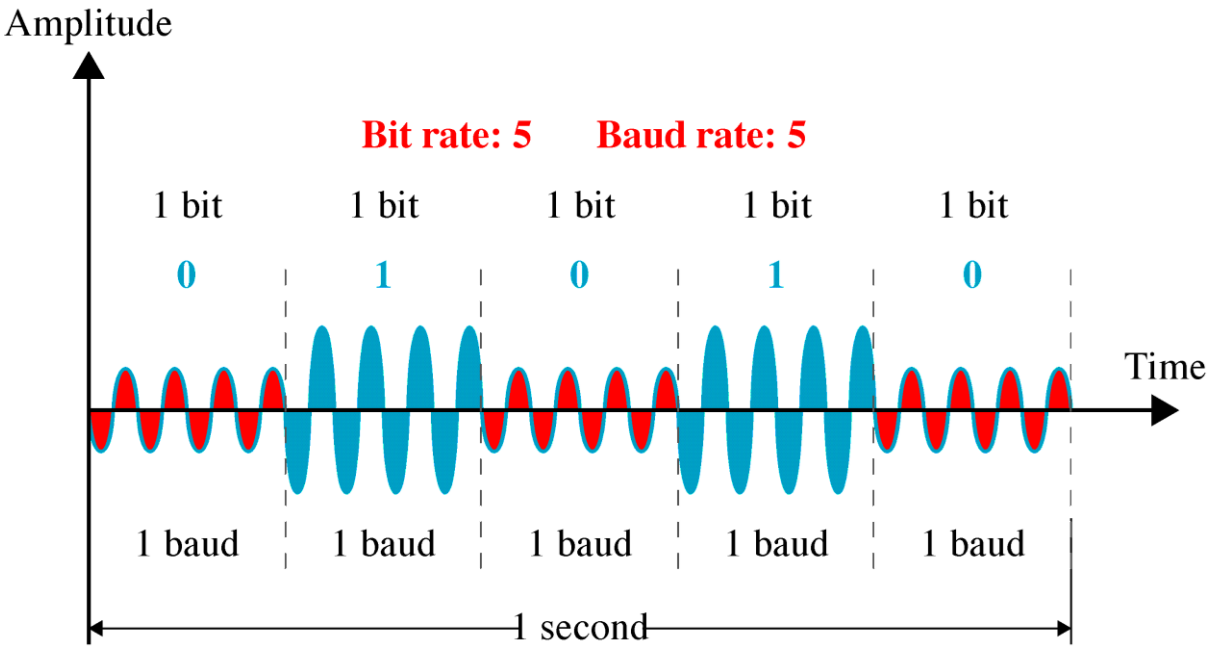
۳ : PSK .از ویژگی فاز برای ساختن سمبل های آنالوگ استفاده می کند.

۴ : QAM .از ترکیبی از ویژگی دامنه و فاز برای ساختن سمبل های آنالوگ استفاده می کند.



ASK

در این حالت، سمبل ها را با ویژگی های دامنه مشخص می کنیم و فرکانس و فاز ثابت مـی باشـد .یعنـی دامنه را تغییر می دهیم .همانطور که در شکل دیده می شود، دامنه بیت صفر به اندازه r و دامنه بیت یک به اندازه ۲r می باشد .در حقیقت $r \cos 2\pi f(t) + \theta$ تبدیل مـی شـود بـه $۲r \cos 2\pi f(t) + \theta$ در گیرنده دامنه r و دامنه ۲r با یکدیگر متفاوتند .دامنه r در گیرنده به بیت صفر و دامنـه ۲r در گیرنـده بـه بیت یک تبدیل می شود.



Bit Rateنرخ ارسال اطلاعات در واحد زمان و Baud Rate نرخ تغییر سیگنال ها در واحد زمان می باشد .یعنی سیگنال هایی که در ثانیه فرستاده می شوند.

FSK

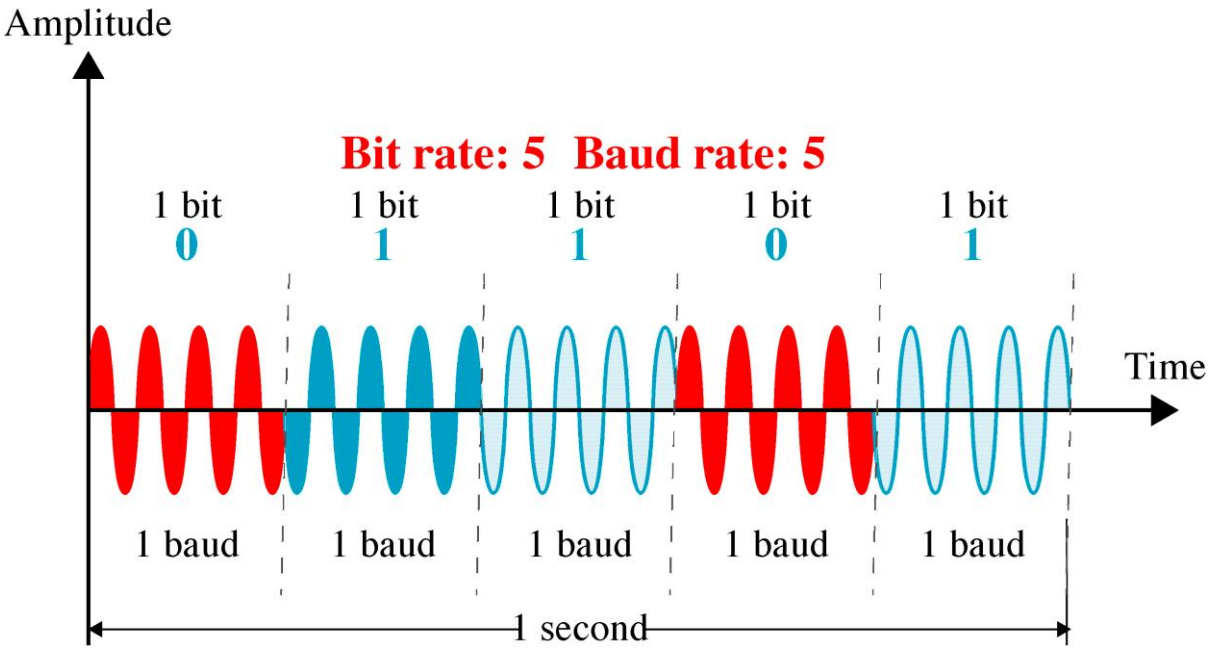
همانطور که گفته شد، در FSK از ویژگی فرکانس برای ساختن سمبل ها استفاده می گردد .در شکل دیده می شود که بیت صفر و بیت یک با هم اختلاف فاز و دامنه ندارند و در اینجا سمبل ها بر اساس تغییر فرکانس ساخته می شوند و گیرنده به راحتی تفاوت بین بیت های صفر و یک را متوجه می شود .چون امواجی با فرکانس متفاوت دریافت می کند .در این شکل Bit Rate و Baud Rate با هم برابرند و ۵ سیگنال در ثانیه فرستاده می شود .برای بیت صفر داریم: $r \cos 2\pi f1(t) + \theta$ و برای بیت یک داریم: $r \cos 2\pi f2(t) + \theta$

PSK

در این حالت فرکانس و دامنه های بیت های صفر و یک با هم برابرند ولی فاز ها متفاوت می باشند .در این حالت سمبل ها با فاز های متفاوت ساخته می شوند.

همانطور که در شکل دیده می شود، در بیت صفر، اختلاف فاز از صفر و در بیت یک از ۱۸۰ شروع شده است و این به راحتی در گیرنده قابل تشخیص است .پس برای بیت صفر داریم: $r \cos 2\pi f2(t) + 0$ و:

برای بیت یک داریم $r \cos 2\pi f2(t) + 180$:در این حالت نیز Bit Rate و Baud Rate با یکـدیگر برابرند .یعنی نرخ ارسال اطلاعات با تعداد سیگنال هایی که در ثانیه فرستاده می شود، برابر است.



راه های افزایش نرخ اطلاعات

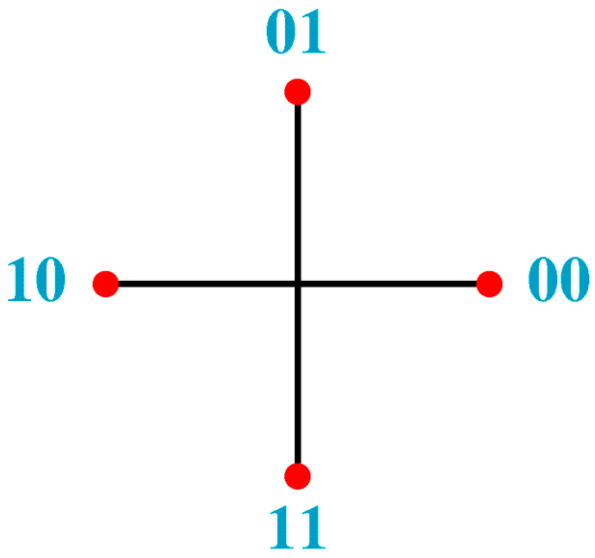
به Baud Rate بر می گردد، اگر بخواهیم Baud Rate را افزایش دهـیم، بایـد پهنـای بانـد را افـزایش دهیم .وقتی که می گوییم Baud ۱۰۰ داریم یعنی ۱۰۰ سیگنال داریم و هر سیگنال دارای یک فرکـانس است و پهنای باند هم یعنی حداقل و حداکثر فرکانسی که از خود عبور می دهد .وقتی مـی خـواهیم ۱۰۰ Baudبفرستیم، باید فاز ها را افزایش دهیم پس باید پهنای باند افزایش یابـد .کـه یـک راه افـزایش نـرخ اطلاعات است.

را دیگر آن است که به جای اینکه هر سیگنال یک بیت را حمل کند، دوبیـت را حمـل کنـد، در حقیقـت تعداد حالت ها را افزایش می دهیم و بدون بدون تغییر در پهنای بانـد مـی تـوان نـرخ انتقـال اطلاعـات را افزایش داد.

در این روش،به جای اینکه سیگنال ها دو حالت صفر درجه و ۱۸۰ درجه داشته باشـند بایـد چهـار حالـت صفر، ۹۰ ، ۱۸۰ و ۲۷۰ درجه داشته باشند .که این چهار معنی مختلف دارد، یعنی دو بیت و اگر گیرنـده سیگنال هایی را با اختلاف فاز صفر دریافت کرد یعنی بیت ۰ ۰ و اگر سیگنالی با اختلاف فـاز ۲۷۰ درجـه دریافت کرد یعنی بیت ۱ ۱ و...

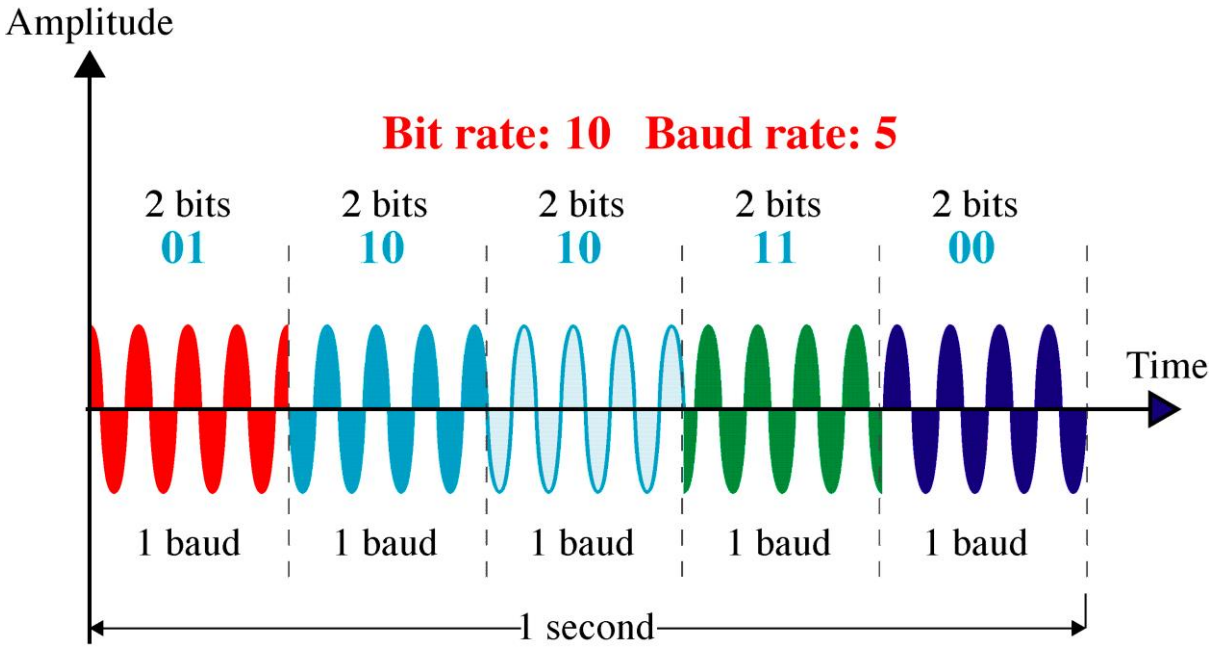
Dibit	Phase
00	0
01	90
10	180
11	270

Dibit
(2 bits)



Constellation diagram

قبلا باکس ورودی یک بیت یک بیت بر می داشت و می خواند و حال دوبیت دوبیت بر می دارد و می خواند .در این حالت، ۱۰ بیت ارسال شده در حالیکه Baud Rate ، ۵ است یعنی فقط ۵ سیگنال فرستاده می شود .در حقیقت با این کار نرخ ارسال اطلاعات بیشتر و دو برابر شده است .می توان از این روش استفاده کرد و نرخ انتقال اطلاعات را تا ۱۰ برابر نیز افزایش داد .چون در این حالت فقط از چهار حالت مختلف استفاده شده و ما می توانیم از ۸ ، ۱۶ ، ۳۲ ، ۶۴ و ... حالت مختلف نیز استفاده کنیم .این تکنیک بیشتر در مودم ها استفاده می شود.



QAM :

می توان از ترکیب این ویژگی ها برای افزایش نرخ انتقال اطلاعات استفاده نمود .یعنی اگر علاوه بر چهر حالت که برای تغییر فاز گفته شد، r و ۲r را در نظر بگیریم.

در اینجا ۸ حالت رخ می دهد، یکی اختلاف فاز صفر، ۹۰ ، ۱۸۰ ، ۲۷۰ با دامنه r و دیگری اختلاف فاز صفر، ۹۰ ، ۱۸۰ ، ۲۷۰ با دامنه ۲r که روی هم ۸ حالت رخ می دهد.

مودم ها استاندارد هایی دارند .از جمله آن استاندارد ها می توان به V_۳۲ اشاره نمود یعنی از ۳۲QAM استفاده می کند و ۳۲ حالت ختلف را با ۵ بیت حمل می کند که سرعت آن ۹۶۰۰bps می باشد.استاندارد دیگر ، V_۳۲BIS با ۶۴QAM یعنی ۶ بیت حمل می کند و سرعت آن به ۱۴۴۰۰bps می رسد.استاندارد دیگر V_۴۲ است که ۴۰۹۶QAM دارد یعنی هر سیگنال ۱۲ بیت را حمل می کند و سرعت آن به ۲۸۸۰۰bps رسیده است .استاندارد بعدی V_۴۲BIS است که سرعت آن ۳۲۶۰۰bps می باشد و ماکزیمم سرعت انتقال اطلاعات روی خطوط تلفن می باشد.

قانون نایکوئیست

$$c = ۲WLog_۲^M$$

برای اینکه بتوانیم ماکزیمم نرخ اطلاعت © روی یک باند را بدست آوریم از این قانون استفاده می کنیم .که wپهنای باند و M تعداد سمبل ها و یا تعداد حالت هایی است که یک Symbol به خود می گیرد .در FSKیا ASK ، M برابر ۲ است چون تعداد حالت ها در آنها برابر می باشد .پس وقتی M=۲ خواهیم داشت:

$$c = ۲W \Rightarrow c = ۲WLog_۲^۲$$

بنابراین می توان گفت قانون نایکوئیست برای محیط های عادی از نویز و محیط هایی که ایده آل هستند برابر(۲W دو برابر پهنای باند)می باشد .

قانون شانول

گفته می شود که محیط های ما محیط های عاری از خطا نیستند و محیط های نویزی هستند، پس از قانونی بنام شانول به صورت زیر استفاده می شود:

$$c = ۲W Log_۲^{(۱+s/n)}$$

$$\left(\frac{s}{n}\right) db = ۱۰\ logs/n$$

مثال :برای انتقال اطلاعات روی خطوط تلفن به وسیله یک مودم، اگر نسبت سیگنال به نویز ۳۰ دسی بل باشد، ماکزیمم نرخ انتقال اطلاعات در این محیط برابر است با:

$w = ۳۴۰۰ - ۳۰۰\ HZ = ۳۱۰۰Hz$ (پهنای باند)

$$\frac{s}{n} db = ۳۰ = ۱۰ \log \frac{s}{n} \Rightarrow \frac{s}{n} = ۱۰۰$$

$c = ۲ \times ۳۱۰۰ \log_{۲}^{۱ + ۱۰۰۰} = ۶۱۷۸۷\ bps$ (نرخ انتقال اطلاعات)

نسبت توان سیگنال به توان نویز هر چه بیشتر باشد یعنی به راحتی می توان نویز را از سیگنال جدا کرد.

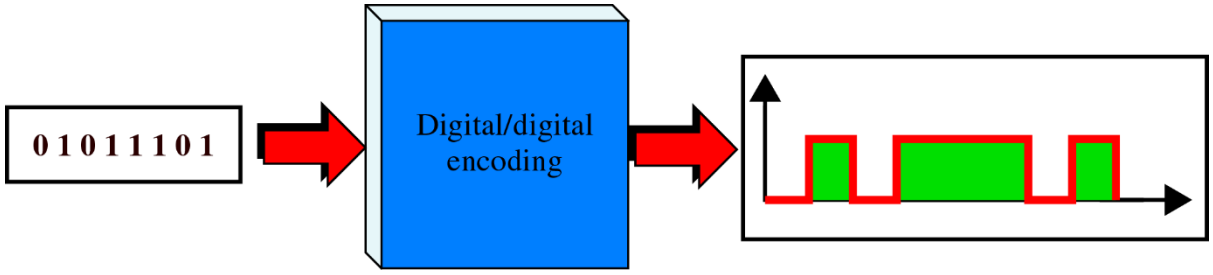
در محیط اگر S/N را افزایش دهیم، در حقیقت C یا Transfer Rate را افزایش داده ایم.

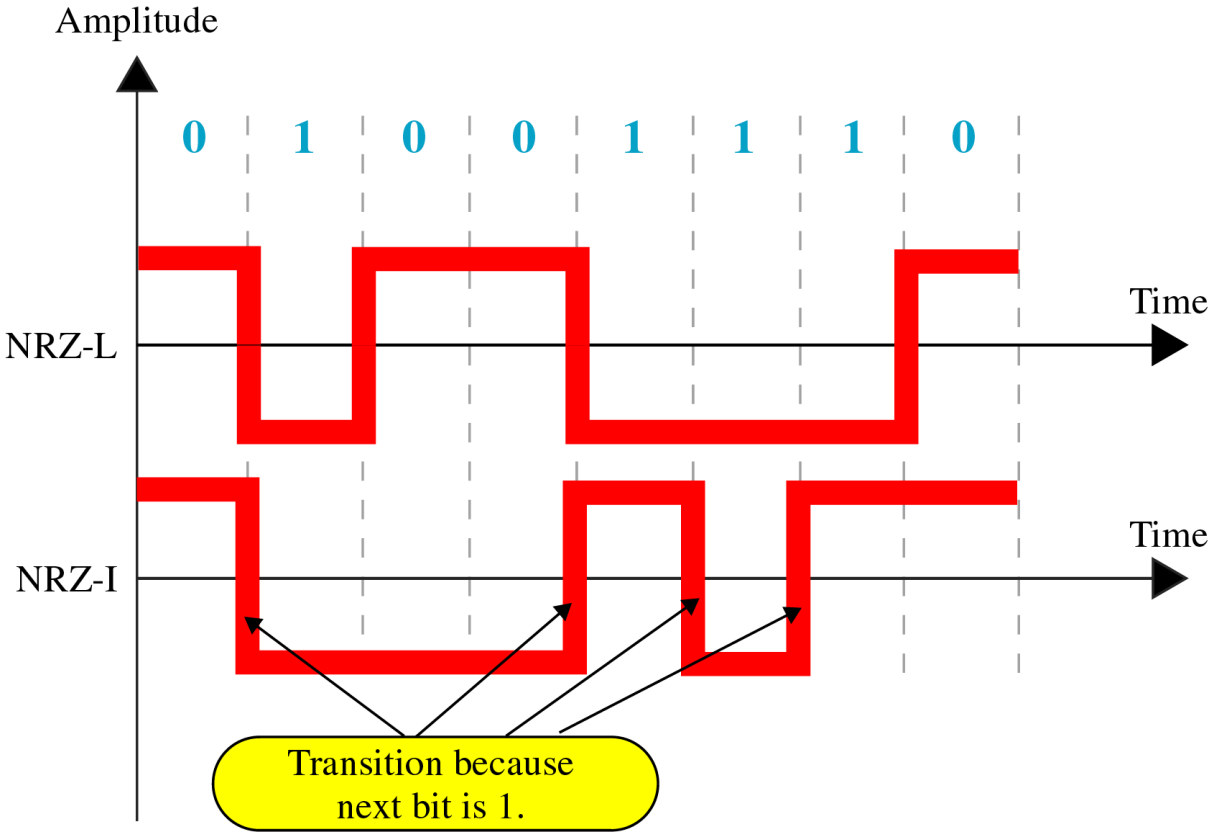
Digital to Digital Encoding

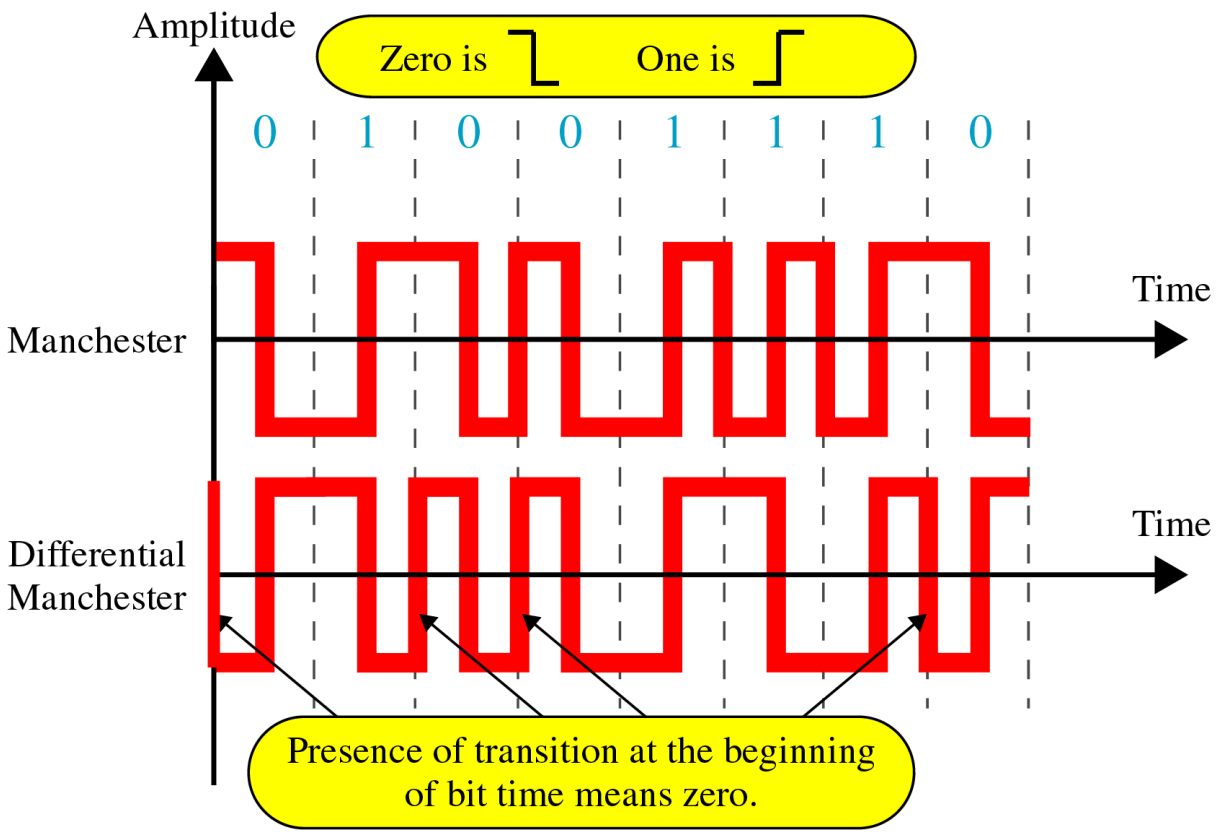
متنعی دیگر وجود دارد که در آن ورودی داده دیجیتالی و خروجی سیگنال دیجیتال می باشد .مثال ارتباط pcبه pc یا به مودم و یا pc به printer و ... سیگنال آنالوگ موجی است پیوسته و سیگنال دیجیتال موجی است گسسته.

به عنوان روشی دیگر می توان از این روش نام برد که در آن برای بیت صفر یک نیم پالس مثبت و یک نیم پالس منفی می فرستد و برای بیت یک، یک نیم پالس منفی و یک نیم پالس مثبت می فرستد .در این حالت امکان اینکه صفر ها و یک های طولانی به وجود آید، نداریم یک مشکل در این روش وجود دارد و آن اینکه پهنای باند در این روش کم می شود.در حقیقت پهنای باند نصف می شود .یک بیت را هر سیگنال حمل می کند و برای یک بیت یک تک ولتاژ منفی و یک تک ولتاژ مثبت استفاده می گردد .در آنالوگ گفته شد که یک سیگنال بیت های بیشتری را حمل می کند ولی در اینجا دقیقا برعکس است یعنی یک بیت را هر سیگنال حمل می کند و به همین دلیل پهنای باند نصف می شود .ولی این روش عملا ویژگی Self Clockرا دارد .یعنی با فرستادن بیت، Clock را هم ارسال می کند .کاربرد این روش در شبکه

های LAN می باشد .در شبکه ها نیز از این روش استفاده می کنند چون عموما شبکه دارای پهنـای بانـد بالایی است و به دلیل ویژگی Self Clock استفاده می گردد.

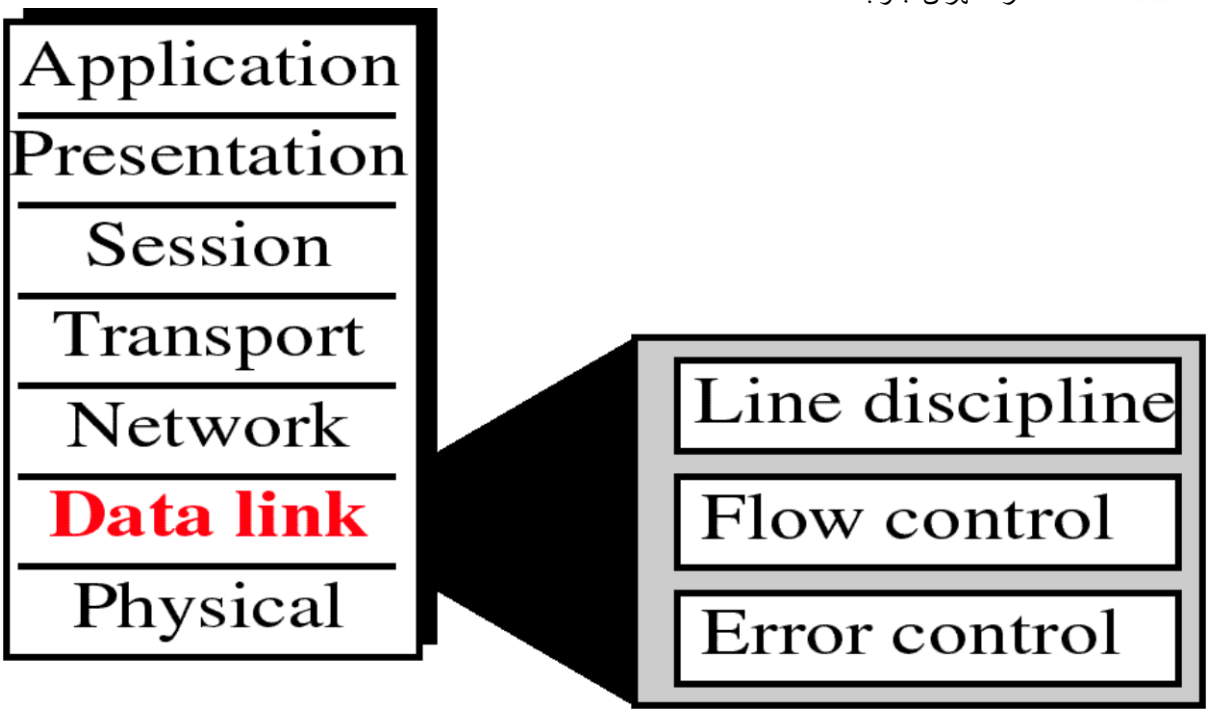






لایهData Link

همانطور که گفته شد، محیط های ارتباطی ما محیط های عاری از خطا نمی باشند و امکان خطا در آن ها وجـود دارد .گفتـیم کـه لایـه Physical ، سـطوح بیـت هـا، عـرض بیـت هـا، روش هـای Coding ، Interfacingرا مشخص می کند و نه بیشتر .بنابر این در برابر خطا عکس العملی نشـان نخواهـد داد .در لایه Physical حتی شکل کانکتور ها هم مطرح است که باید از استاندارد خاصی پیروی کنند (Com۱ ، Com۲و (... پس مشکل خطا را لایه Data Link حل می کند .ممکن است فرستنده اطلاعات را سریع ارسال کند و گیرنده نتواند دریافت کند و بخشی از اطلاعات از بین برود .لایه Physical قـط بیـت هـای صفر و یک را به سیگنال مناسب تبدیل می کند و متوجه وجود خطا نمی شود، بنابراین لایه Data Link این کار را انجام می دهد .همچنین ممکن است وقتی که ما مـی خـواهیم اطلاعـاتی را بفرسـتیم، گیرنـده آنلاین نباشد و نتواند اطلاعات را دریافت کند .باز هم وظیفه لایه Data Link است که اطـلاع دهـد و مـا اطلاعات مورد نظر را نفرستیم .لایه Data Link دارای چهارچوبی اسـت کـه در قالـب آن چهـارچوب بـه وظایف خود عمل می کند و این چهارچوب از سه قسمت تشکیل شده است:



قسمت Line Discipline مشخص می کند که چه کسی اول اطلاغات را بفرستد.

قسمت Flow Control مشخص می کند که چه مقدار اطلاعات بفرستیم تا گیرنده قادر بـه دریافـت آن باشد.

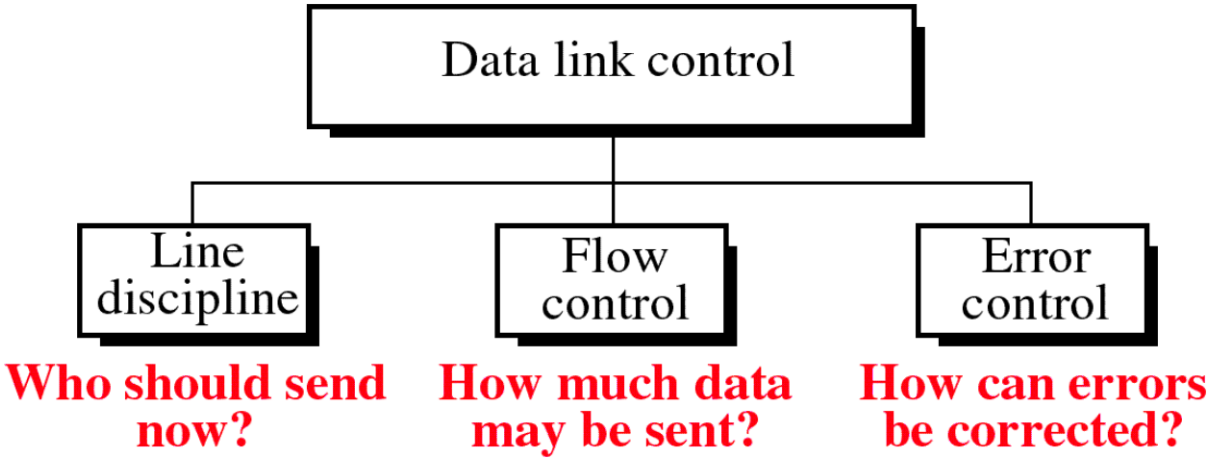
قسمت Error Control مشخص می کند که خطا چگونه تشکیل داده شود.

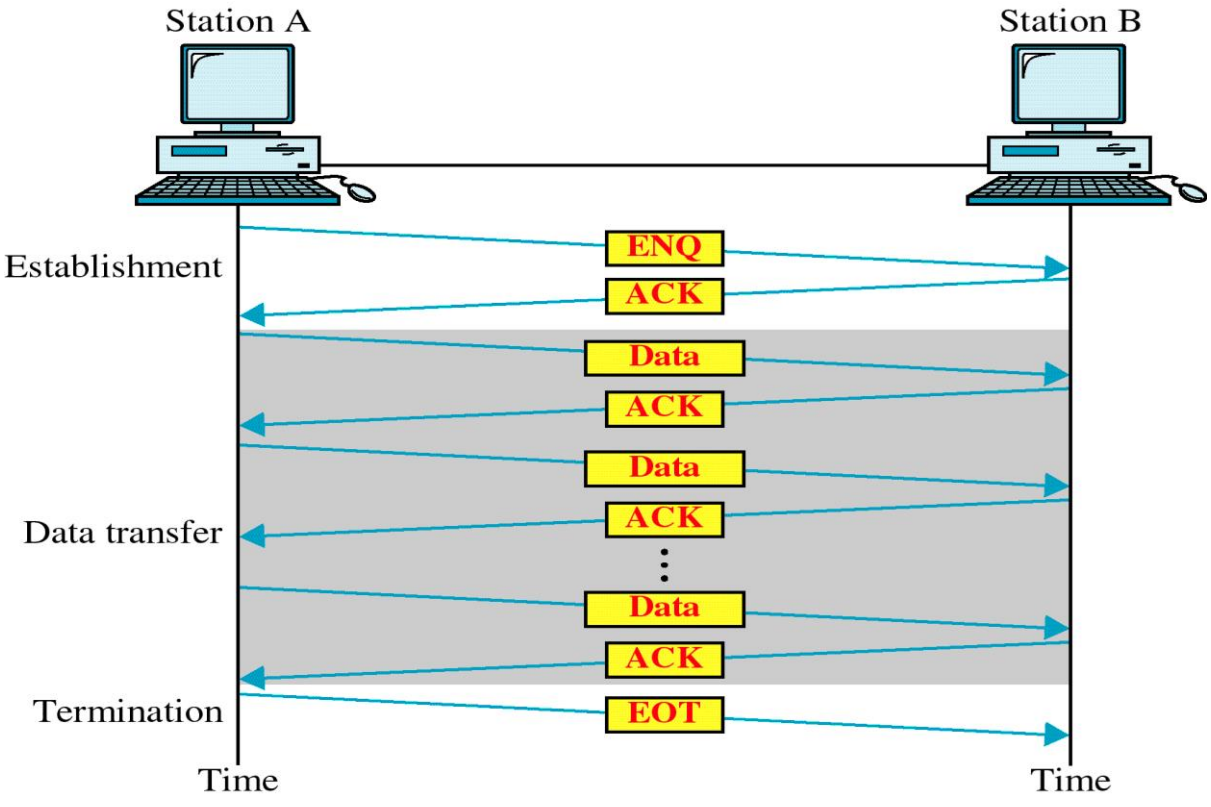
پس این سه قسمت وظایف لایه Data Link را انجام می دهد.

Line Discipline

در این مثال رو Station دیده می شود .این روش، روش بسیار ساده اس است که در آن کافی است قبل از فرستادن اطلاعات فرستنده یک درخواست ENQ می فرستد و گیرنده نیز برای تأیید ACK را می فرستد و پیغام می دهد که آماده دریافت می باشد.واحد تبادل اطلاعات در لایه Data Link ، Frame است که همین Frame از دیدگاه لایه Physical ، بیت است و فریم را به صورت مجوعه ای از بیت ها دریافت می کند .کافیست فرستنده فریمی را که می سازد، یـک بیـت از آن را بـه بیـت درخواسـت ارتبـاط (ENQ) اختصاص دهد و آن را ON کند و به لایه فیزیکی بدهد .لایه فیزیکی فریم دریافت شده را به صورت صفر و یک تبدیل کرده و روی خط می فرستد.لایه ی فیزیکی گیرنده این صفر و یک ها را دریافت کرده و تحویل لایه Data Link می دهد و چون هم پروتکل) قرارداد (هستند می بیند که یک درخواست ارتبـاط آمـده است و کافیست گیرنده همین بیت را On کند و به عنوان تأییده بیت ACK را ON می کند) بیت ۵ .(

فرستنده می بیند که فریمی آمده و بیت پنجم آن یک شده استپس برقراری ارتباط تأیید شده است .ایت بیت ها، بیت های کنترلی هستند که هنگام تولید یک فریم ساخته می شوند .هنگامی که بیت های کنترلی ساخته می شوند، دوبیت ENQ و ACK را در نظر می گیریم و به عنوان پروتکل یا قرارداد منظور می کنیم که هرگاه ENQ ، یک شد یعنی در خواست ارتباط و هرگاه ACK یک شد، یعنی تأییدیه درخواست ارتباط .همانطور که در شکل دیده می شود، همانطور اعلام ارتباط می شود از طرف فرستنده، باید قطع ارتباط نیز اعلام گردد.(EOT)

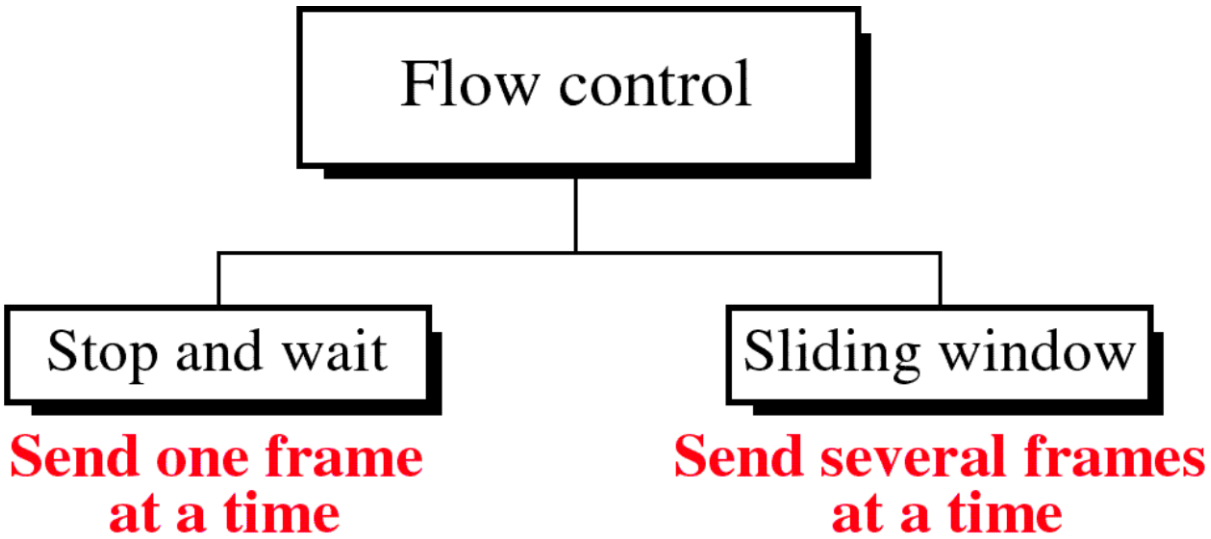




چون محیط های ارتباطی مااغلب محیط های اشتراکی است، بحث دیگری که مطرح می شود مسئله زمان بندی است .اگر ما اطلاعات زیادی را یکباره روی خط قرار دهیم، خط به مدت زیادی اشغال خواهد بود و نوبت به دیگران نخوهد رسید .پس زمان بندی می کنیم و در واقع اطلاعات را در قالب های کوچک تر می شکنیم و در این صورت خط زیاد اشغال نخواهد شد و دیگران هم می توانند از آن استفاده کنند که این یکی از ویژگی های شکستن اطلاعات به واحد های کوچکتر است .از ویژگی های دیگر شکستن به واحد های کوچک تر این است که اگر یک بیت از کل اطلاعات خراب شد و از بین رفت، لازم نیست که کل ازلاعات دوباره ارسال شود و همان فریمی که بیت آن خراب شده، دوباره ارسال می گردد .شکستن اطلاعات به واحد های کوچکتر نیز از وظایف لایه Data Link است که اطلاعاتی را که حجم زیادی دارند به ۱۰۰ یا ۵۰۰ فریم می شکند و کل اطلاعات را در قالب یک فریم نمی فرستد .این روش زمان بیشتری می برد ولی در مقایسه با این که اطلاعات را از دستنمی دهیم روش خوبی است.

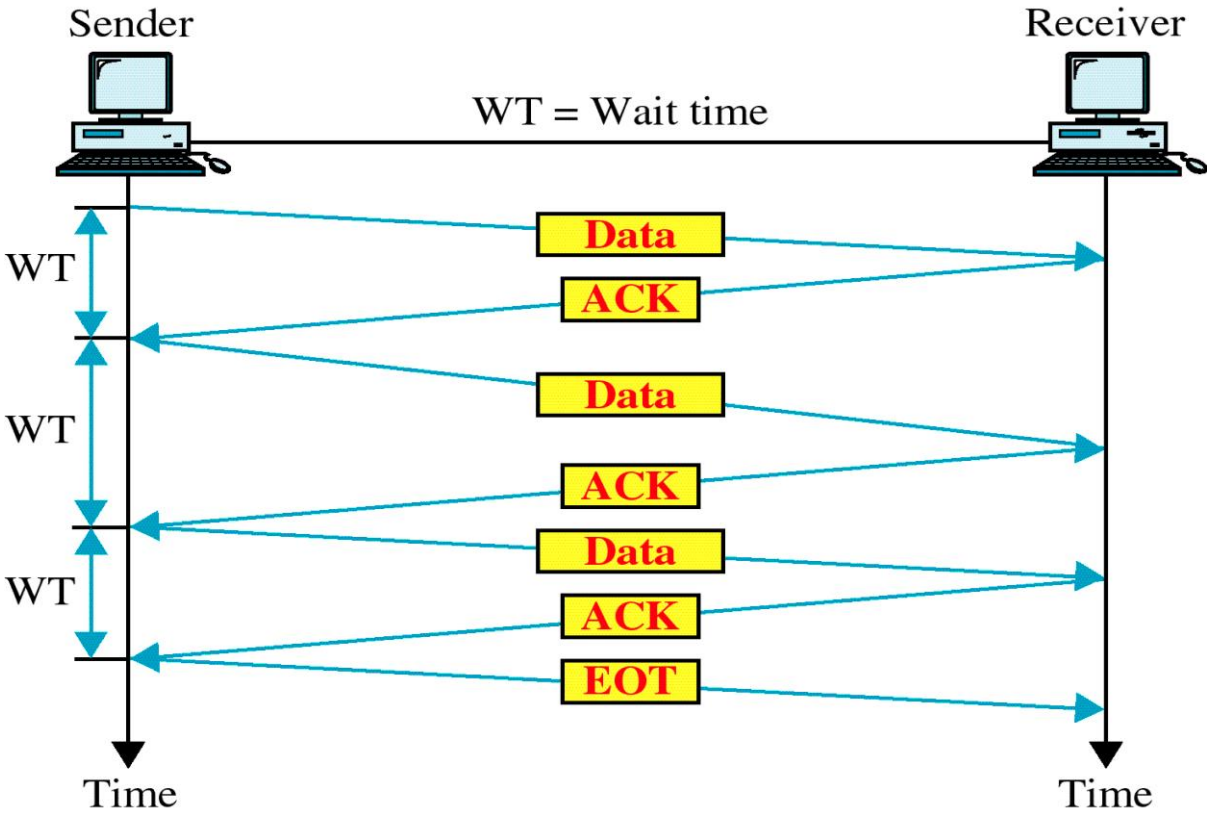
کنترل جریان(Flow Control)

وقتی که فرستنده حجم زیادی از اطلاعات را می فرستد و گیرنده نمی تواند آن را دریافت کند، بخشی از اطلاعات از بین خواهد رفت .برای جلوگیری از این مشکل دو روش ارائه شده است.



روشStop and Wait

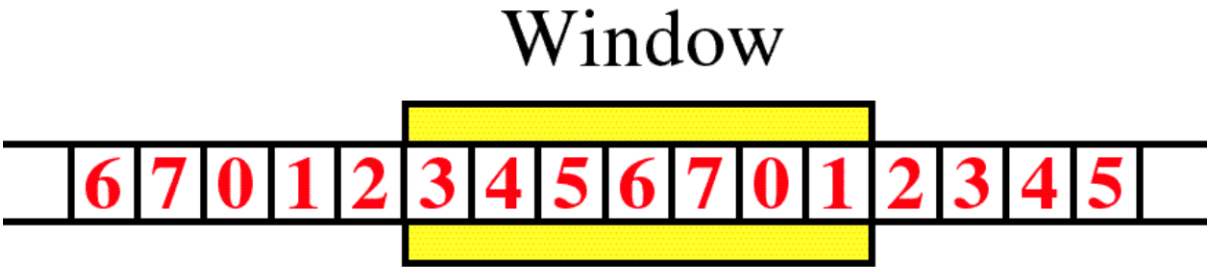
در این روش یک فریم از طرف فرستنده ارسال می شود و صبر می کند تا تأییدیه در یافت فریم بیاید و بعد فریم بعدی را ارسال می کند .در این روش، گیرنده دقیقا فرستنده را کنترل می کند و پس از فرستادن هر فریم، تأییدیه آن را ارسال می کند .در واقع در این روش در آن واحد یک فریم بیشتر منتقل نمی شود .در این روش، wait time (wt) وجود دارد و چون Data با تأخیر به گیرنده می رسد) تأخیر انتشار (و این تأخیر ناشی از بعد مسافت و سرعت نور می باشد .زمان تأخیر انتشار عموما کم در نظر گرفته می شود.



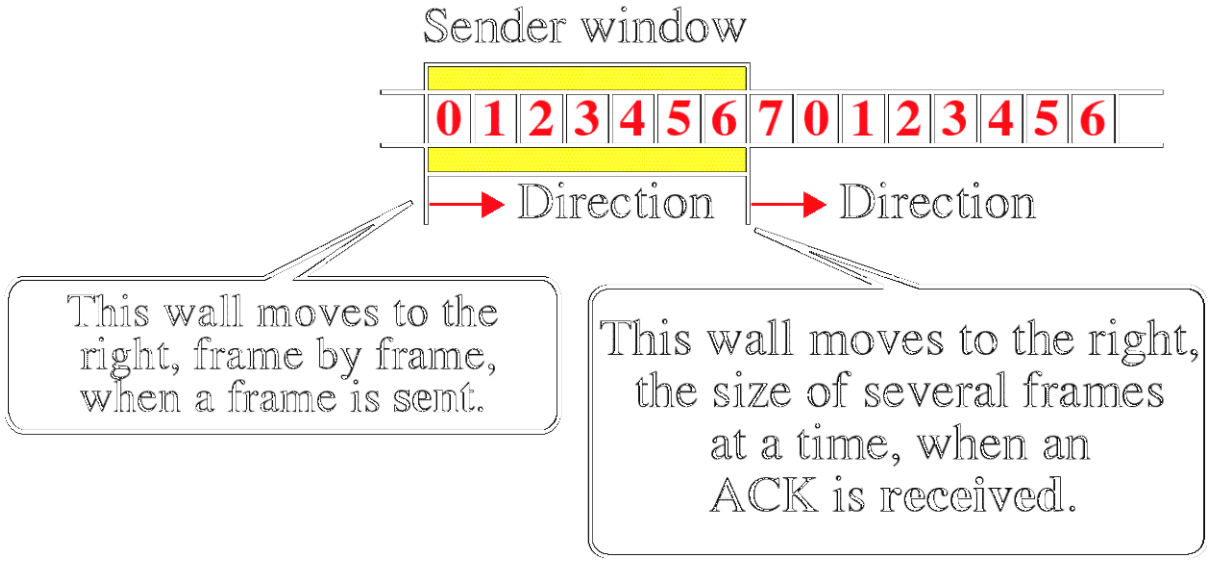
در شبکه های عادی چیز هایی که ما را محدود می کند زمان تولید فریم است نه زمان تأخیر انتشار .در شبکه های High Speed مثل ATM ، دقیقا برعکس است یعنی زمان تآخیر بیشتر از زمان تولید Frameاست .زمان تولید فریم به مراتب کاهش می یابد ولی زمان انتشار افزایش می یابد.

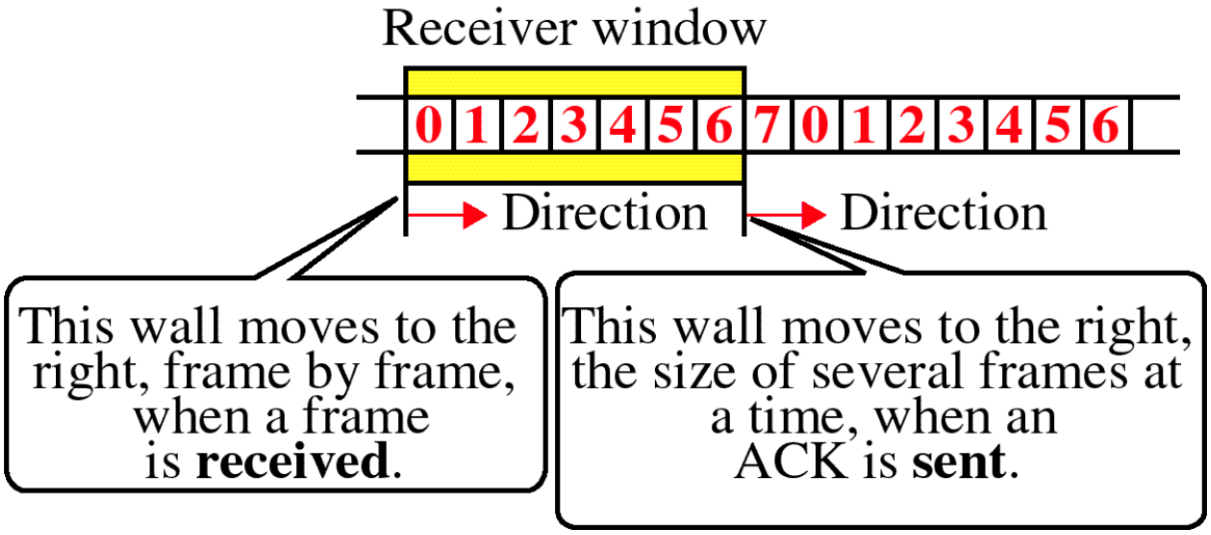
روش پنجره لغزان(Sliding Window)

در این روش فرستنده برای اینکه زیاد بیکار نماند، چندین Frame را به طور هم زمان ارسال می کند بدون اینکه تأییدیه ای بیاید.



در روش Sliding Window ، گیرنه تعداد فریم ها را محدود می کند .مثلا گیرنده می گوید که در آن واحد می تواند ۱۰ فریم را دریافت کند و نه بیشتر و فرستنده هم ۱۰ فریم بیشتر نمی تواند بفرستد .به این ترتیب، گیرنده می تواند فرستنده را کنترل کند.





در این روش چندین فریم پشت سر هم ارسال می شود بدون اینکه تأییدیه دریافت شود .اما تنها به اندازه سایز پنجره می توان فریم بدون تأییدیه ارسال نمود.

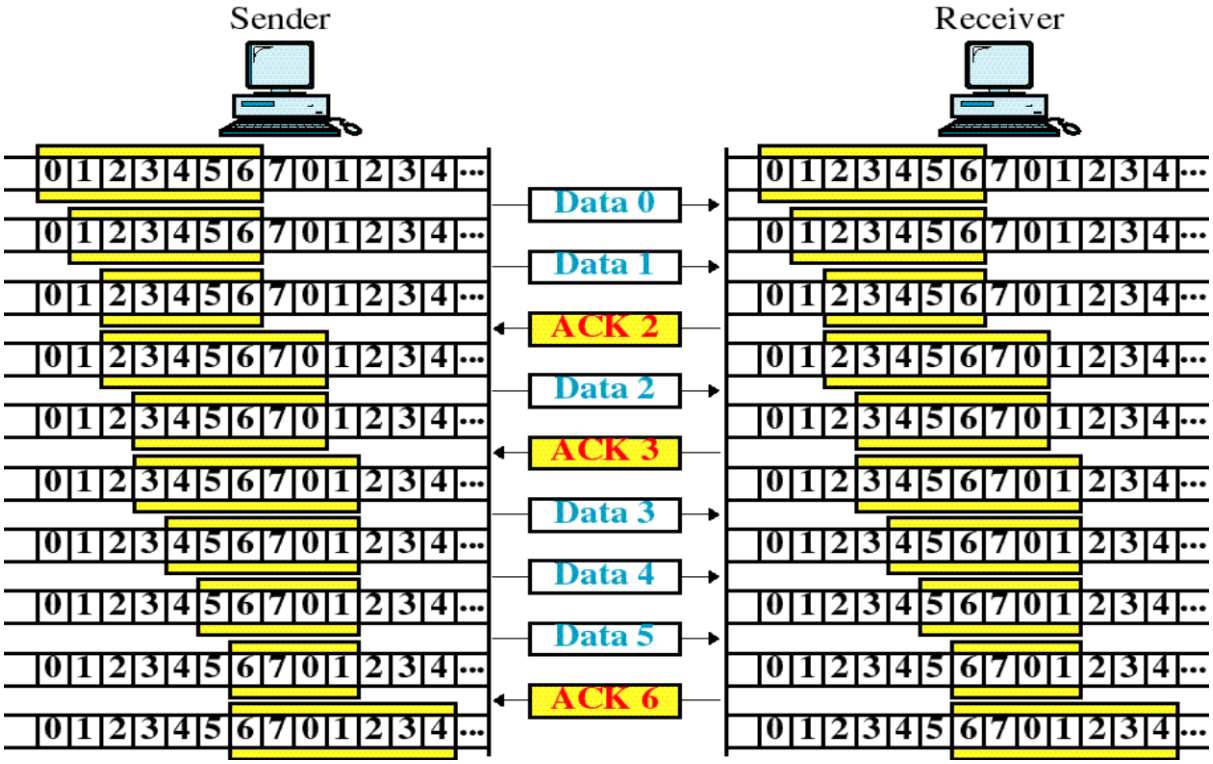
Frameها باید در حجم کوچک باشند در این صورت اولا خط کمتر اشغال می شود و به دیگرن اجازه استفاده از خط داده می شود و ثانیا در صورت رخ دادن خطا در هر یک از فریم ها تنها کافی است فریمی که دارای خطا بوده دوباره ارسال شود.

در این روش، به هر یک از فریم ها یک شماره ترتیبی تخصیص داده می شود به عنوان مثال اگر سایز پنجره ۸ تایی باشد یعنی ۳ بیتی و با ۳بیت می توان ۸ حالت مختلف را نمایش داد پس می توان شماره ترتیب فریم ها را از صفر تا ۷ و به همین ترتیب در نظر گرفت.

پس با در نظر گرفتن یک شماره ترتیب ۳بیتی می توان OverHead فریم ها را به مراتب کاهش داد و به جای استفاده از شماره های صفر تا ۱۰۰۰ و یا بیشتر که یک Sequence Number ، ۱۰ یا ۲۰ بیتی لازم دارد از Sequence Number ، ۳بیتی استفاده نمود .فرستنده یک پنجره دارد که مشخص کننده شماره فریم هایی است که می خواهد ارسال کند .گیرنده هم بر مبنای همین ایده یک پنجره دارد که مشخص کننده فریم هایی است که می خواهد دریافت کند .این پنجره مثل یک آرایه ای است که داخل آن یک سری عدد و دارای ۲ ایندکس می باشد .پنجره مشخص کننه شماره فریم هایی است که آماده ارسال هستند.به ازای ارسال فریم دیواره سمت چپ حرکت کرده و جلو می رود و به ازای دریفت تاییدیه فریم دیواره سمت راست حرکت می کند) در فرستنده .(بنابراین پنجره از یم طرف جمع شده و از طرف دیگر باز می شود .به همین دلیل به آن پنجره لغزان گفته می شود .وقتی فرستنده فریم ها را ارسال می کند، دیواره را یکی جلو تر می برد و کپی فریم را نگهداری می کند که اگر اتفاقی برای فریم افتاد، بتواند آن را دوباره ارسال کند .پس بایستی یک بافر برای نگهداری فریم های ارسالی داشته باشیم در گیرنده نیز به همین ترتیب پنجرا ای وجود دارد که برای آن مشخص می کند که چه شماره فریمی را باید دریافت نماید

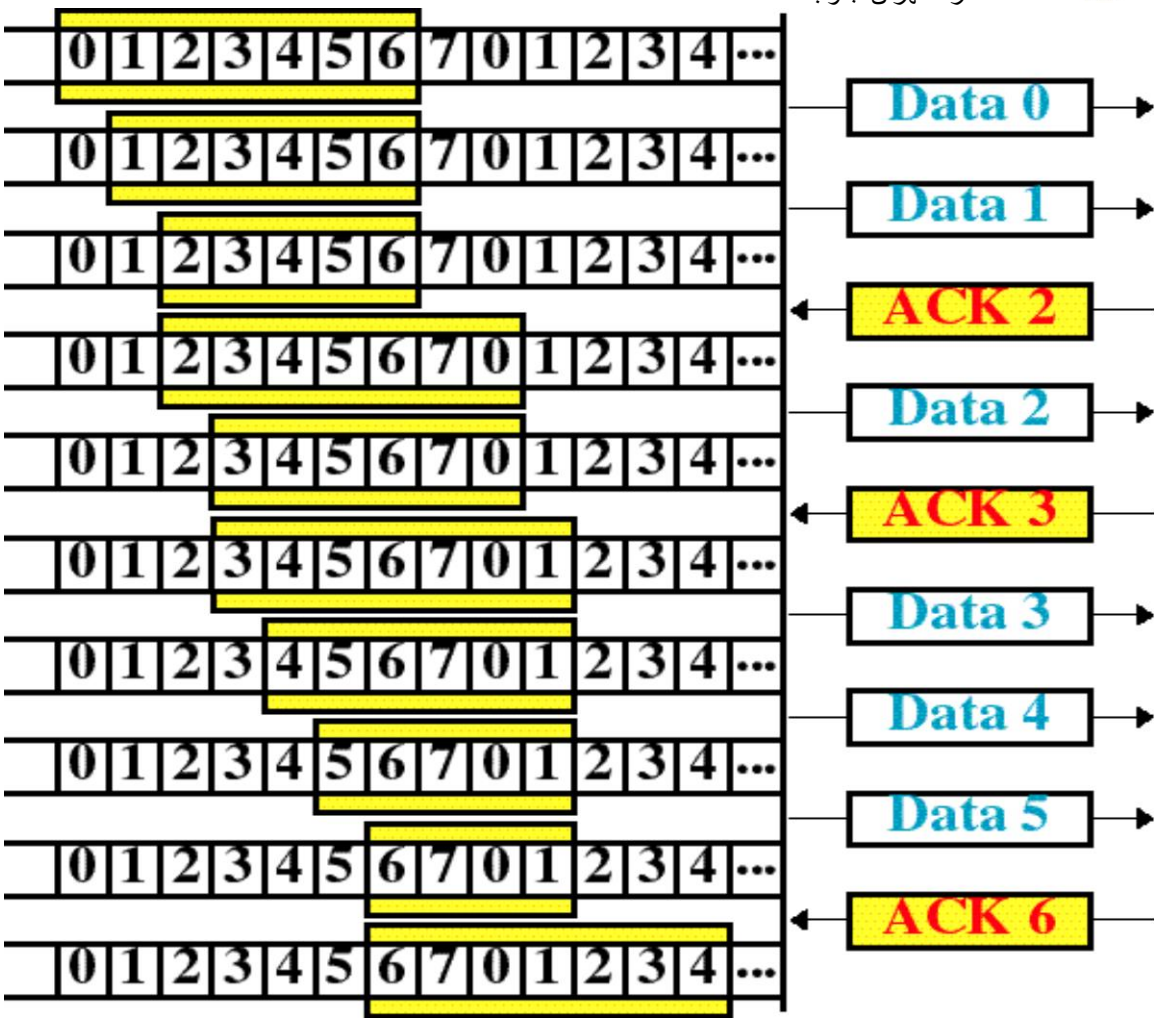
وقتی که فریم ها ارسال می شوند) فریم اول، دوم و سوم(، از طرف گیرنده تأییدیه برای فرستنده ازسال می گردد که به ازای این تأییدیه دیواره سمت راست حرکت می کند .اگر فرستنده فریم ها را تا شماره ۶ ارسال نماید و هیچ تأییدیه ای دریافت نکند، نمی تواند فریم بعدی را ارسال کند چون با گیرنده توافق کرده بود که به اندازه سایز پنجره فریم ارسال نماید.

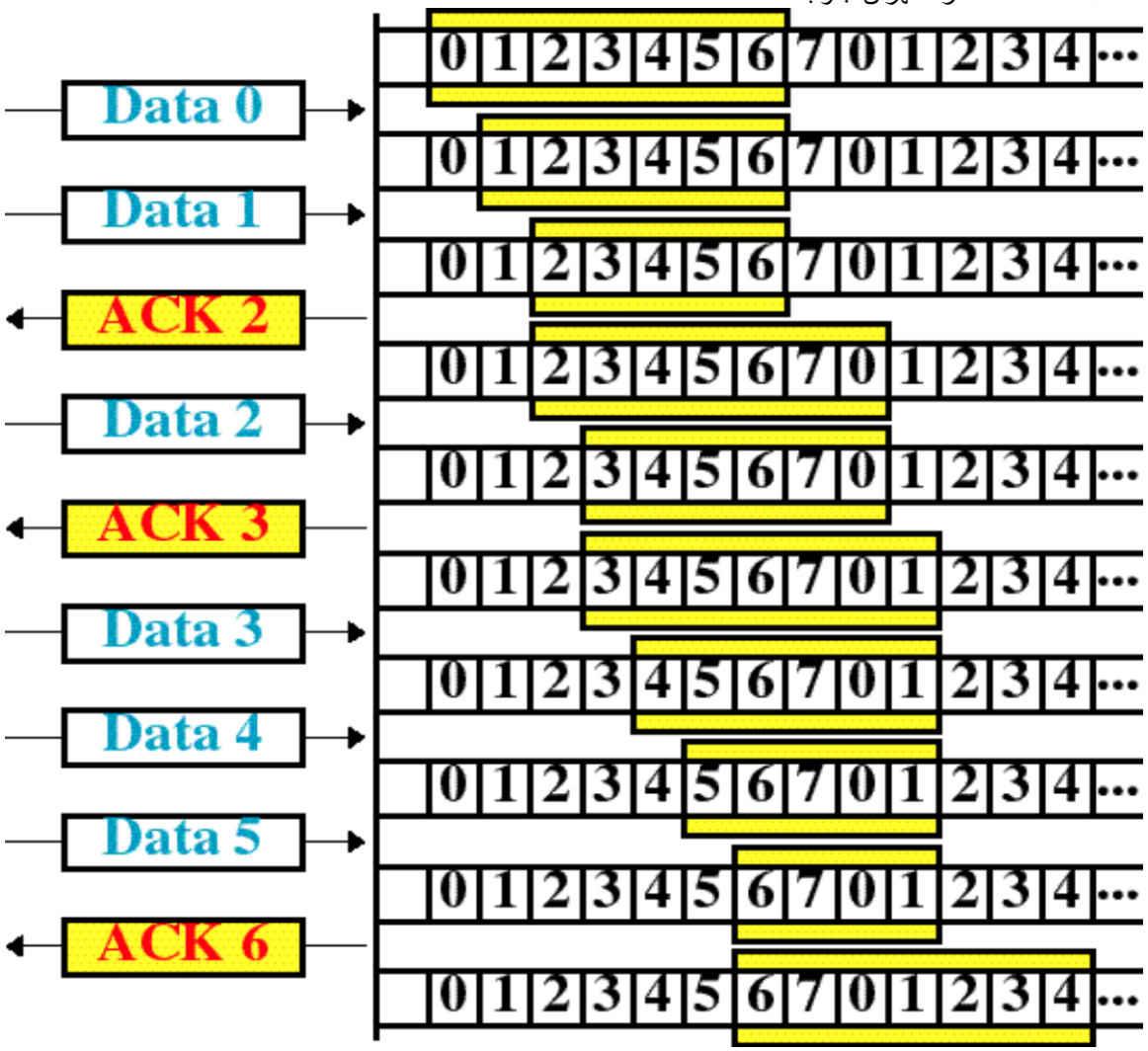
مثال پنجره لغزان:



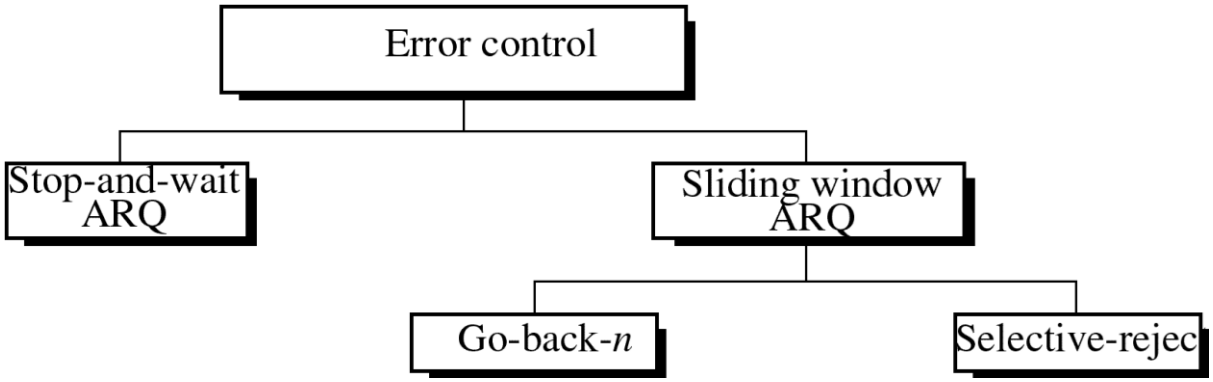
گفتیم Sliding Window مشکل فرستنده گیرنده را حل می کند همینطور که Stop & Wait این مشکل را حل می کرد ولی به کندی .در این مثال، فرستنده و گیرنده بر روی سایز ۷تایی توافق کرده اند . یعنی فرستنده می تواند ۷ فریم ارسال کند بدون اینکه تأییدیه ای دریافت کند ولی بدون تأییدیه بیش از ۷ فریم را نمی تواند ارسال نماید .فرستنده فریم شماره صفر و یک را می فرستد و گیرنده در قبال آن ACK ۲را ارسال می کند یعنی من فریم شماره صفر و یک را دریافت کردم و متظر فریم شماره ۲ هستم .به ازای این تأییدیه دیوار سمت راست پنجره به اندازه ۲ فریم) فریم های ۷ و صفر (جلو می رود .بابراین این پنجره مدام روی رسته ای از بیت ها حرکت می کند.

در روش Sliding Window می خواهیم زمان Wait Time در روش Stop & Wait را به حداقل برسانیم.


0 1 2 3 4 5 6 7 0 1 2 3 4 ...
Data 0
Data 1
ACK 2
Data 2
ACK 3
Data 3
Data 4
Data 5
ACK 6



Error Control



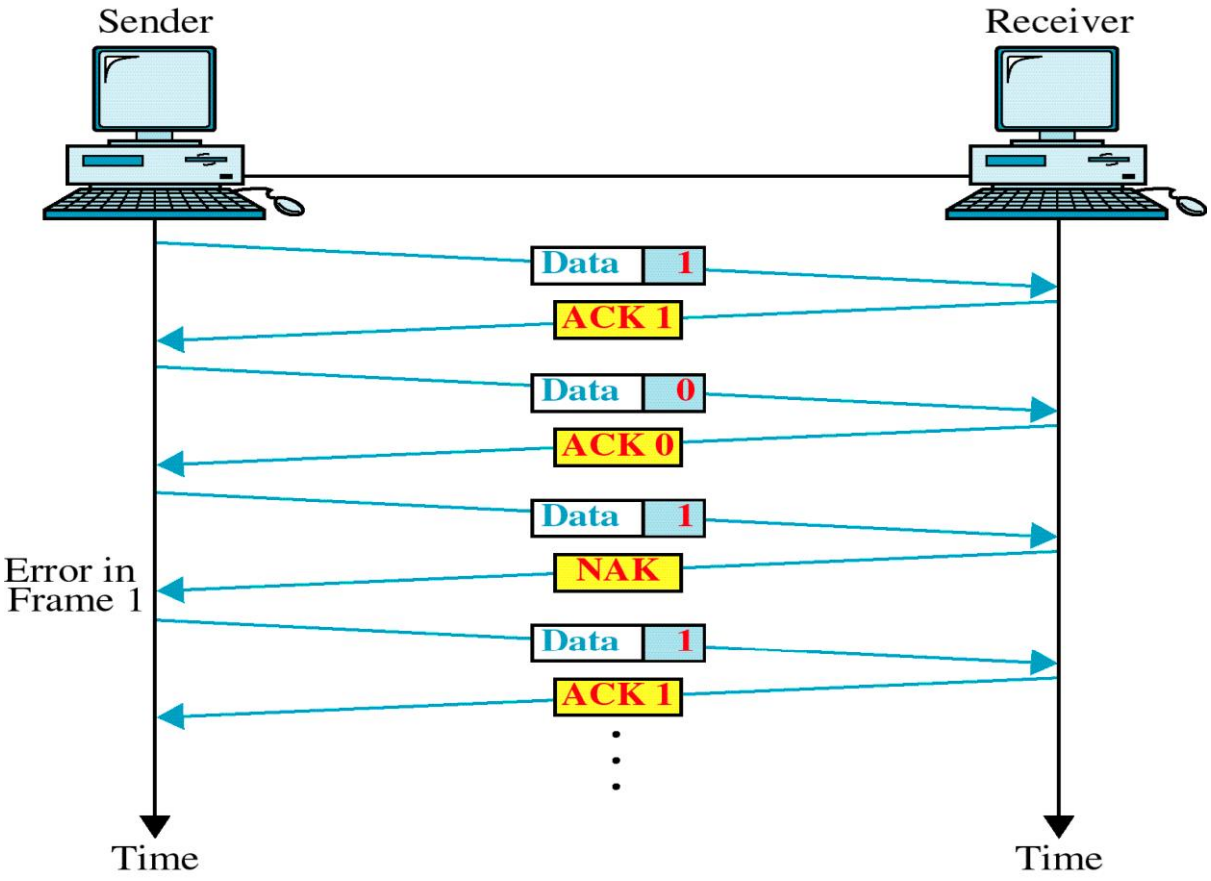
محیط های ما محیط های عاری از خطا و ایده آل نیستند و امکان رخ دادن خطا در آنها وجود دارد .فـریم ممکن است گم شود و یا اینکه فریمی که روی خط ارسال می شود، در اثر وجود نویز Data آن تغییر کند و در گیرنده فریم خراب دریافت شود .پس فریم در ارسال اطلاعات یا Lost می شود و یا.Damege

چون لایه ی فیزیکی نسبت به این خطا ها عکی العمل نشان نمی دهد، این وظیفه لایه ی Data Link است که در قالب Error Control این کار را انجام دهد .برای حل این مشکل، مکانیزم هایی به نام ARQ (Automatic Repeat Request)مطرح می شود.

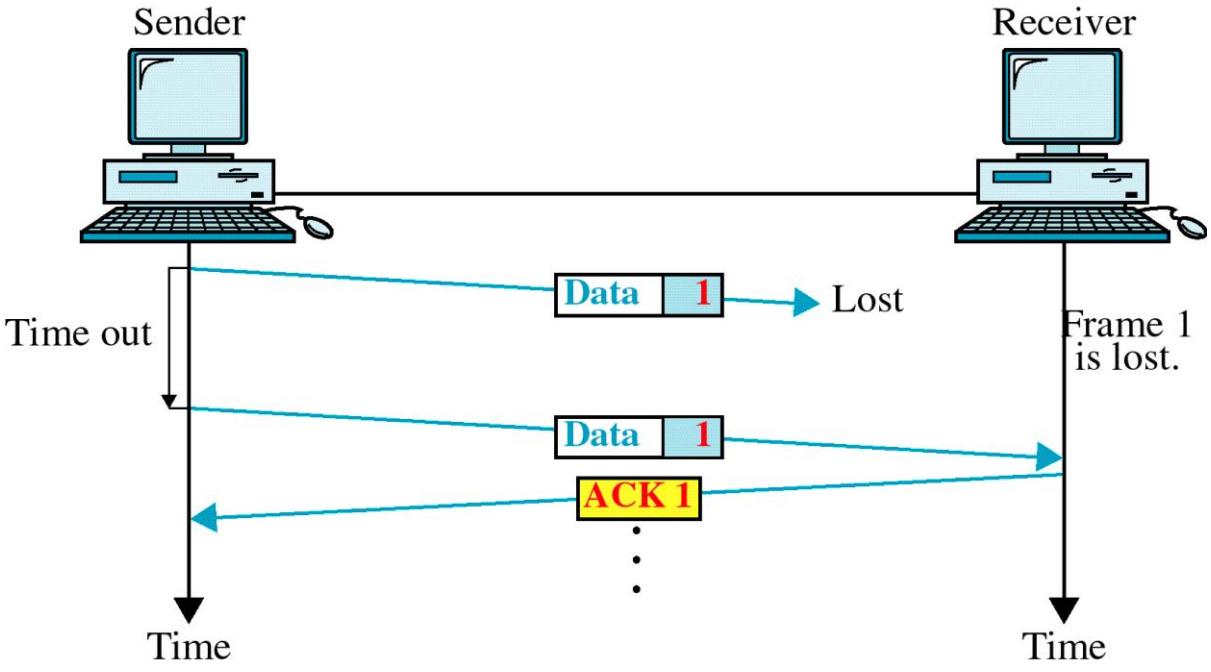
به عنوان مثال ACK ای که از گیرنده به فرستنده ارسال می شود، یک مکانیزم ARQ است که اگر فرستنده فریم را ارسال کند و گیرنده تشخیص دهد که فریم ارسال شده خراب است، کافی است به جای ACK، یک NAK بفرستد که یعنی فریمی که ارسال شده خراب است و باید دوباره ارسال شود .حال اگر فریمی را ارسال کنیم که در بین راه Lost شود، گیرنده هیچ فریمی را دریافت نمی کند، پس تأییدیه ای صادر نمی کند .در اینجا فرستنده از یک Time استفاده می کند و تایمر را Set می کند که اگر بعد از یک مدتی تایمر Time Out داد و جوابی نیامد، فریم دوباره ارسال گردد .بنابراین مکانیزم هایی مانند ACK، NAK و Timer را روش هایی از ARQ در نظر می گیریم از این روش ها استفاده می کنیم تا بتوانیم مشکلات رخ داده در روش های Stop & Wait و Sliding Window حل کنیم.

Stop & Wait ARQ

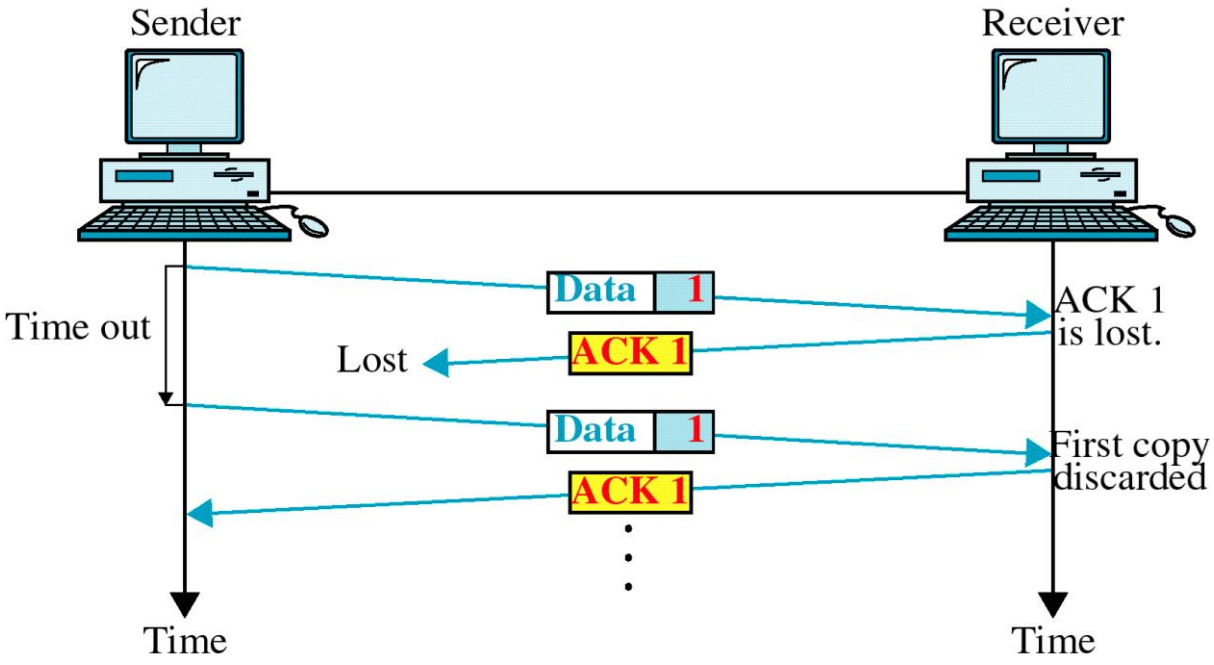
حال باید دید که چگونه می توان مشکلات Stop & Wait را با این مکتنیزم ها حل نمود .در حالت اول فریم ارسال می گردد و در گیرنده به صورت Damege دریافت می شود) یعنی گیرنده فریم را دریافت کرده و توسط مکانیزم هایی در فریم تشخیص خطا می دهد .(در این حالت، کافیست که گیرنده یک NAKبفرستد و فرستنده با دیدن NAK چون کپی فریم را از قبل در بافر خود دارد، دوباره همان فریم را ارسال می کند پس با بیت ساده ای به نام NAK این مشکل حل می شود.



در حالت دوم فریم ارسال می شود و در بین راه Lost می گردد .در این حالت تایمر فرستنده Time Out میدهد و فریم دوباره ارسال می گردد در بد ترین شرایط، یک جواب تایمر بیاید در غیر این صورت این اتفاق رخ داده است.



کم یا زیاد گرفتن زمان Timer خیلی مهم است .زمان Timer باید طوری در نظر گرفته شود که نه تأخیر زیاد شود و نه اطلاعات تکراری فرستاده شود .در حالت سوم ACK گیرنده، Lost می شود .در این حالت هم تایمر فرستنده، Time Out می دهد و دوباره همان فریم ارسال می گردد .پس در گیرنده با فریم های تکراری مواجه می شویم در اینجا کافیست که فریم ها شماره هایی داشته باشند تا گیرنده با دیدن شماره تکراری آن فریم را حذف کند .برای شماره گذاری، یک بیت برای Sequence num در نظر می گیریم که این بیت ها می توانند صفر یا یک باشند، به عنوان مثال بیت اول صفر، بیت دوم یک، بیت سوم صفر و به همین ترتیب .در این حالت گیرنده منتظر دیافت فریم هایی با شماره های صفر،یک، صفر،یک و به همین ترتیب می باشد و با دیدن شماره تکراری آن را Discard می کند.



Sliding Window ARQ

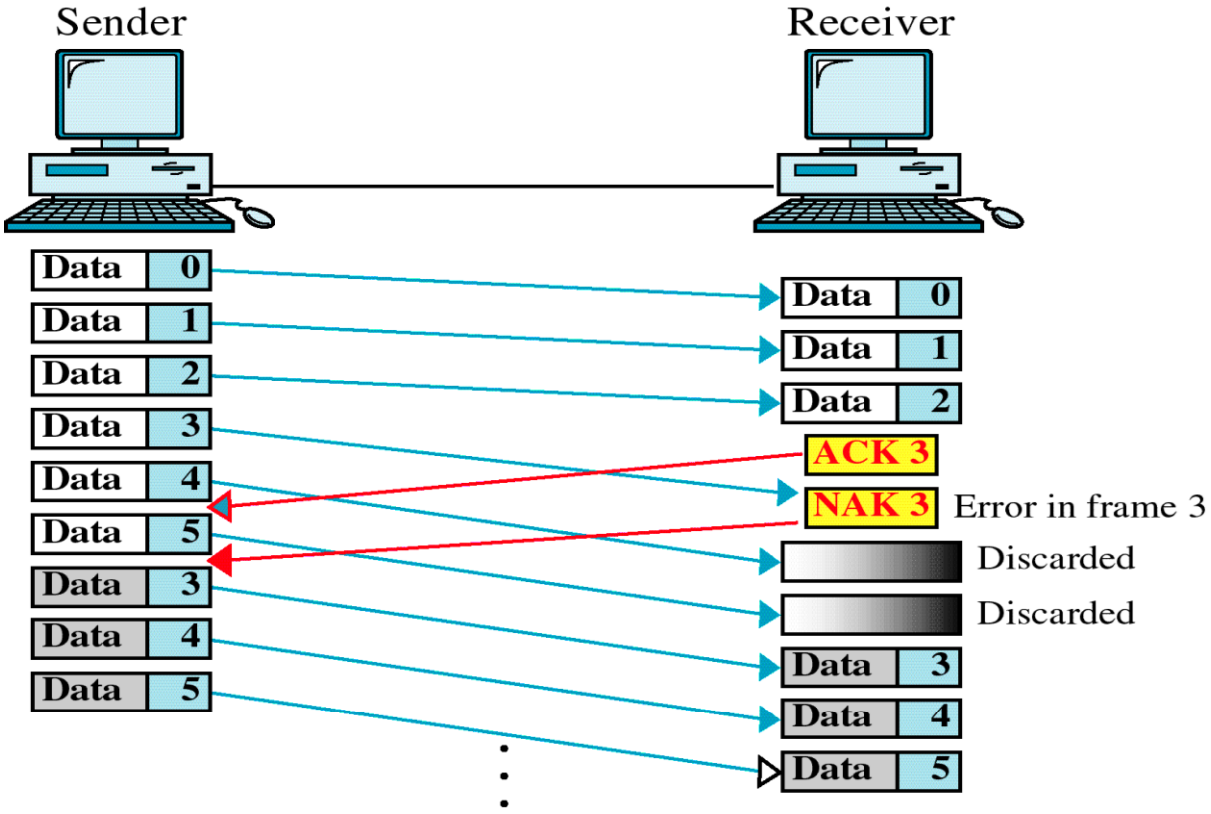
در پنجره لغزان دو روش برای حل این مشکل وجود دارد:

۱) Go_Back_n .بازگشت به شماره(n

۲) Selective Reject .تکرار انتخابی(

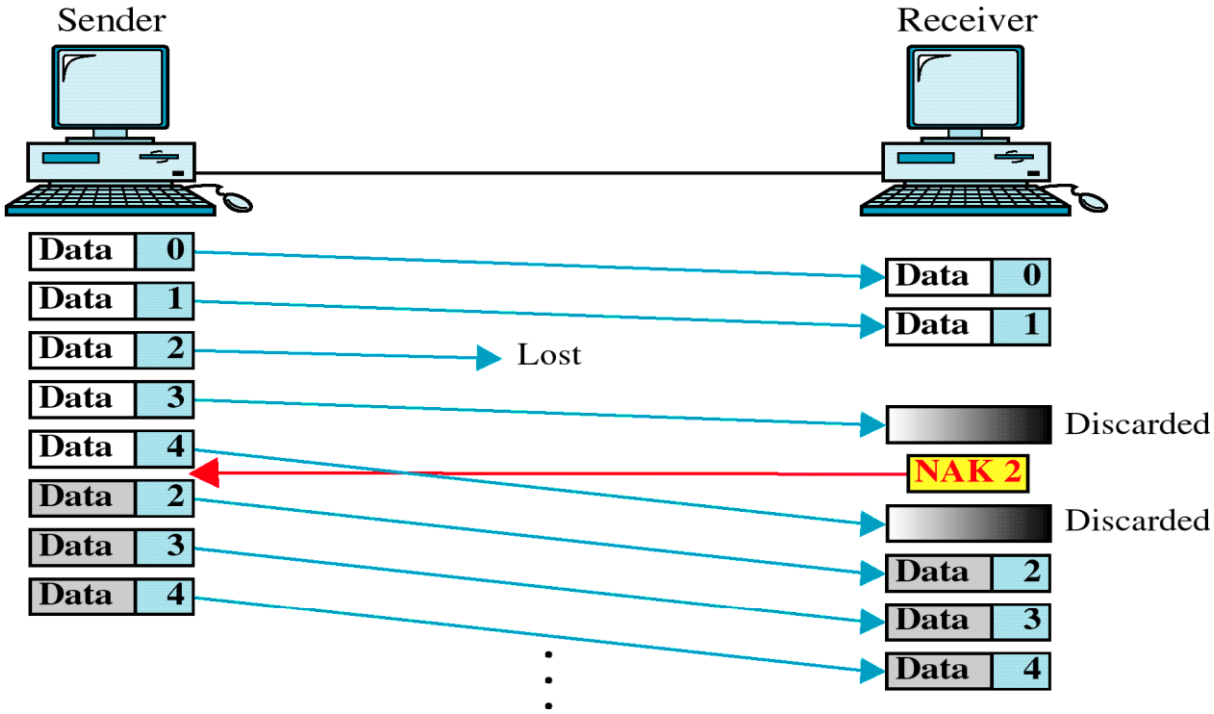
Go_Back_n:

در حالت اول، فریم ارسالی خراب است .طبق شکل فریم اول، دوم و سوم ارسال می شود و از طرف گیرنده هم ۳ ACK ارسال می گردد .فریم شماره ۳ ارسال می شود ولی خراب است و گیرنده ۳ NAK راارسال می کند .اما ۳ NAK زمانی به فرستنده می رسد که فریم های ۴ و ۵ هم ارسال شده است .روش Go_Back_nمی گوید برگرد به آخرین شماره ای که درست ارسال کرده ای، بافر را خالی و پنجره را اصلاح کن و از شماره ۳ شروع کن به ارسال کردن و به گیرنده هم می گوید که بافر را خالی و پنجره را اصلاح کن حتی فریم هایی که درست ارسال شده اند) مثل ۴و ۵(

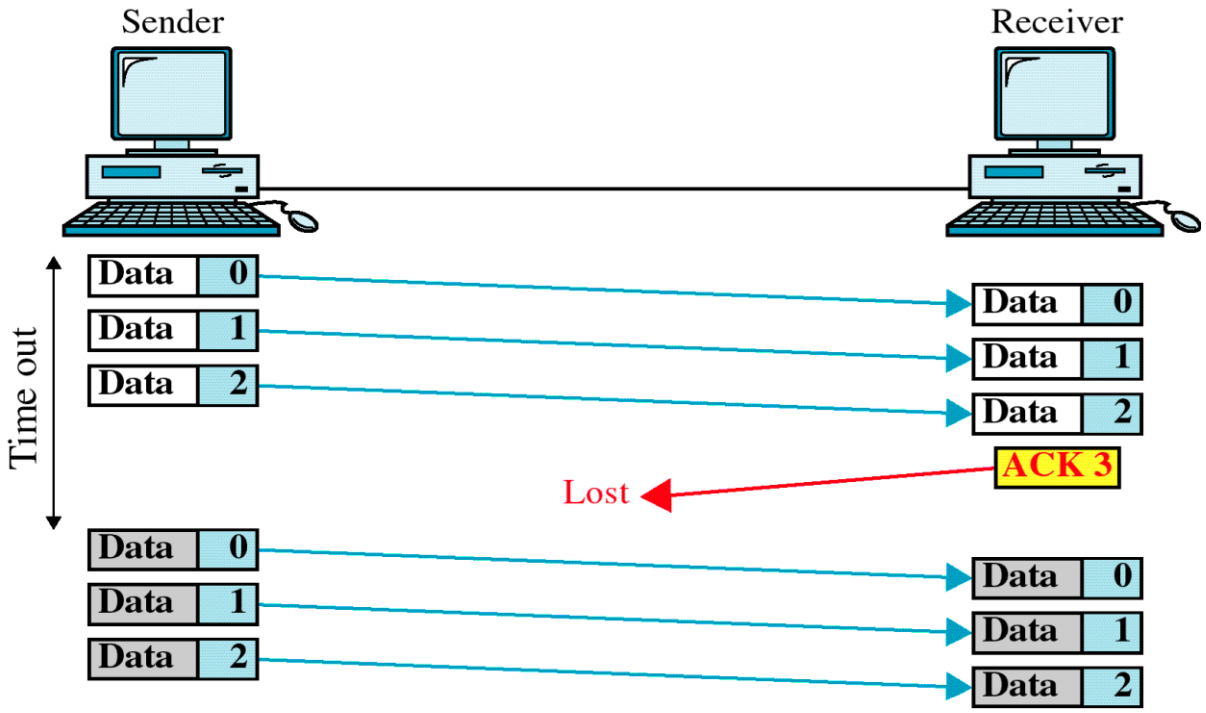


مشکل این روش این است که فریم های ۴ و ۵ درست دریافت شده اند ولی دور ریخته می شوند پس Over Headبالا می رود و ویژگی این روش این است که ترتیب فریم ها در گیرنده رعایت شده است . چون ترتیب رعایت می شود و می توان با دریافت اولین فریم آن را به لایه بالا تر ارسال کرد و به همین ترتیب دومی و سومی .پس می توانیم فقط یک بافر داشته باشیم و دیگر نیازی نیست که فریم ها را نگه داریم و وقتی کامل شد به لایه بالاتر بفرستیم .اگر بر مبنای این منطق نبود به مشکل بر می خوردیم .چون باید همه فریم ها را نگه می داشتیم، آن ها را مرتب می کردیم و بعد به لایه بالاتر تحویل می دادیم .پس این نکته از ویژگی های مهم این روش است.

حالت بعدی فریم Lost می شود .فریم شماره صفر و یک ارسال و دریافت می شود .فریم شماره ۲ ارسال و گم می شود و به گیرنده نمی رسد بعد بلافاصله فریم های ۳و ۴ را ارسال می کند و وقتی می خواهد فریم بعدی را ارسال کند ۲ NAK می آید و گیرنده در این حالت فریم های شماره ۳ و ۴ را Discard می کند و دوباره از شماره ۲ شروع به ارسال می کند.



در حالت بعدی، ACK گم می شود .فریم های شماره صفر و ۱ و ۲ ارسال و دریافت می شـود ۳ ACK . نیز ارسال و گم می شود Timer .پی از مدتی Time Out می دهد و دوباره فریم های شماره صفر و یک و دو را ارسال می کند .گیرنده با فریم های تکراری مواجه شده و آن ها را Discard میکند.



روشSelective Reject

Error Detection

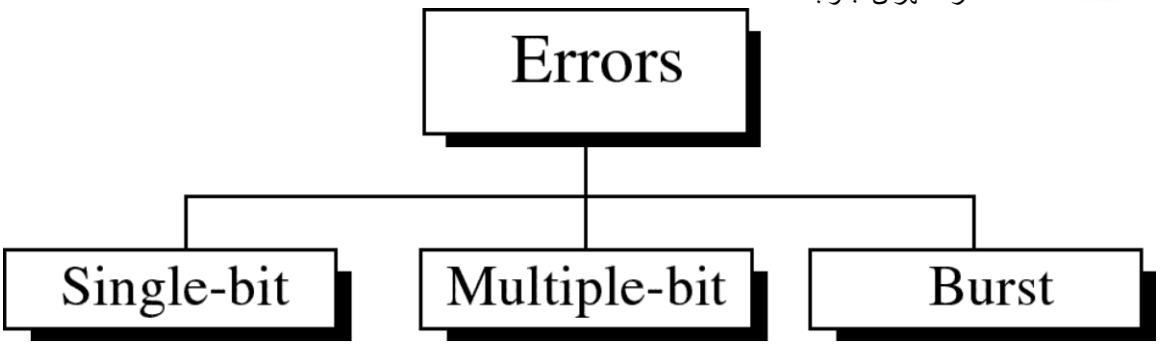
انواع خطا ها(Error)

امکان رخداد خطا در این ۳ قالب وجود دارد.

۱. Single_bit Error

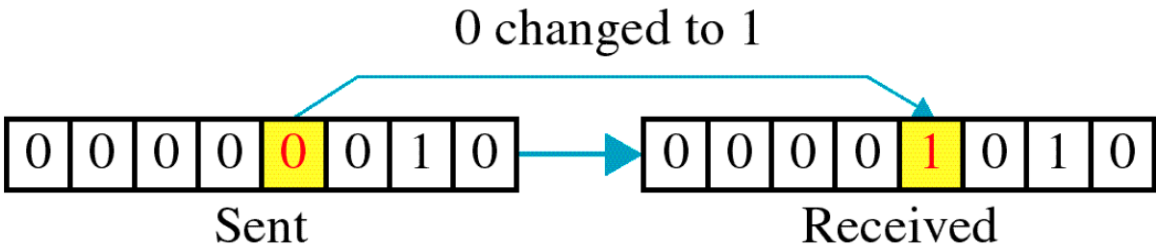
۲. Multiple_bit Error

۳. Brust Error



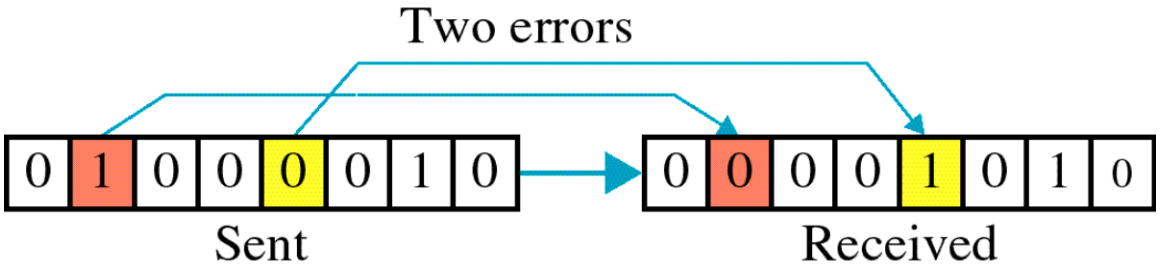
Single_bit Error

اطلاعات از طرف فرستنده ارسال و در گیرنده دریافت می شود و در یک بیت تغییر دیده می شود که به این خطا، خطای تک بیتی گفته می شود.



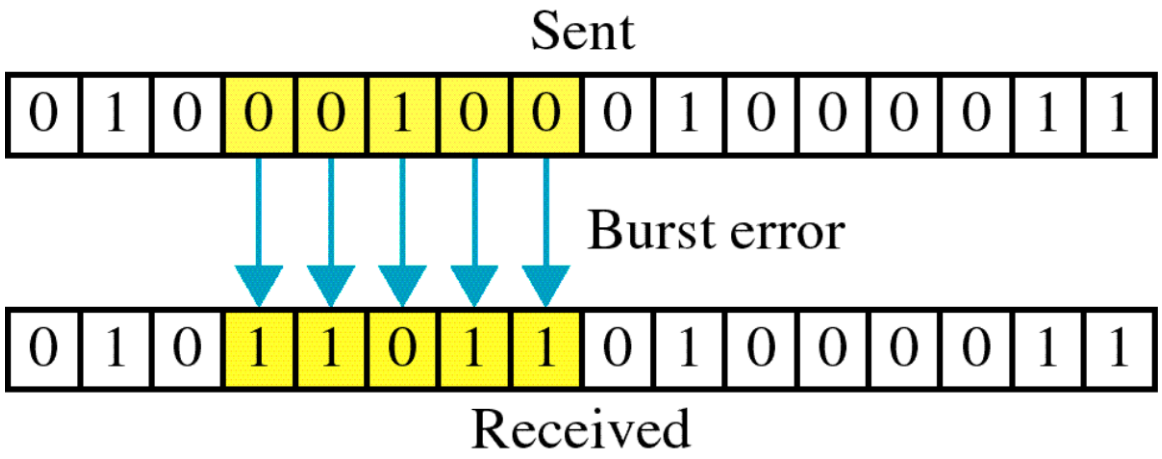
Multiple_bit Error

در این حالت، دو یا سه چند خطا دیده می شود که خطای چند بیتی نام دارد.



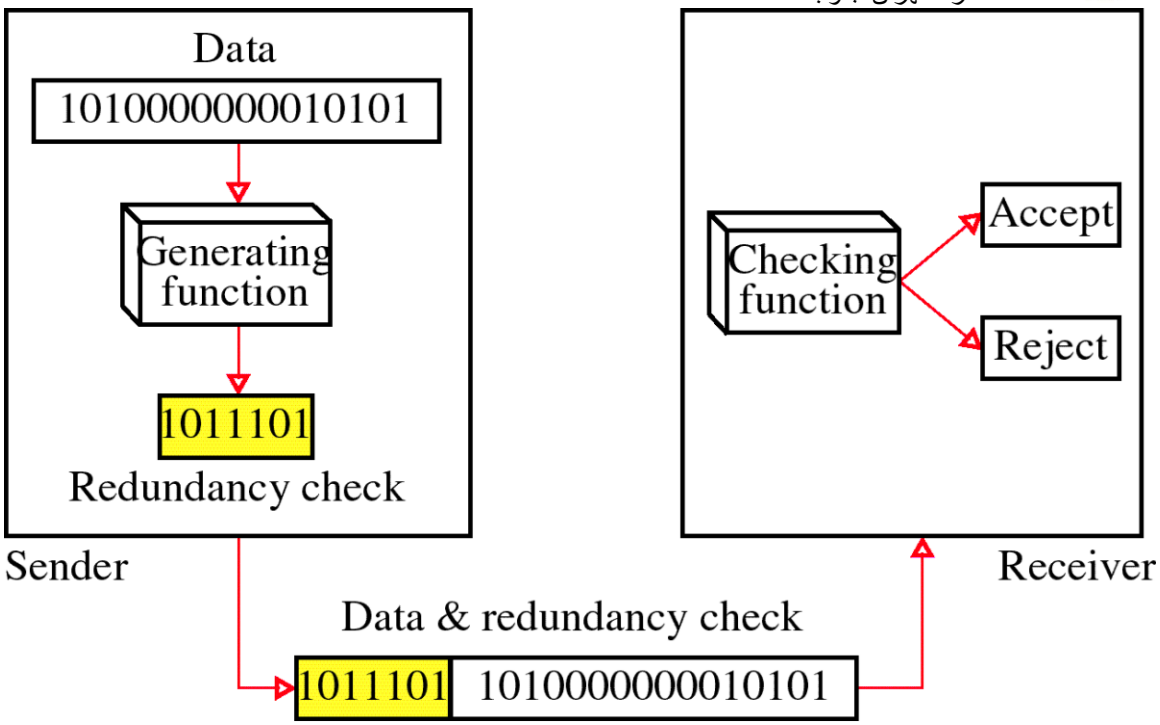
Brust Error

در این حالت، خطا هایی پشت سر هم اتفاق می افتند .مثلا در اثر رعد و برق این رخ داد صورت می گیرد و موجب می شود که مثلا ۵۰ بیت تغییر پیدا کند که به این خطا، خطای انبوه گفته می شود.



Redundancy (افزونگی)

بدون داشتن اطلاعاتی در مورد داده فرستاده شده نمی توان خطای آن را تشخیص داد .بنابراین برای کشف خطا، بایستی یک سری اطلاعات اضافی داشته باشیم تا بتوانیم خطا را تشخیص دهیم .به همین دلیل یک سرس اطلاعات کنترلی به همراه داده ارسال می گردد، تا گیرنده بفهمد که داده ای که دریافت کرده، صحیح بوده یا دارای خطا می باشد .ایت اطلاعات کنترلی بر اساس Function طراحی می گردد که در گیرنه همان Function در نظر گرفته شده است و پس از گرفتن داده با استفاده از Function مورد نظر خطا را تشخیص می دهد.



Type Of Detectionروش های کشف خطا

۱) Parity Check .بیت توازن

۲) Parity Check (Row_Column) .بیت توازن سطری ـ ستونی

۳)CRC (Cyclie Redmndancy Check) .

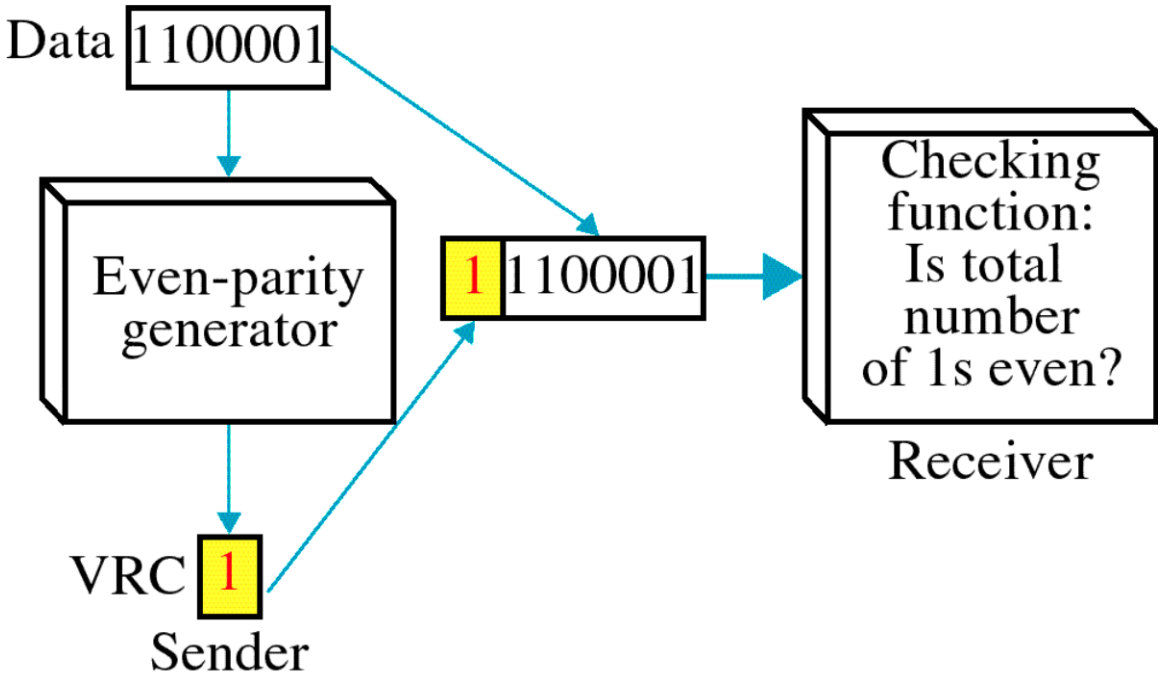
۴)Check Sum .

Parity Check

بیت توازن از خواصیت XOR استفاده می کند .XOR بیت های زوج، صفر و XOR بیت های فرد، یک می باشد و یا از XNOR استفاده می شود که بالعکس XOR است.

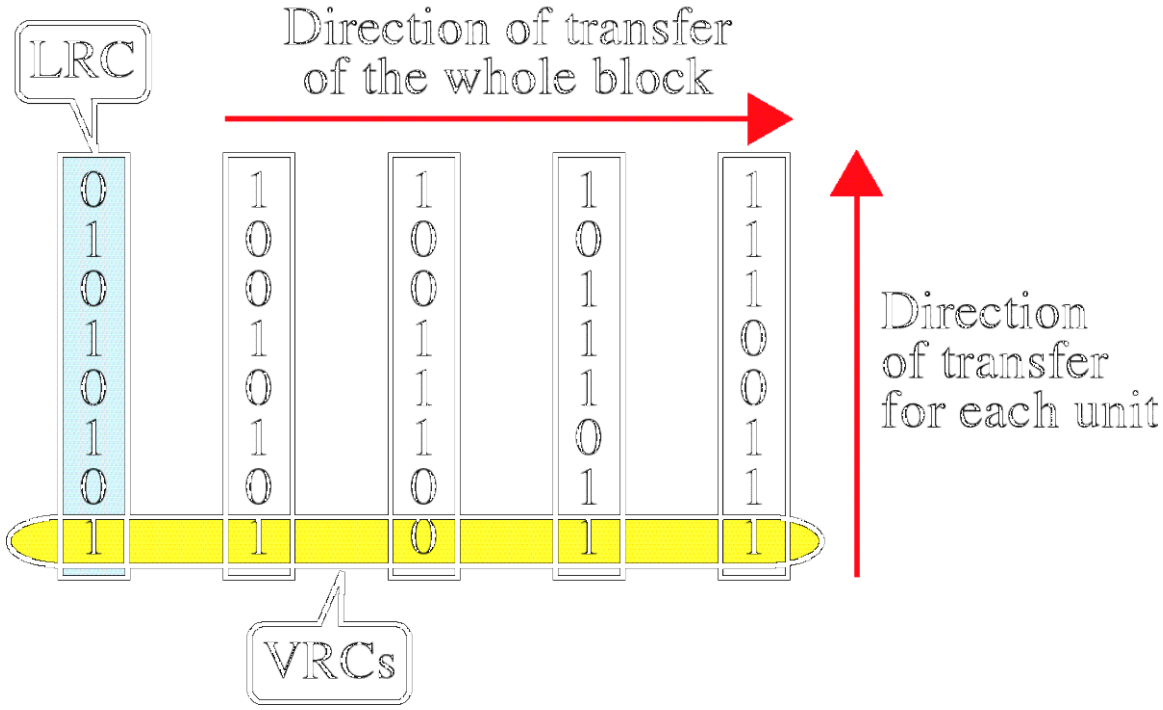
Functionدر این حالت بر مبنای استفاده از Parity زوج و یا فرد است که از این ویژگی برای کشف خطا استفاده می گردد .داده به گیرنده ارسال می شود Function فرستنده و گیرنده یکسان است، بـر اسـاس Parityزوج و فرد چک می کند XOR).می کند (و خطا را در صورت وجود تشخیص می دهد .این روش

فقط خطای تک بیتی را تشخیص می دهد و اگر بیش از یک خطا داشته باشیم، با استفاده از این روش امکان تشخیص آن را نداریم .همچنین این روش از نظر پیاده سازی بسیار ساده است.



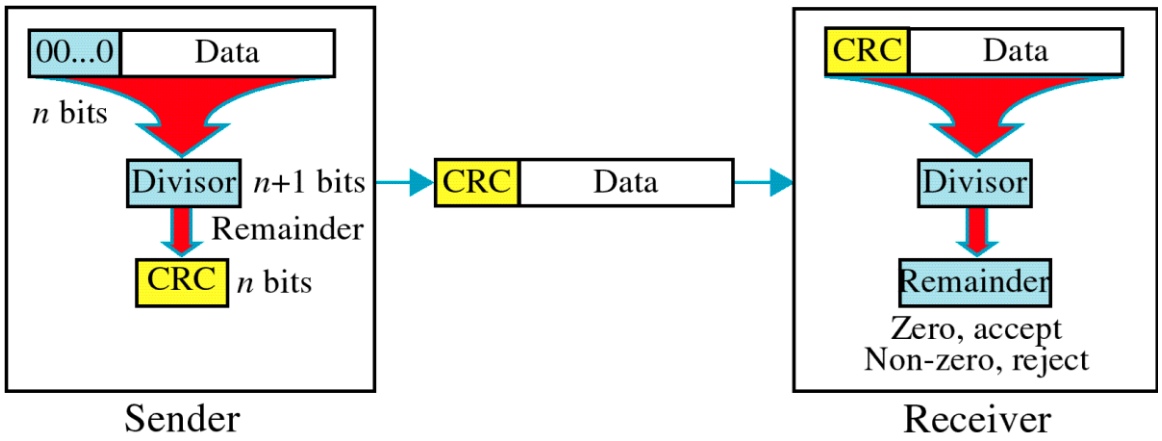
Parity Check (Row_Column)

می خواهیم روشمان را ارتقا دهیم تا خطای بیشتری کسف کنیم .ماتریسی از اطلاعات را قرار می دهیم و روی سطر و ستون Parity می زنیم .اگر تعداد یک ها زوج بود، بیت Parity صفر و اگر تعداد یک ها فرد بود یک خواهد بود .پس پس جریان اطلاعات را می گیریم و در قالب یک ماتریس n X n قرار می دهیم و روی سطر و ستون آن Parity میزنیم .این روش خطا های یک بیتی و دو بیتی را صد ردصد تشخیص می دهد .خطا های سه بیتی را هم تشخیص می دهد ولی خطا های چهار بیتی را نمی تواند تشخیص دهد.



روش CRC

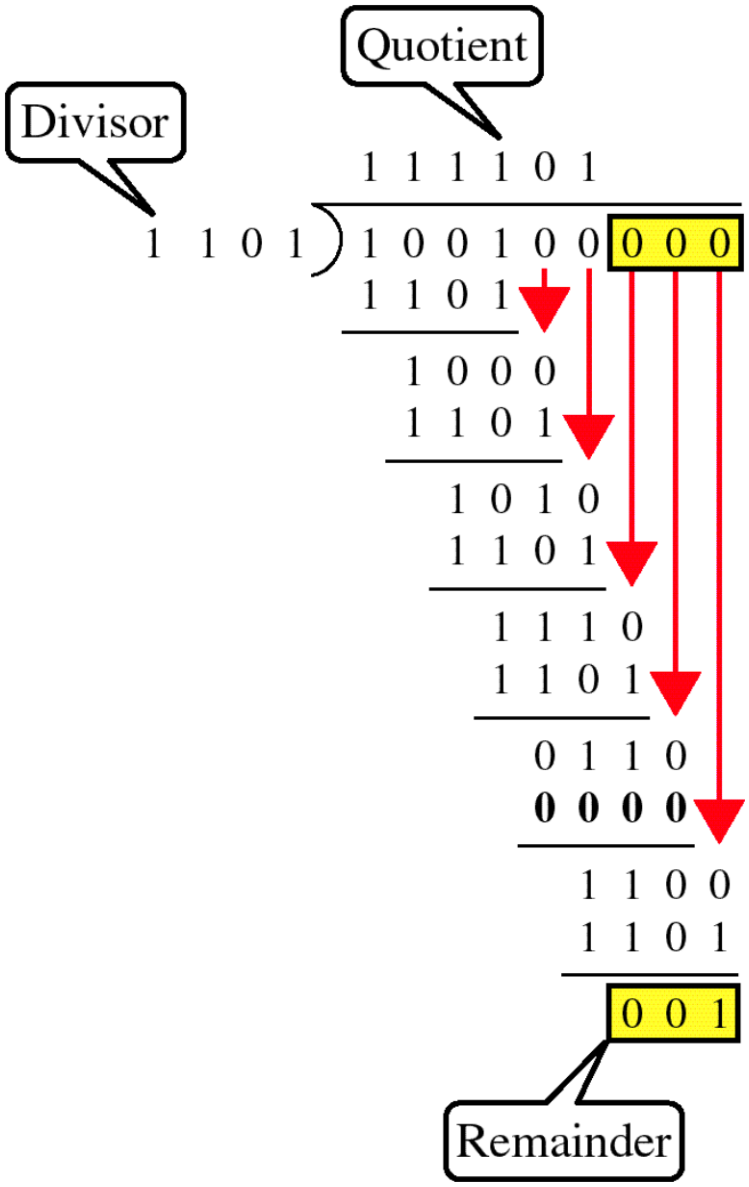
این روش، بسیار پرکاربرد و معروف است و منطق آن بر مبنای این ایده است که فرستنده و گیرنده یک الگوی بیتی دارند که بینشان مشترک است و اصطلاحا به آن Divisor یا مولد گفته می شود . Divisor ها حالت های خاصی دارند و استاندارد های بین المللی هستند.



اگر Divisor فرستنده و گیرنده n + ۱ بیتی باشد، کافی است به تعداد n - ۱ بیت یعنی n بیت اطلاعات را شیفت دهیم و n بیت صفر جلوی Data قرار دهیم و نتیجه را بر Divisor تقسیم بندی کنیم و باقی مانده که حاصل می شود، CRC را به ما می دهد که این بیت های CRC بیت های افزون هستند که گیرنده بر اساس آن باید خطا را تشخیص دهد .گیرنده داده را دریافت می کند و چون Divisor بین

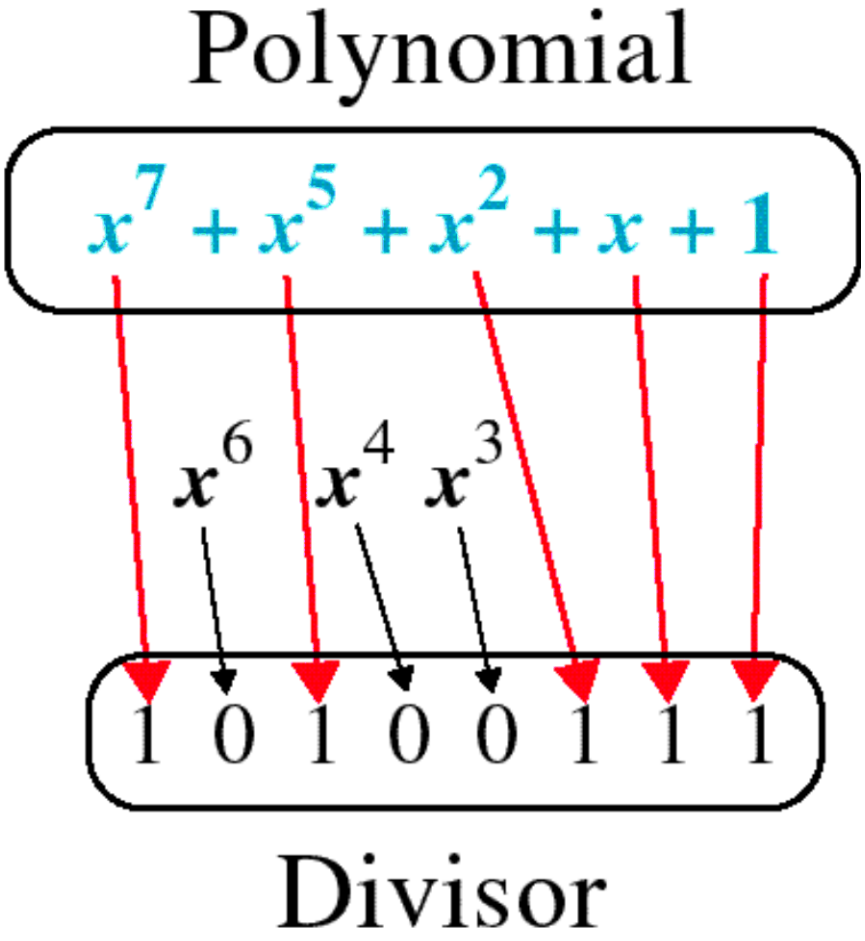
فرستندخ و گیرنده یکی است، کافیست داده و CRC آن را بر Divisor تقسیم کنیم، اگر صفر شد یعنی خطا اتفاق نیفتاده و اطلاعات را می پذیرد و اگر یک شد، یعنی خطا اتفاق افتاده و اطلاعات را Reject می کند.

مثال :طبق این شکل Divisor ، چهار بیتی است، هر سه بیت شیفت می دهیم و بـر Divisor تقسـیم Binaryمی کنیم و باقیمانده CRC را نشان می دهد که Data اضافه می گردد تا بتوان به وسیله آن در گیرنده خطا را تشخیص داد.



این روش، روش تقسیم باینری بود .روش دیگری هم وجود دارد که اطلاعات را به Mode چند جملـه ای می برد و تقسیم چند جمله ای انجام می دهد که عمل راحت تری است.

که ازو x^2 و x^1 و x تا x^n وجود دارد .که در این روش ماژول های هم نام حذف مـی شـوند .یعنـی x^7با x^7 حذف می شود، به مولد ها اصطلاحا CRC می گویند کـه CRC-۱۲ ، CRC-۱۶ ، CRC-ITU، CRC-۳۲، CRC های استاندارد بین المللی هستند، یعنی همان الگو های مشترک بین فرستنده و گیرنده.



وقتی که پروتکل تعریف می کنیم پروتوکل حتما Data Link را دارد که باید در آن مشخص کنیم از چه مکانیزم تشخیص خطایی استفاده می کنیم .اگر از CRC استفاده می کنیم باید این را هم مشخص کنـیم که از کدام CRC استفاده می کنیم (CRC-۱۲ ، CRC-۱۶ یا(...

CRC-12

$x^{12}+x^{11}+x^{3}+x+1$

CRC-16

$x^{16}+x^{15}+x^{2}+1$

CRC-ITU

$x^{16}+x^{12}+x^{5}+1$

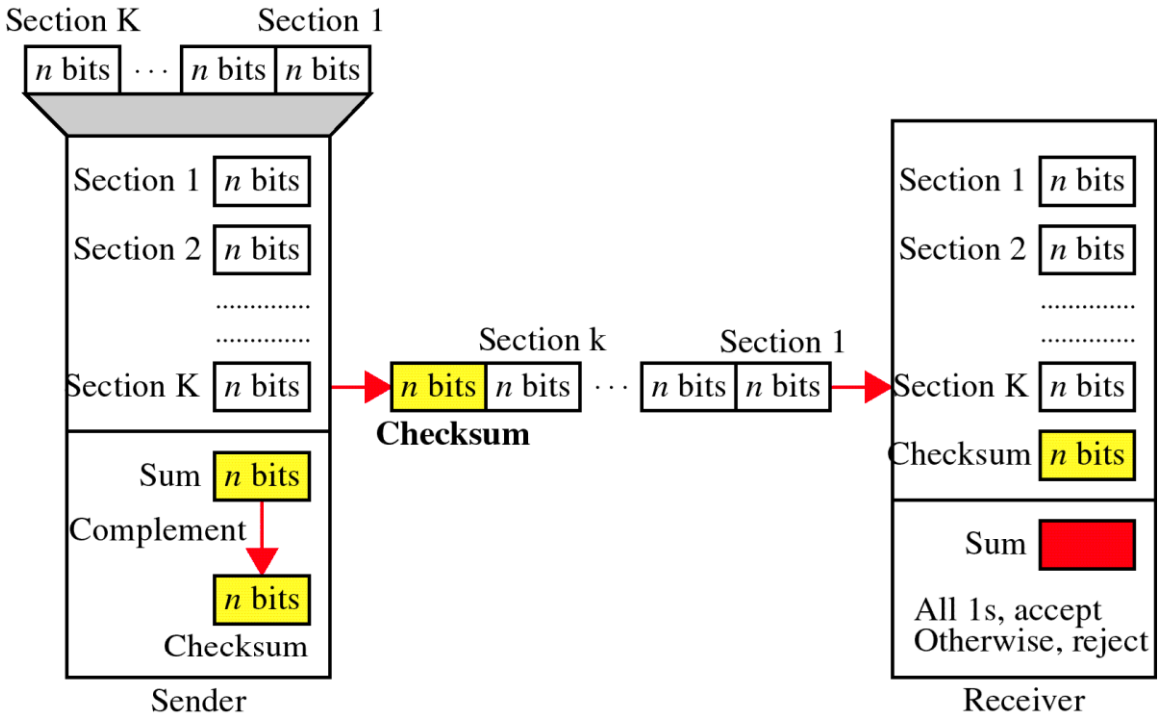
CRC-32

$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

روش CRC تمامی خطا های Single_bit را تشخیص می دهد .خطا های زوج) دو بیتی (را بـه شـرطی که سه تا یک (x) داشته باشد تشخیص می دهد و همه خطاهای فرد را تشخیص می دهد به شرطی که x ۱ +را داشته باشد .خطاهای Burst را نیز به تعداد طول CRC یا کمتـر از طـول CRC تشـخیص مـی دهد .به عنوان مثال CRC-۱۲ ، ۱۳ بیت است .پس خطای Burst را به اندازه ۱۳ بیت یا کمتر تشخیص می دهد و یا CRC_۳۲، ۳۳ بیت است .پس خطاهای کوچکتر مساوی ۳۳ بیت را تشخیص می دهد.

روش Check Sum

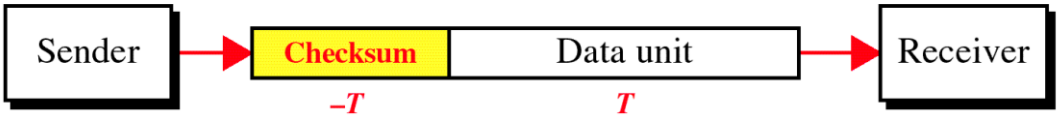
این روش کاربرد زیادی دارد .در این روش اطلاعات را به k بخش n بیتی تقسیم می کنیم .مثلا اگر Data صد بیتی باشد، به ده تا section ده بیتی تقسیم می کنیم و یا...



که این section ها را با هم جمع می کنیم و نتیجه آن را Complemat می کنیم و Check Sum بدست می آید که آن را به عنوان بیت افرونه به ابتدای اطلاعات اضافه کرده و برای گیزنده ارسال می کنیم . گیرنده عمل عکس را انجام می دهد Section .های یک k را نوشته و با Check Sum جمع می کند و اگر نتیجه ffff شد، خطا اتفاق نیافتاده در غیر این صورت خطا صورت گرفته است چون جمع هر عدد با notخود ۱۱۱۱ یا ffff می شود .به عنوان مثال، فرض می کنیمData ی ما ۲۴ بیتی است و می خواهیم از روش FCS برای کشف خطا استفاده کنیم .۲۴ بیت را به چهار تا Section ، ۶ بیتی تقسیم می کنیم:

طبق شکل زیر، اگر Data را T در نظر بگیریم، Check Sum آن T~ خواهد بود که به عنوان اطلاعات کنترلی به همراه Data به گیرنده ارسال می شود .گیرنده وقتی که T~ و T را با هم جمع می کند باید به FFFFبرسد تا بدون خطا باشد .این روش، خطا های Single را تشخیص می دهد ولی خطا های انبوه را نمی تواند تشخیص دهد.

The receiver adds the data unit and the checksum field. If the result is all 1s, the data unit is accepted; otherwise it is discarded.



Data link portocals

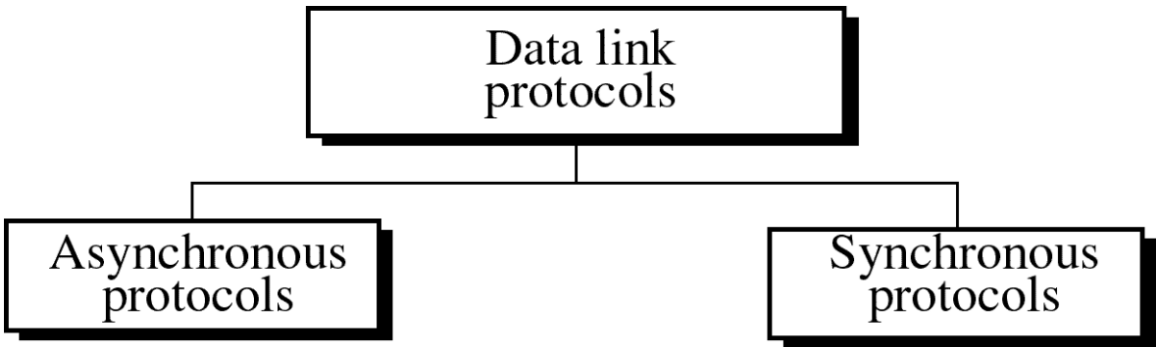
در لایه Data Link پروتکل های مختلفی وجود دارد که هر یک به طور جداگانه Line Disciplin ، Flow Contو Error Error را دارا هستند .این پروتکل ها عبارتند از:

۱. Asynchronous Protocols .

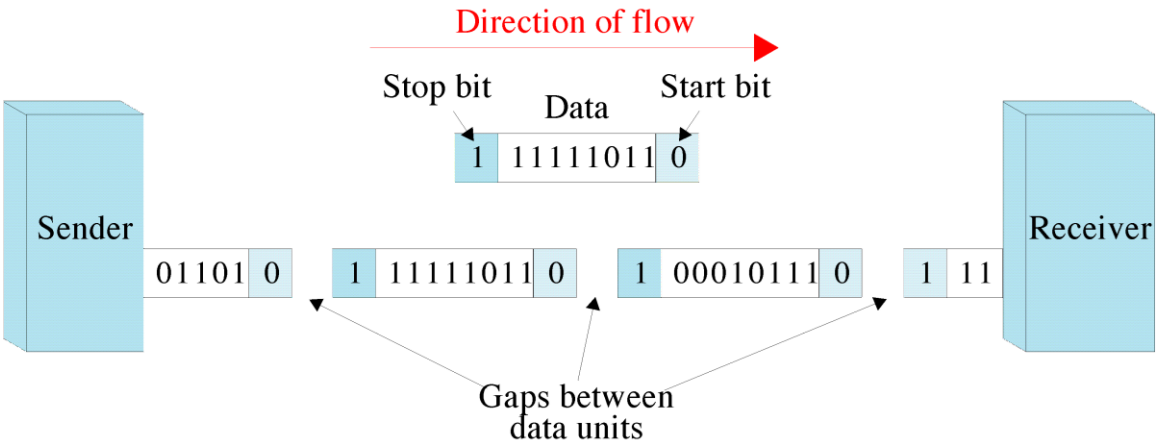
۲. Synchronous Protocols .

۳. Charaster-Oriented Protocols .

۴. Bit-Oriented Protocols .



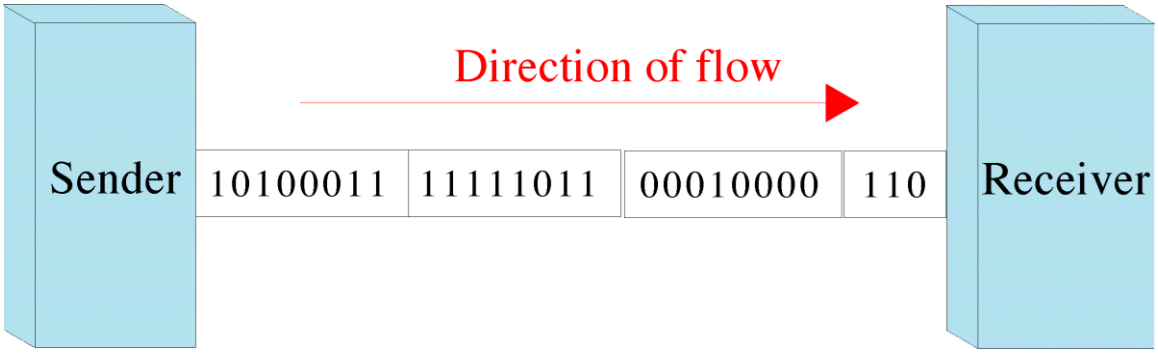
انتقال اطلاعات آسنکرون



مشکل صفر های طولانی و یک های طولانی را می توان با استفاده از این روش حل کرد بدین صورت که اگر ما اطلاعات را به صورت کاراکتر برداریم و ۸بیتی در نظر بگیریم و یک Start bit و Stop bit و یک بیت به عنوان بیت Parity در نظر بگیریم این مشکل حل خواهد شد در انتقال اطلاعات آسنکرون، Start bitیک بیت صفر و Stop bit یک بیت یک می باشد پس اگر یک های طولانی داشته باشیم با Stop bit و Start bit مشکل بر طرف می شود .عموما یک بیت Parity نیز در این روش وجود دارد همچنین بعد از کاراکتر های gap زمانی وجود دارد وقتی که خط بیکار است پشت سر هم یک ارسال می شود .یک در NRZIیعنی سیگنال منفی .وقتی که پشت سر هم یک ارسال می شود به محض اینکه صفر ارسال شد

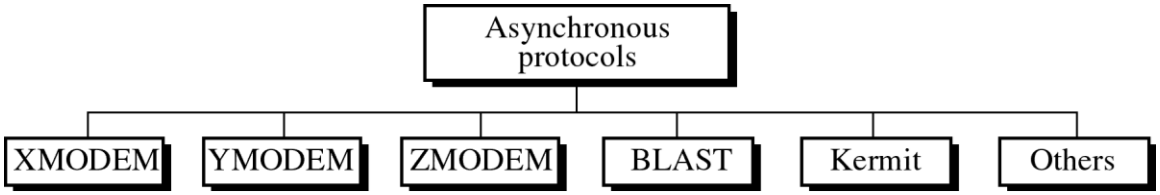
یعنی Start bit است و کاراکتر ها را می گیرد .مشکل این روش این است که Over Head خیلی بالایی دارد .چیزی حدود ٪۳۰ Over Head دارد .پس روش آسنکرون برای ارسال اطلاعات با حجم زیاد استفاده نمی شود .پس با برخورد با مشکل dc می تواند از طریق این روش آن را حل نمود.

انتقال اطلاعات سنکرون

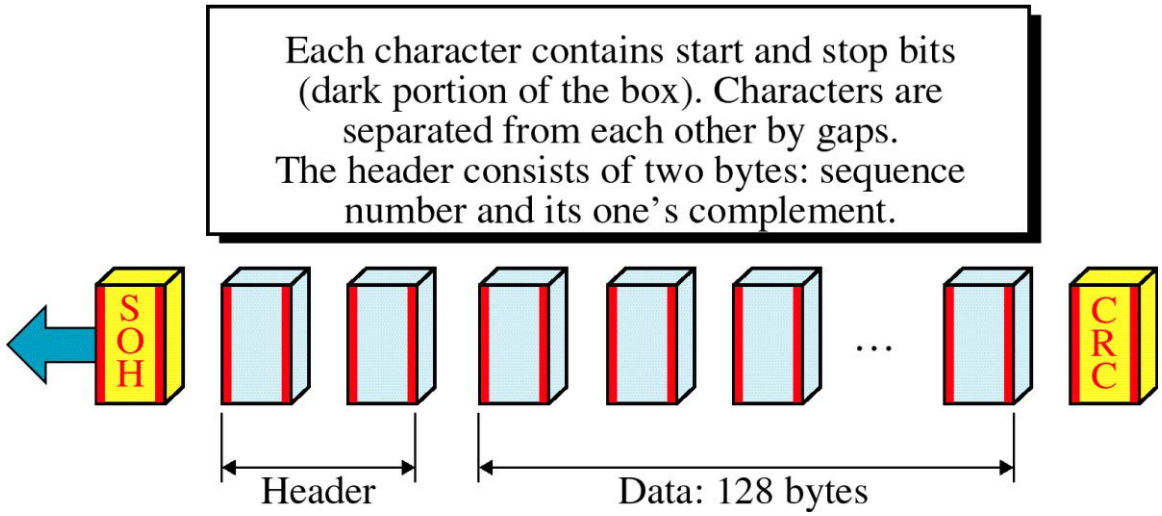


در انتقال اطلاعات سنکرون، اطلاعات پشت سر هم ارسال می شوند و برای اینکه مشکل صفر ها و یک های طولانی حل شود، از کدینگ های متفاوتی استفاده می شود .مثلا) Manchester Coding همچنین برای رفع مشکل (dc انتقال اطلاعات آسنکرون پیاده سازی بسیار ساده ای دارد و انتقال اطلاعات سنکرون گرانتر است .بیشتر مردم ما امروز از انتقال اطلاعات آسنکرون استفاده می کنند چون ارزانتر است .ولی اگر حجم اطلاعات زیاد باشد نمی توان از سنکرون استفاده کرد و باید از انتقال اطلاعات سنکرون استفاده کنیم.

پروتکل های آسنکرون



پروتکل XMODEM



پروتکل XMODEM پروتکلی دانشگاهی بوده است که برای انتقال اطلاعات text روی خطوط تلفنی مطرح شد Error Control : Stop & Wait ARQ Flow Control : Stop & Wait .انتقال اطلاعات آن یکطرفه است (Half Duplex) .و حجم واحد های اطلاعاتی آن ۱۲۸ بایت است.

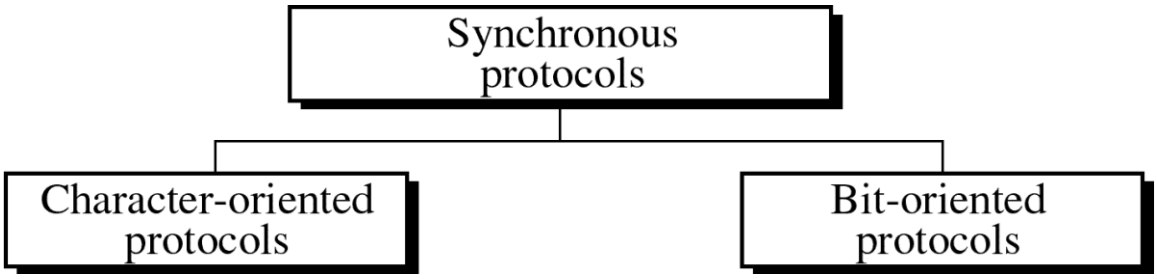
پروتکل YMODEM

حجم اطلاعات را از ۱۲۸Byte به ۱۰۲۴Byte انتقال داده است .چک خطا های دقیق تر را انجام می دهد. ارسال چندین فایل همزمان را نیز ساپورت می کند.

در پروتکل YMODEM ، هر کاراکتر یک start و یک stop دارد و کاراکتر ها با gap از یکدیگر جدا شده اند .انتقال اطلاعات آن آسنکرون است .کل فریم در شکل نشان داده شده است با فرمت CRC در

سنکرون هم به همین شکل است یک Header دارد و سپس ۱۲۸B ارسال کرده و بعد فورمت .CRC در اینجا بین کاراکتر ها گپ زمانی داریم ولی در سنکرون گپ زمانی وجود ندارد.

پروتکل های سنکرون



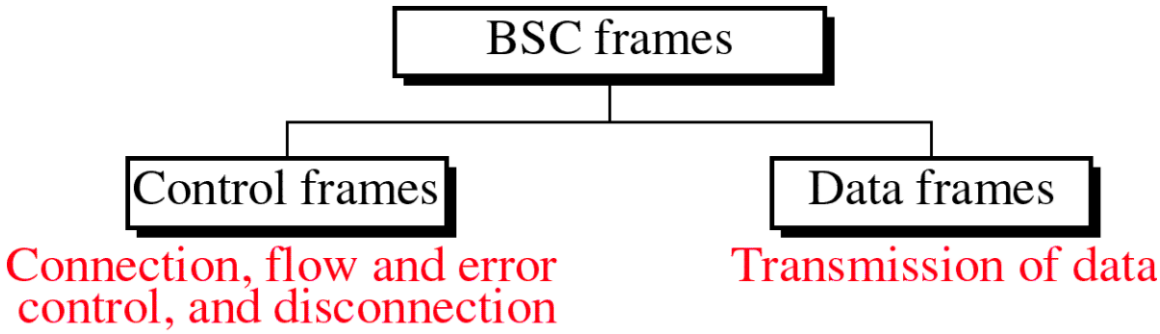
پروتکل های Character-Oriented

پروتکل های Character-Oriented دیدشان کاراکتری است پس برای انتقال اطلاعات متنی طراحی شده اند .پروتکل های Bit_Oriented ، دیدشان بیتی است پس کلیه فرمت ها را می توان انتقال دهند.

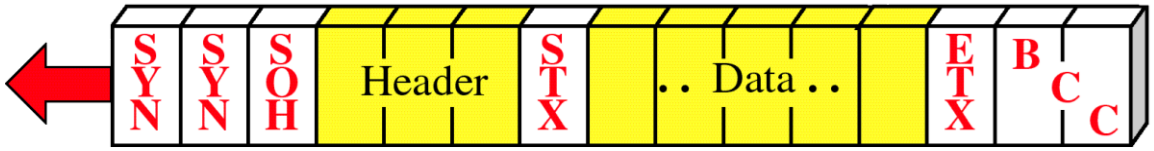
یکی از پروتکل های معروف سنکرون پروتکل BSC (Binary Synchronous Communication) می باشد.

پروتکل: BSC

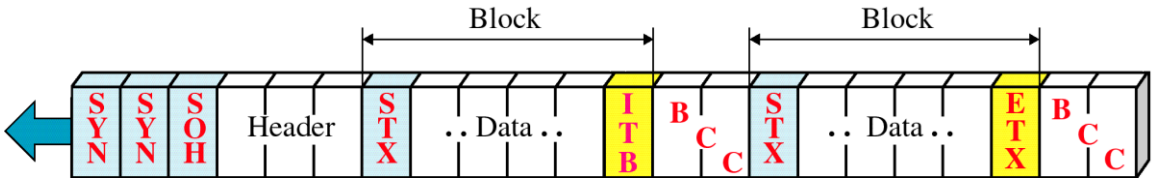
پروتکلی است که توسط IBM طراحی شده و در حال حاضر کمتر مورد استفاده قرار می گیرد .انتقال اطلاعات آن به صورت Half Duplex است و از مکانیزم Stop & Wait برای Flow Control و Error Controlاستفاده می کند و برای انتقال اطلاعات متنی استفاده می شود .دارای دو نوع فریم است.



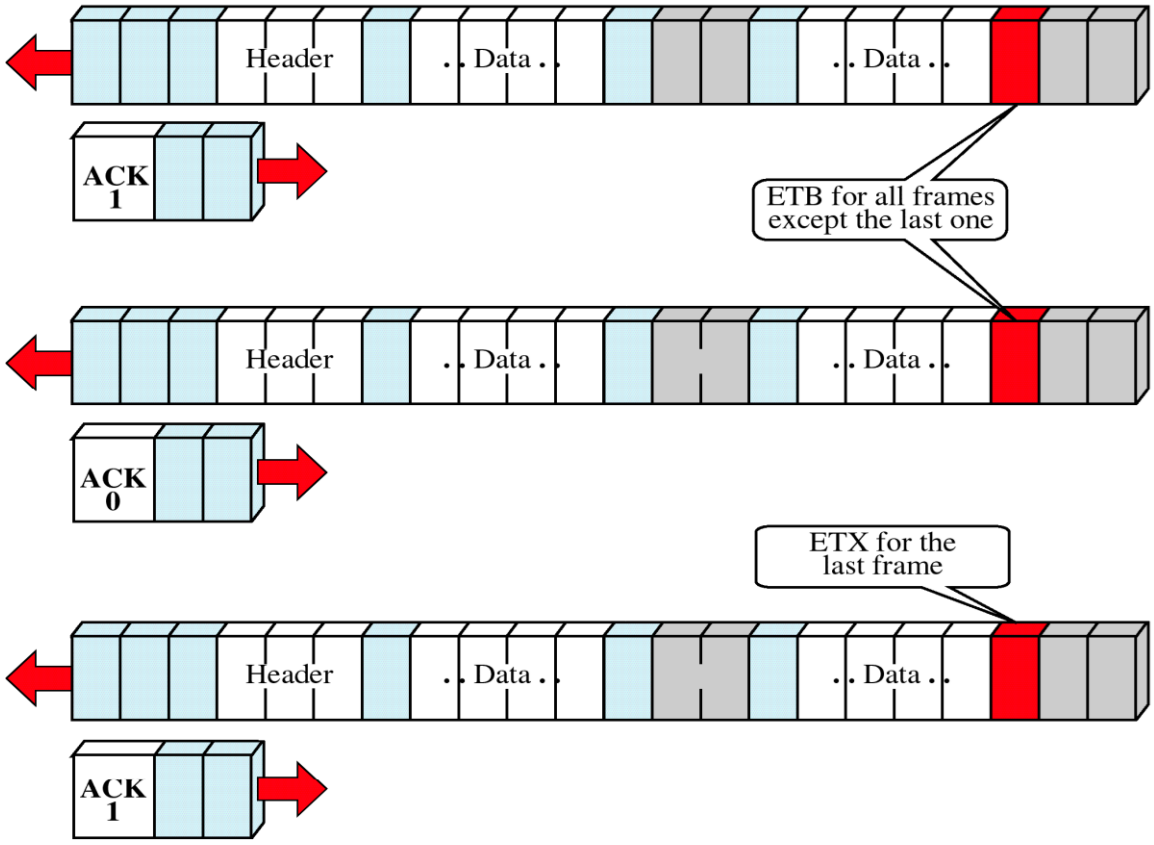
اگر از لایه Data Link به آن نگاه کنیم به صورت فریم است و دارای یک سری اطلاعات کنترلی و Data می باشد .دارای دو کاراکتر SYNC می باشد که برای همزمانی بین فرستنده و گیرنده است .این دید، دید کاراکتری است .سپس Header آن و بعد Start ، Data ، Stop و یک سری کاراکتر های کنترلی در آن ها قرار گرفته اند.



در شکل زیر دیده می شود که اطلاعات بلوک بندی شده اند که در واقع مانند قبل است ولی دارای بلوک های متفاوتی می باشد.

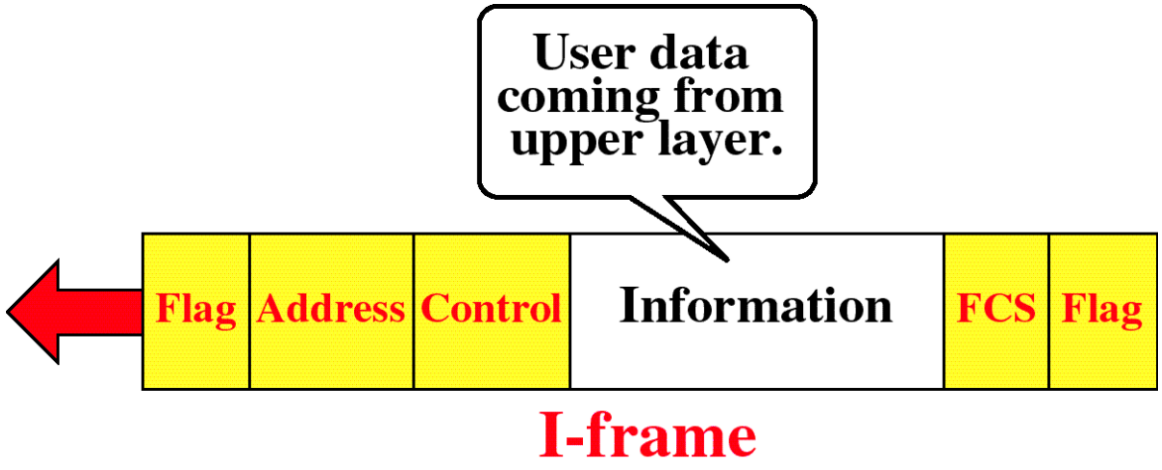


فرستنده فریم را در لایه Data Link می سازد و روی خط می گذارد .گیرنده دریافت می کند و چون مکانیزم Stop & Wait است گیرنده باید ACK بفرستد، ACK یک فرم فریم است که دارای دو کارکتر SYNCمی باشد و ACK همینطور که فریم ها فرستاده می شود، ACK آن ها هم از طرف گیرنده صادر می شود این دید، دید کاراکتری است که دیگر استفاده نمی گردد .از پروتکل های Bit_Oriented استفاده می گردد.



الگوی فریم کنترلی به این صورت است که یک فریم کنترلی دارای سه بایت است که دو بایت آن SYNC است و یک بایت برای ACK می باشد.

Information اطلاعاتی هستند که از لایه بالاتر گرفته شده اند .که کشف خطا مـی باشـد FCS . Flag انتهایی نیز پایان فریم را مشخص می کند.



LAN Technology

دلیل مطرح شدن شبکه های LAN در این قسمت معماری متفاوت این شبکه ها در لایه Physical و Data Linkمی باشد .تا اینجا لایه های فیزیکی و پیوند داده مورد بررسی قرار گرفتند و همانطور که گفته شد، این شبکه ها) شبکه های (LAN در این دو لایه دارای معماری متفاوتی می باشند.

از معماری های متفاوت شبکه LAN می توان به مورد زیر اشاره کرد که این معماری ها، ۴ معماری رایج در شبکه های LAN می باشند.

_ Project ۸۰۲

_ Ethernet (IEEE ۸۰۲,۳)

_ Token Bus (IEEE ۸۰۲,۴)

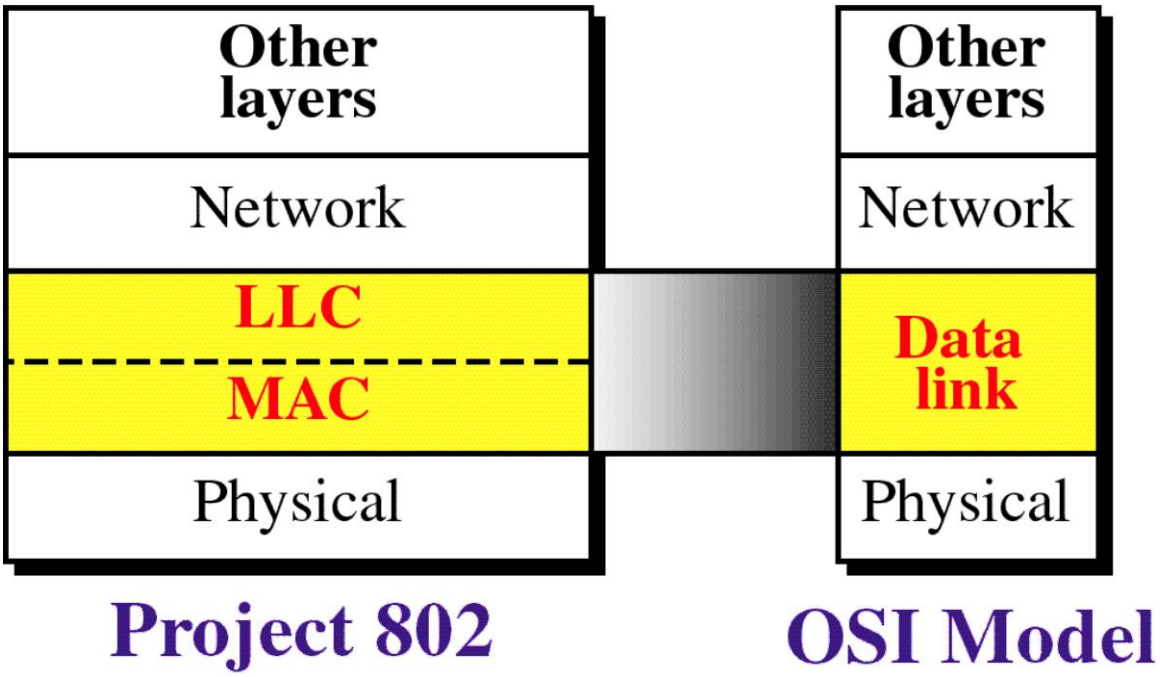
_ Token Ring (IEEE ۸۰۲,۵)

_ FDDI

Project ۸۰۲

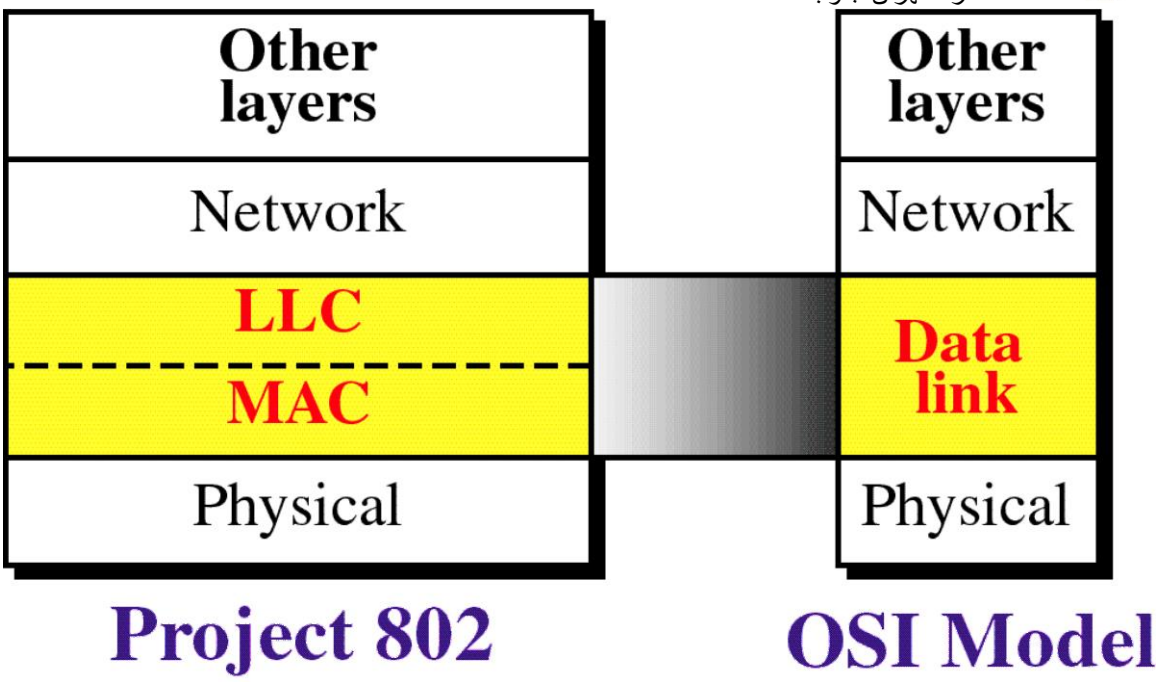
مشکلی که سازندگان شبکه های LAN داشتند، این بود که فریم گسترش سریعی که این شبکه ها پیدا کرده بودند، هیچ استاندارد معتبری نبود که شبکه ها با یکدیگر هماهنگ شوند به همین دلیل شرکت معتبر IEEE پروژه ای به نام ۸۰۲ تعریف کرد که سازندگان بر اساس استاندارد های این پروژه می توانستند تغییرات شبکه LAN را بسازند.

تفاوت Project ۸۰۲ مبدل: OSI



پروژه ۸۰۲ تنها در لایه Data Link با مدل OSI متفاوت است .در ۸۰۲ Project لایه Data Link به دو Sub Layer بـه نـام هـای LLC و MAC تبـدیل شـده اسـت .MAC مخفـف Media Access Controlو LLC مخفف Logical Link Control می باشد .بقیه لایه ها، مشابه لایه های مدل OSI می باشد.

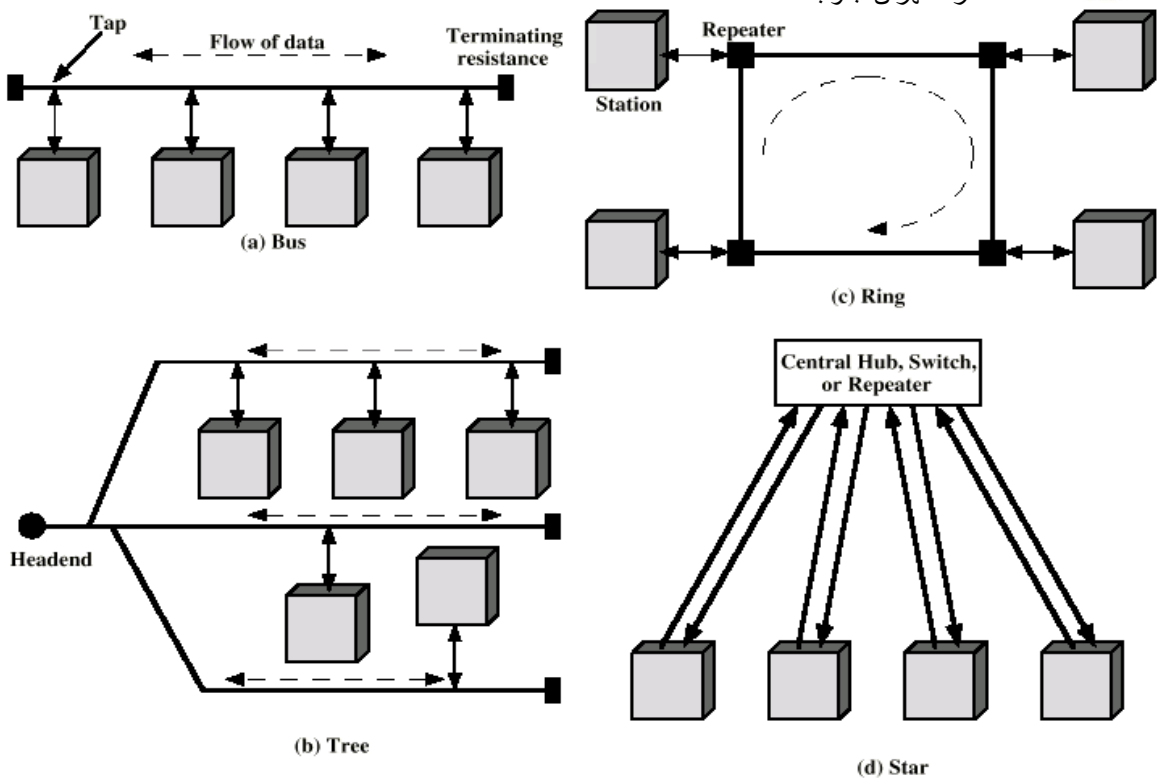
پروژه ۸۰۲,۱ اولین پروژه ای که برای ارتباطات بین شبکه ای طاحی شد) در لایه (Network سپس لایه Data Linkو لایه Physical تبدیل شد به پروژه ۸۰۲,۲ که همان LLC می باشـد و لایـه Physical تعریف پروژه های مختلف ۸۰۲.۳ ، ۸۰۲,۴ ، ۸۰۲.۵ و ... که به استاندارد های مختلف برای طراحی شبکه های LAN بر می گردد.



بخش Physical و قسمتی از Data Link شامل کشف خطا، Encoding و ... را انجام می دهد . Flow Controlبه LLC مربوط می شود .برای معماری های متفاوت شبکه ها LLC یکشان و مشترک داریم و همه شبکه های LAN با معماری های متفاوت از این Sub Layer طبعیت می کنند .ما در یک شبکه ممکن است پروتکل های متفاوتی از قبیل TCP/IP ، SPX/IPX ، Apple Talk و ... و پروتکل های متفاوتی که از لحاظ ساختار با هم متفاوتند استفاده کنیم که این پروتکل ها هیچ ارتباطی به لایه زیزین ندارند .بخشی از LLC بر می گردد به آدرس های پروتکل ها که به عنوان مثال پروتکل گیرنده TCP/IPو پروتکل فرستنده SPX/IPX است پس اطلاعات کنترلی برای تبادل اطلاعات بین پروتکل های متفاوت را دارد که LLC آن را حمل می کند.

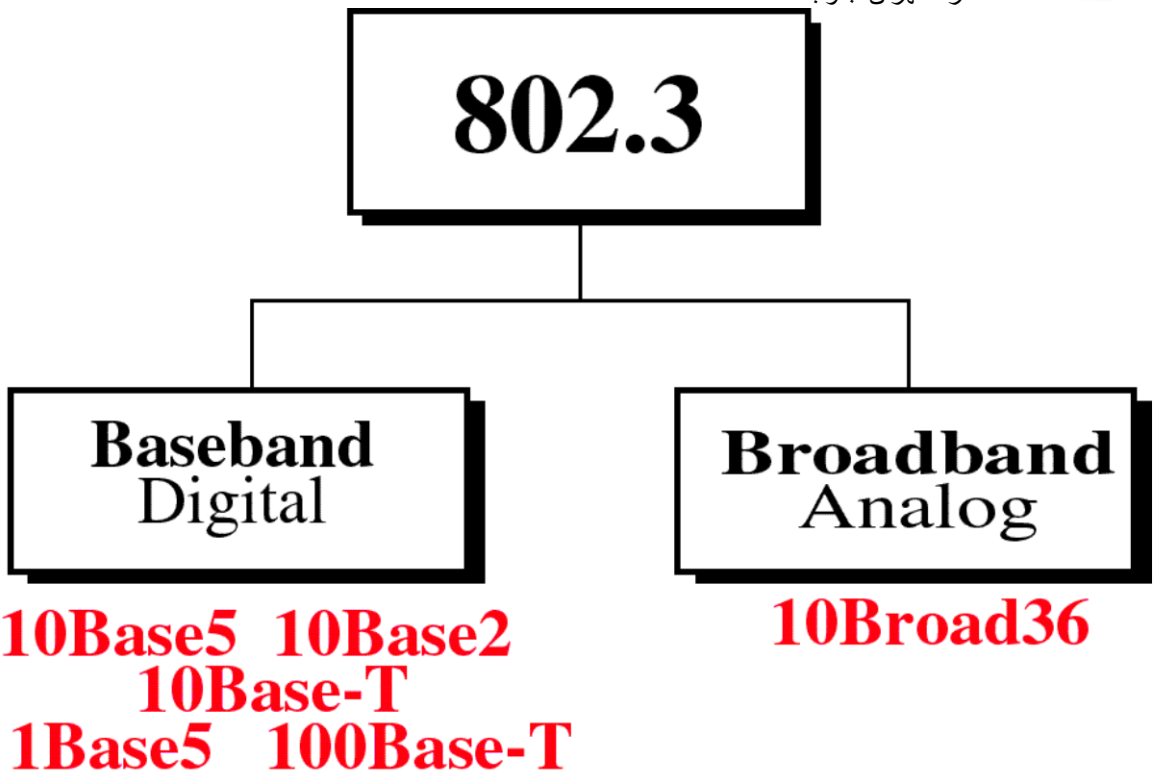
توپولوژی های مختلف شبکهLAN

توپولوژی های مختلفی در شبکه LAN وجود دارد که قبلا در مورد آن ها بحث شد .از قبیل توپولوژی Bus، Ring ، Star و . Tree شبکه های LAN با معماری های متفاوت می توانند از یکی از این توپولوژی ها استفاده کنند.



Ethernet (IEEE ۸۰۲,۳)

شبکه ای به نام شبکه اترنت توسط شرکت زیراکس بوجود آمد و روز به روز گسترش پیده کـرد و چـون از نظر کارایی و قیمت شبکه مناسبی بود و به دلیل رشد این شبکه، IEEE یک استانداردی بـه نـام ۸۰۲,۳ برای شبکه های اترنت طراحی کرد تا سازندگان بتوانند از این استاندارد ها استفاده کرده و تجهیزات مـورد نیاز را بسازند.

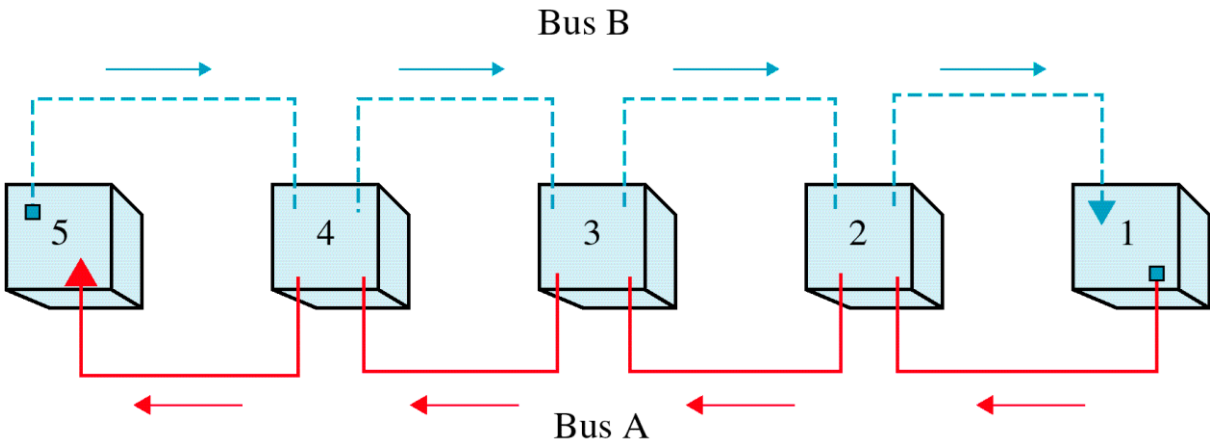


این شبکه) استاندارد (دارای دو نوع پیاده سازی می باشد Base Band :که پیاده سازی دیجیتالی است و Broad Bandکه پیاده سازی آنالوگ می باشد .که انواع متفاوتی از این پیاده سازی) که در شکل نوشته شده است (وجود دارد.

در حال حاضر پیاده سازی دیجیتالی Base Band است و از پیاده سازی آنالوگ استفاده نمی شود.

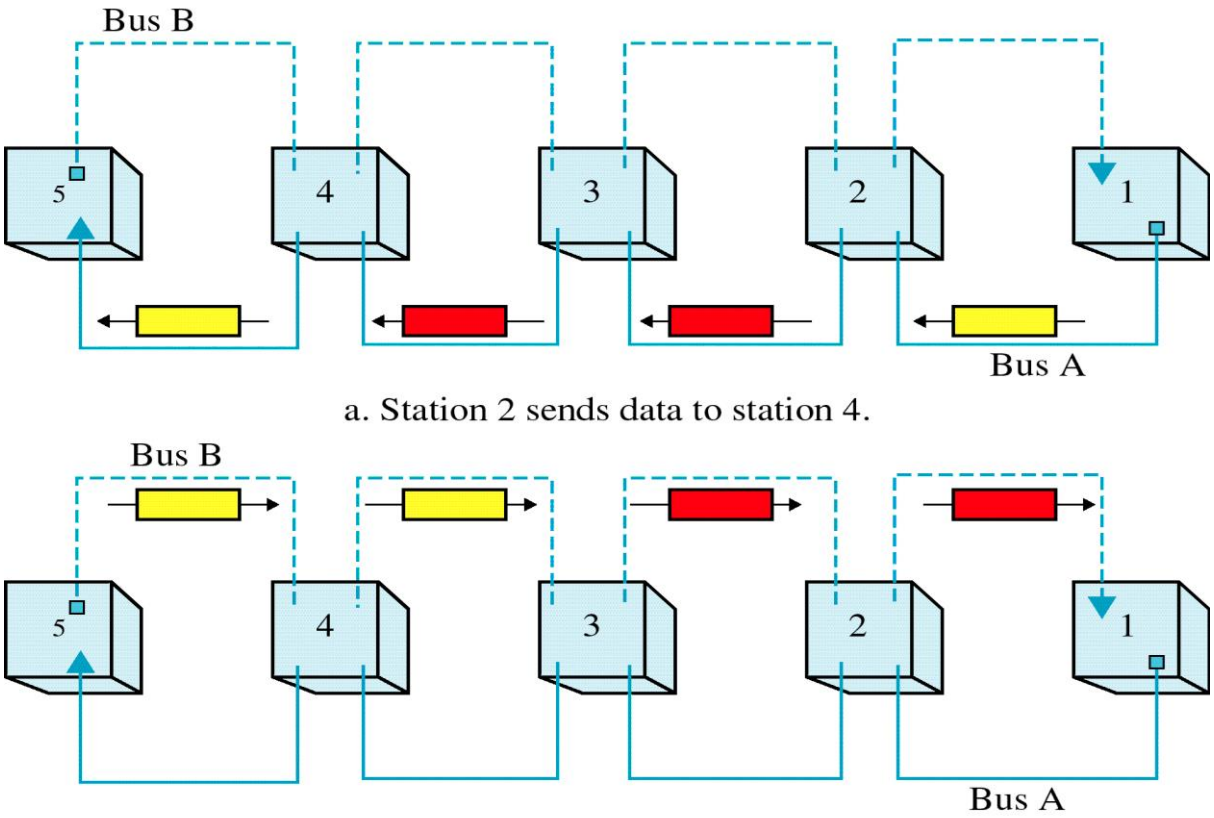
Metropolitan Area Network (MAN)

گفته شد که شبکه های LAN تا یم فاصله معینی پاسخگوی ما می باشد و هنگامی که فاصله بیش از حد معینی شد دیگر شبکه های LAN پاسخگو نیستند و بحث شبکه های MAN مطرح می شود. پس با افزایش طول و با افزایش تعداد nodeها به سراغ شبکه های Dataمی رویم.



شکل ۱۲۸: DQDB (Distributed Queues, Dual Bus)

در این روش بین Nodeها دو شبکه ارتباطی وجود دارد Bus A و Bus B هر باس یک Headدارد که وظیفه آن تولید سلول ۵۳ بایتی خاصی را دارد که از این ۵۳ بایت ۹ بایت آن اطلاعات کنترلی و بقیه اطلاعاتی است که می فرستد (Data) برای فرستادن اطلاعات هر Node باید بداند که از کدام باس باید استفاده کند که این موضوع بر مبنای آدرس فرستنده به گیرنده نشخص می شود به عنوان مثال، ایستگاه ۲ می خواهد اطلاعات را برای ایستگاه ۴ بفرستد. ۴ از لحاظ آدرسی و مکانی پایین تر از ۲ است اصطلاحا به آن (۴) Down Stream گفته می شود. زمانی که یک ایستگاه می خواهد برای ایستگاه Down Stream خود اطلاعات بفرستد از Bus A استفاده می کند که جهت آن از راست به چپ می باشد. (Bus ها یکطرفه هستند)



شکل ۱۲۹: DQDB Data Transmission

Head موجود در Bus A بر مبنای سرعت خود پشت سر هم سلول خالی ۵۳ بایتی تولید مـی کنـد کـه این سلول ها دارای یک بیت Busy یا Requestمی باشند که مشخص می کنند سلول خالی اسـت یـا نه. سلول های خالی به دست ۲ می رسند. ایستگاه شماره ۲ اطلاعات خود (data ۴۳ بایتی) را در آن قـرار داده بیت Busy را فعال و آدرس گیرنده را هم قرار داده و به گیرنده ارسال می کند.

سلول به دست ایستگاه ۳ می رسد. شماره ۳ می بیند که سلول Busy است پس نمی تواند برای فرستادن اطلاعات از آن استفاده کند و آدرس هم با آدرس خودش هماهنگی ندارد. پس سلول را بـه ایسـتگاه بعـدی می دهد. این ایستگاه ۴ سلول را دریافت می کند و می بیند که آدرس خودش است اطلاعـات را برداشـته و بیت Busy را غیر فعال کرده و سلول را به ایستگاه بعدی می دهد.

در مثال بعدی ایستگاه ۲ می خواهد به ایستگاه ۱ که بالا دستی یا اصطلاحا Up Stream خـودش اسـت اطلاعات بفرستد، باید منتظر شود که سلول خالی به دستش برسـد و اطلاعـات را در آن قـرار داده و بـرای گیرنده بفرستد.

Node های انتهایی وظیفه دارند سلول خالی که می رسد، خود ببرد یا Discard کنند. مشکلی که در این روش وجود دارد این است که به Nodeهای انتهایی در هر باس ممکن است که سلول خالی شود یعنی ۴ می خواهد برای اطلاعات بفرستد همه سلول های خالی را استفاده می کند و دیگر سلول خالی به ۲ نخواهد رسید این مشکل برای سیستم های Real Time و Video/Audio وجود دارد که می تواند تأثیر در ارسال اطلاعات داشته باشد.

برای حل مشکل سلول ها رزرو می شوند در Bus A ، Nodeهای ۱ و ۲ به سلول های خالی بیشتری دسترسی دارند و در Bus B ، Nodeهای ۴ و ۵ به سلول خالی بیشتری دسترسی دارند. وقتی که می خواهید به ۵ اطلاعات بفرستد (از طریق Bus A) امکان اینکه سلول خالی به آن برسد خیلی کم است پس از طریق Bus B در می تواند در این Bus ابتدای خط قرار گرفته به Node های دیگر اطلاع دهد که می خواهد از سلول خالی استفاده کند و در حقیقت از این روش سلول خالی را رزرو می کند. پس با این درخواست، وقتی که در خواست به Node های ۱ و ۲ می رسد و آن ها در خواست را می بینند سلول های خالی را کمتر استفاده کرده و اجازه می دند که سلول خالی به ۴ هم برسد. در این حالت در محیط های Read هم به مشکل برخورد نخواهد کرد.

SMDS (Switched Megabit Data Services)

نمونه های استاندارد IEEE ۸۰۲,۶ و DQDB را به شبکه های SMDS میبینیم که دارای سرعت ۴۵Mbps و گستردگی ۱۶۰Kmمی باشند. عموما شبکه های SMDS برای ارتباط بین شبکه های LANاستفاده می گردد.

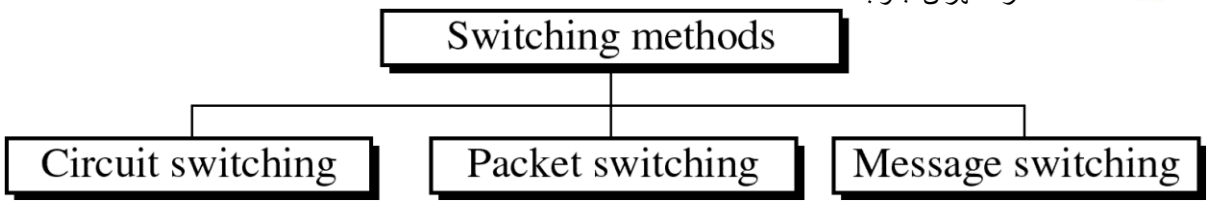
Switching

به ایجاد ارتباط موقتی میان مبدا و مقصد Switching گفته می شود و تنها راه گسترش شبکه، با استفاده از شبکه های مبتنی بر سوئیچ می باشد. سه نوع Switching وجود دارد:

Circuit Switching

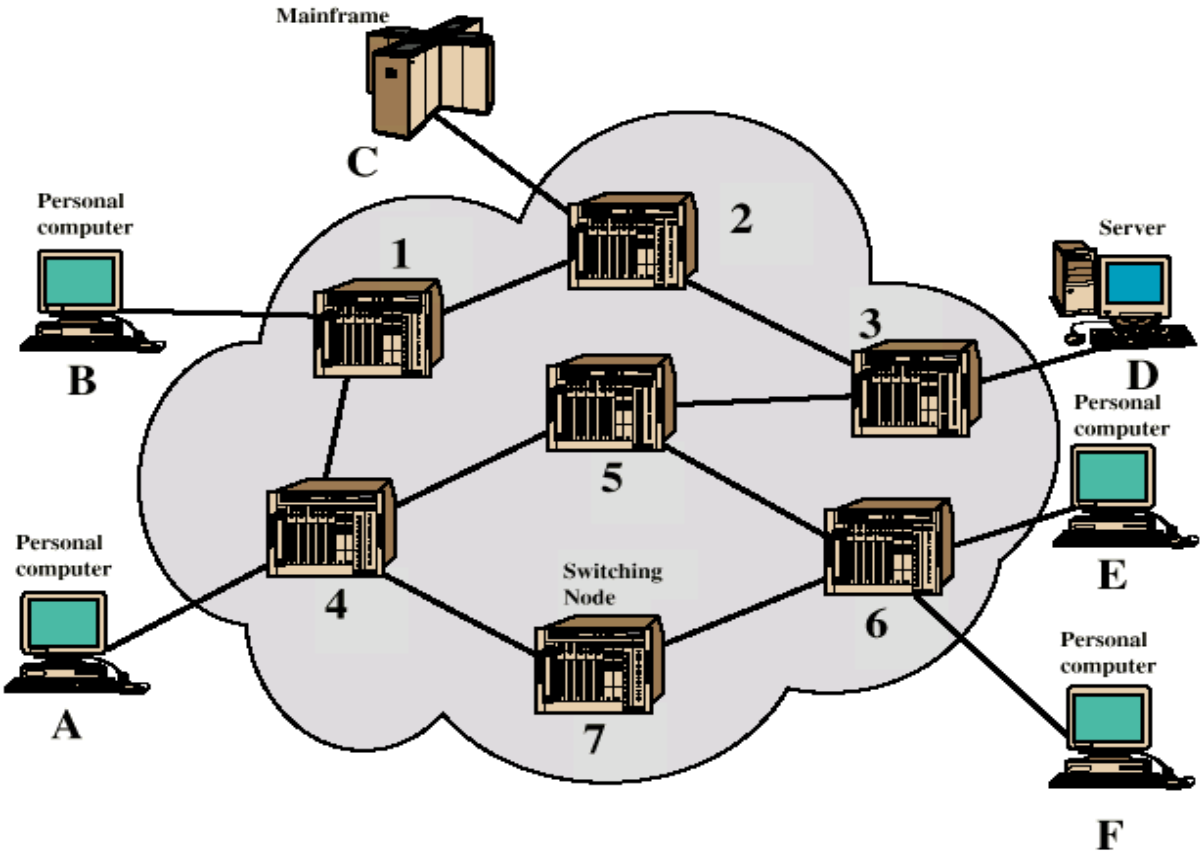
Packet Switching

Message Switching



شکل ۱۳۰: تکنولوژی های سوئیچینگ

Switched Network

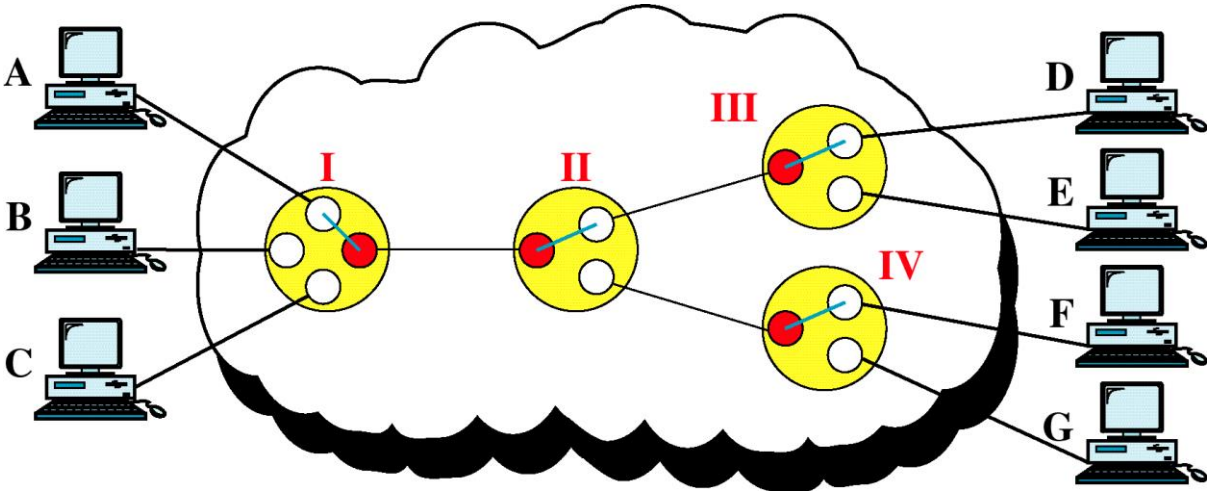


شکل ۱۳۱: Switched Network

در شبکه های مبتنی بر سوئیچ هر Node به سوئیچ های میانی متصلند که همه این سوئیچ ها را حل کرد قرار دارند که Node در خارج از طریق این ابر صورت می گیرد.

وظیفه این ابر به عنوان مرکز سوئیچ این است که وقتی A می خواهد تا E ارتباط برقـرار کنـد، راه گزینـی مناسبی بین سوئیچ ها از طریق ONو OFF کردن آن ها داشته باشد تا بتواند با Eارتبـاط برقـرار کنـد. پس این ابر یک ارتباط موقتی است بین مبدا و مقصد برقرار کرده است و شروع به تبادل اطلاعات می نماید.

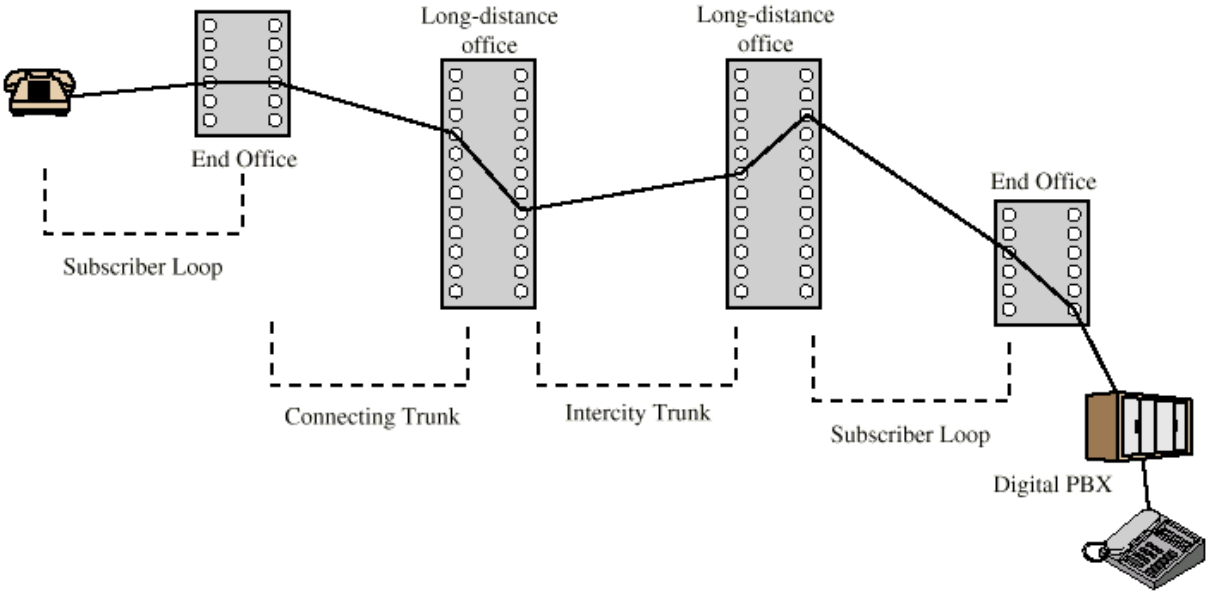
Network Circuit Switching



شکل ۱۳۲: Circuit Switching Network

در این شبکه نیز یک ابر سوئیچینگ وجود دارد. که ارتباط بین Node ها برقرار می گردد. به عنوان مثال Aمی خواهد با Dارتباط برقرار کند. سوئیچ Iو IIو III سوئیچ می کنند تا ارتباط با Dبرقـرار گـردد. در همان لحظه A نمی تواند با F یا با Node دیگری ارتباط برقرار کند چون خط مشغول است.

نمونه یک شبکه تلفنی

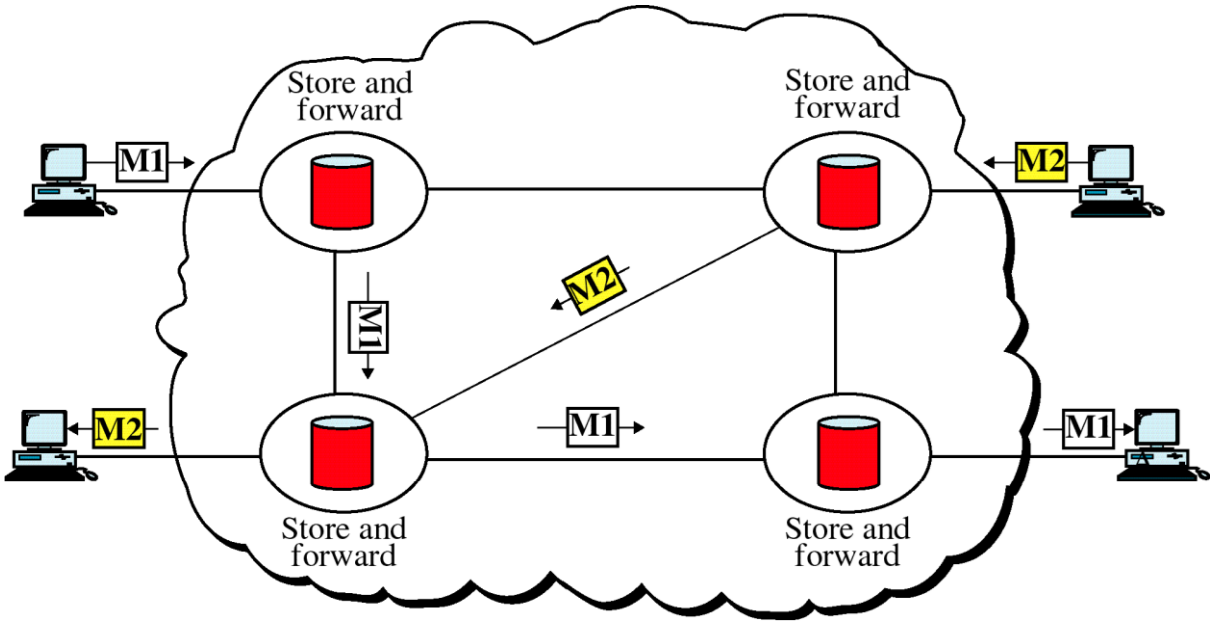


شکل ۱۳۳: Public Circuit Switched Network

در سوئیچ ها بر مبنای شماره هایی که گرفته می شود، بحث راه گزینی مطرح است که کدام راه را به عنوان راه مناسب تر در نظر بگیرند (مراکز سوئیچ) این شبکه که نمونه یک شبکه تلفنی است، مبتنی بر شبکه Circuit Switching می باشد. هر یک از تلفن های مبدا و مقصد به مراکز تلفن های محلی خود وصلند که بین مرکز تلفن های محلی نیز تلفنی میانی قرار دارد که این مراکز ارتباطات را از مبدا به مقصد راه گزینی مناسب برقرار می کند.

وقتی که شماره طرف مقابل را می گیریم، مراکز تلفن یک خط را به ما اختصاص می دهند و این خط رزرو ما می ماند تا زمانی که هر دو طرف گوشی تلفن را بردارند. تا زمانی گوشی تلفن گذاشته نشده (چه حرف بزنیم و چه حرف نزنیم) مرکز تلفن ما را شارژ زمانی می کند و کسی حق استفاده از این خط را ندارد.

این روش زمانی مشکل دارد که ما به اینترنت وصل می شویم و یک متن طولانی را شروع به خواندن می کنیم در طول این مدت خط بیکار می ماند و هیچ رد و بدل اطلاعاتی وجود ندارد ولی خط اشغال است پس در ارتباط برای شبکه های داده ها (مثل اینترنت) مشکل دارد و به دنبال شبکه ای می گشتند که این مشکل را نداشته باشند و ما را بر مبنای حجم اطلاعات شارژ کند نه بر مبنای زمان که این شبکه ها خیلی به نفع کاربران است زمانی که Dataرد و بدل می کنند. بر مبنای این ایده شبکه های Message Switching پیاده سازی شدند.



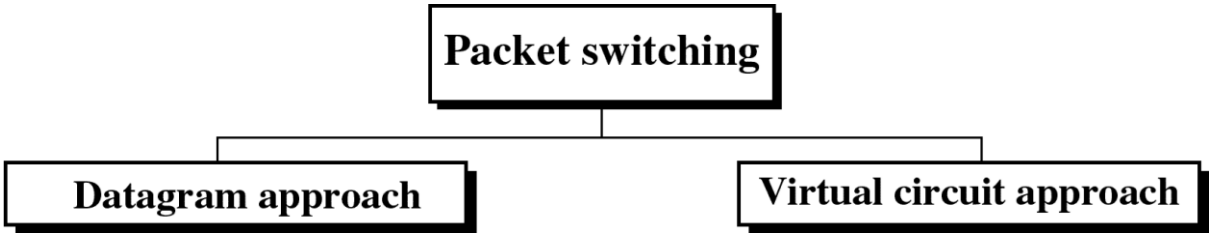
شکل ۱۳۴: Message Switching

پس به دنبال شبکه هایی می گشتند که اولا زمانیکه ما اطلاعات را ارسال می کنیم خط رزرو ما نباشد و دیگران هم بتوانند از خط استفاده کنند. دوما ما را بر مبنای داده ها شارژ کند که بحث Message Switching مطرح شد و ابر همان ابر مخابرات است ولی به جای مراکز سوئیچ و سوئیچ ها، کامپیوتر هایی قرار دادند که یک حافظه جانبی داشت. وقتی کامپیوتر Aمی خواهد اطلاعات را برای کامپیوتر D ارسال کند، کل اطلاعات را در قالب یک Messageبسته بندی می کند (M۱) و تحویل روابط می دهد که رابطه اطلاعات را گرفته در حافظه اش ذخیره می کند و در یک فرصت مناسب مسیری را انتخاب کرده و برای Dمی فرستد. همچنین اگر از Node B بخواهد برای D اطلاعات بفرستد همین عملیات را انجام می دهد. پس در این حالت خط رزرو به کسی نمی ماند.

در این روش نیز اشکالاتی وجود دارد از جمله اینکه اگر Message ارسال شده گم شود، کل اطلاعات باید دوباره فرستاده شود که ممکن است این اطلاعات حجم زیادی داشته و زمانی از دست برود. همچنین برای پیام های فوری نیز مناسب نیست چون رابط آن را نگه داشته و بعد ارسال می کند و از همه مهم تر اینکه میزان بافر رابط مشخص نیست و امکان کمبود بافر در آن وجود دارد پس این روش جوابگوی کار نمی باشد. به همین دلیل شبکه های Packet Switching را ابداع کردند.

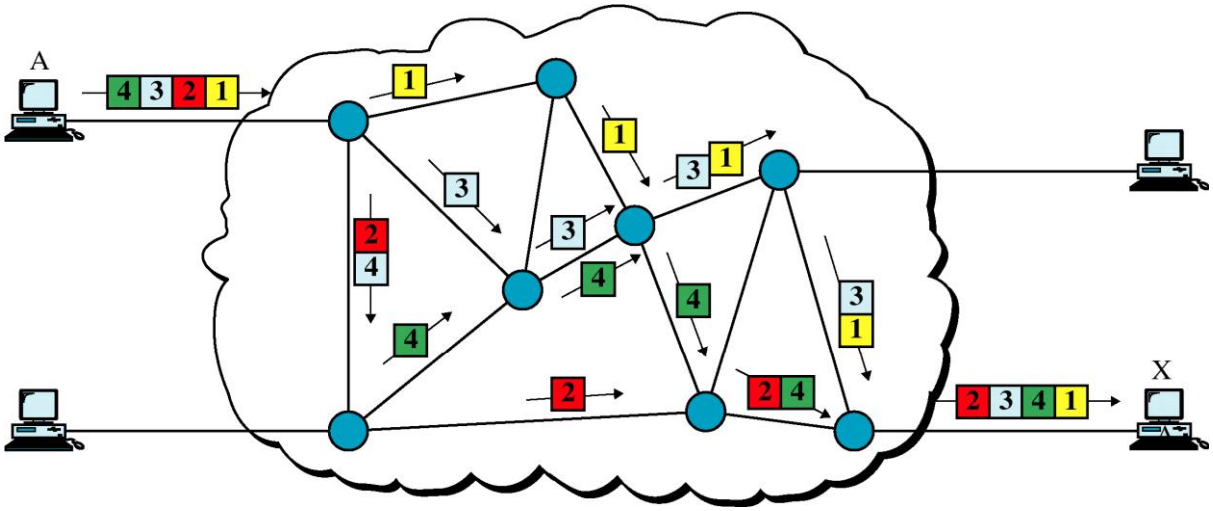
Packet Switching

شبکه های Packet Switching را با دو دیدگاه Approachابداع کردند.



شکل ۱۳۵: دیدگاه های Packet Switching

Datagram Approach



شکل ۱۳۶ : Datagram Approach

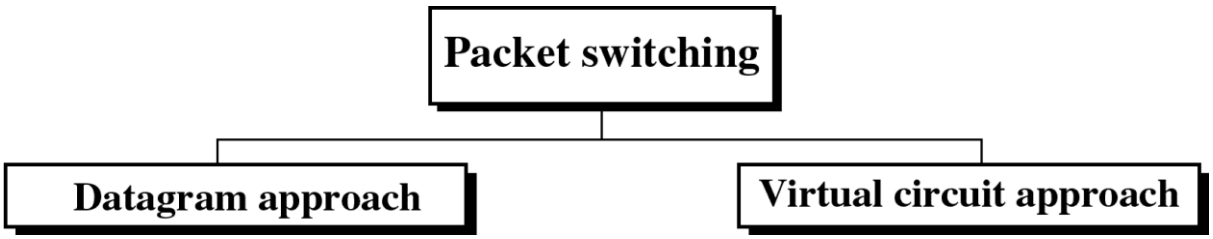
در این روش فریم به بسته های کوچکتر شکسته می شود و هر یک از این مسیر خاص ارسال می شود. وقتی که واحد ها کوچکتر باشد، اولا بافر زیادی احتیاج ندارد و حداقل بافر را نیاز دارد و ثانیا خط اشغال نیست و دیگران هم می توانند از خط استفاده کنند و ثالثا اگر یک قطعه ارسالی خراب شود همان یک قطعه ارسال می شود نه همه اطلاعات. پس ترافیک شبکه هم پایین می آید پس کل مشکلات Message Switching در این روش کاملا حل شده است. در این حالت است که شبکه می تواند ما را برای مبنای حجم اطلاعات شارژ کند نه برای زمان.

در اینجا به Deviceهایی که عمل راه گزینی را انجام می دهند Router یا مسیریاب گفته می شود که عمل مسیریابی را انجام می دهد. نمونه این شبکه ها در شبکه های اینترنت دیده می شود که بر مبنای Packet Switching می باشند و ما را شارژ زمانی نمی کند و بر مبنای حجم اطلاعات ما را شارژ می کند. به عنوان مثال Provider ای که ما از سرویس اینترنت دریافت می کنیم بر حسب کیلوبایت هزینه را پرداخت می کند نه بر حسب ساعت چون شارژ آن بر مبنای حجم است نه بر مبنای زمان.

مشکلی در روش Datagram وجود دارد آن هم عدم ترتیب بسته ها در گیرنده ها می باشد که این بسته ها باید مرتب شود.

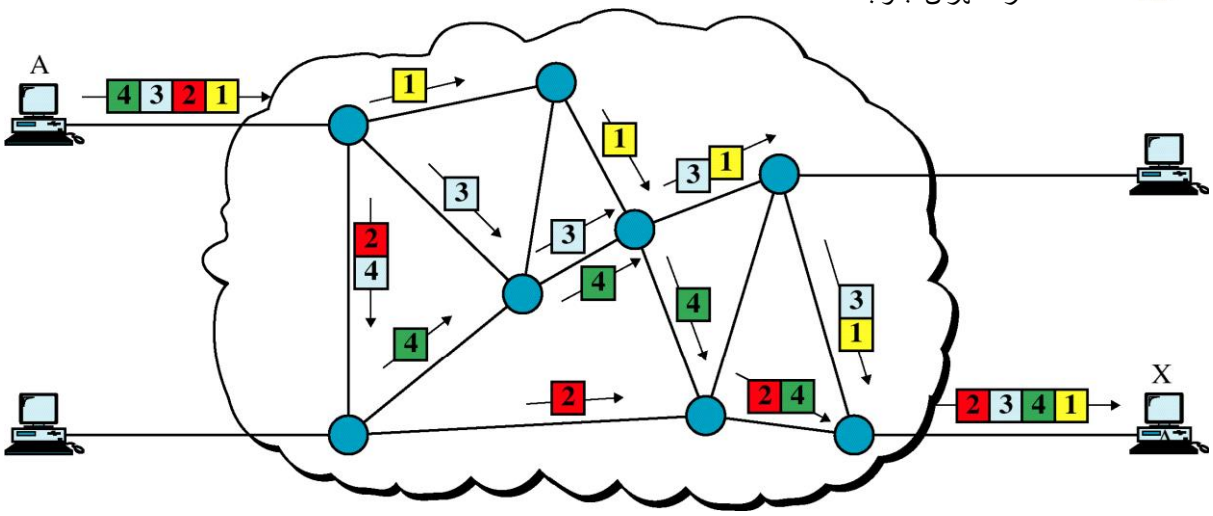
Packet switching

شبکه های Packet switching را با دو دیدگاه Approach ابداع کردند.



شکل ۱۳۵: دیدگاه های Packet switching

Datagram Approach



شکل ۱۳۶: Datagram Approach

در این روش Frame به بسته های کوچکتر شکسته می شود و هر یک از این مسیر خاص ارسال می شـود. وقتی که واحد ها کوچکتر باشد، اولاً بافر زیادی احتاج ندارد و حداقل بافر را نیـاز دارد. و ثانیـاً خـط اشـغال نیست و دیگران هم می توانند از خط استفاده کنند و ثالثاً اگر یک قطعه ارسالی خراب شود همان یک قطعه ارسال می شود نه همه اطلاعات. پس ترافیک شبکه هم پایی ن می آیـد، پـس کـل مشـکلات Message Switching در این روش کاملاً حل شده است. در این حالت است که شبکه می تواند ما را بر مبنای حجم اطلاعات شارژ کند نه بر مبنای زمان.

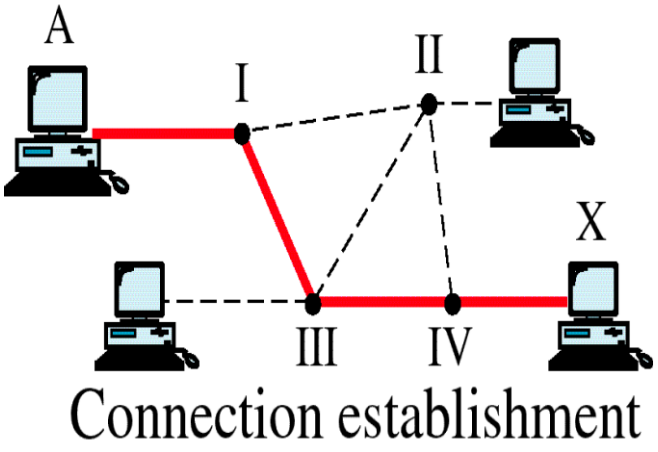
در اینجا به Divice هایی که عمل راه گزینی را انجام می دهند Router یا مسیریاب گفته مـی شـود کـه عمل مسیریابی را انجام می دهند. نمونه این شبکه ها در شبکه های اینترنت دیده می شود کـه بـر مبنـای Packet Switching می باشند و ما را شارژ زمانی نمی کنند و بر میبنای حجم اطلاعات ما را شـارژ مـی کنند. به عناون مثال Providerای که ما از سرویس اینترنت دریافت می کنیم بر حسب کیلو بایدت هزینه را دریافت می کند نه بر حسب ساعت چون شارژ آن در مبنای حجم اطلاعات می باشد نه بر مبنای زمان.

مشکلی در روش Data gram وجود دارد، عدم ترتیب بسته ها در گیرنده ها می باشد که این بسته ها باید مرتب شود.

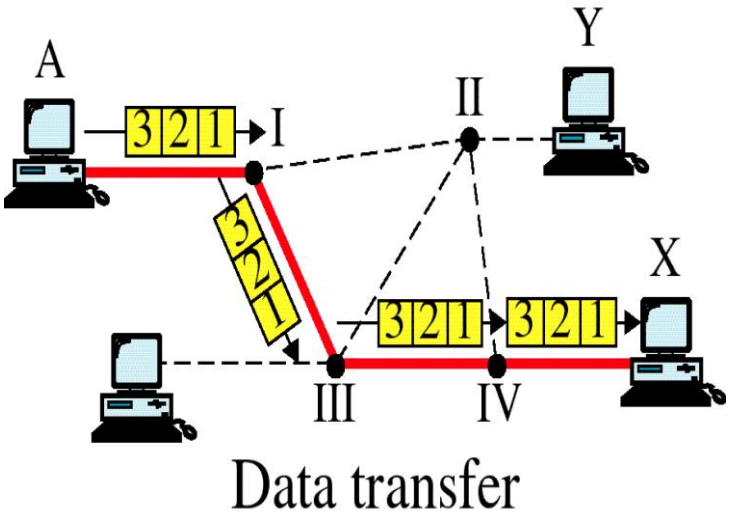
Virtual Circuit Approach

در این حالت سه فاز داریم:

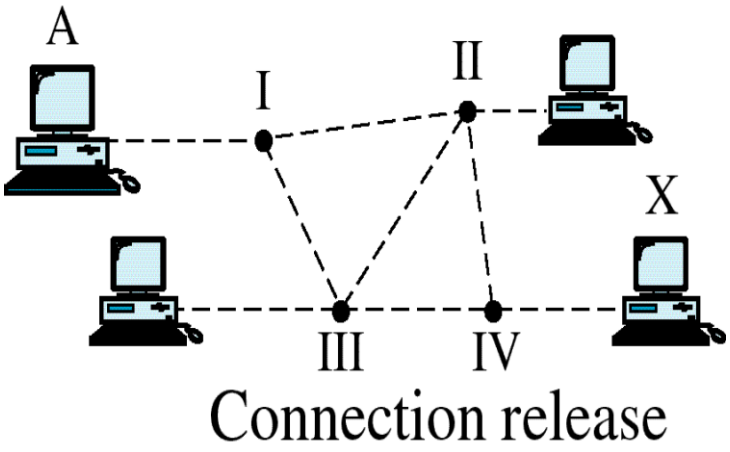
۱. Connection establishment
۲. Data Transfer
۳. Connection Release



شکل ۱۳۷: Connection establishment



شکل ۱۳۸: Data Transfer



شکل ۱۳۹: Connection Release

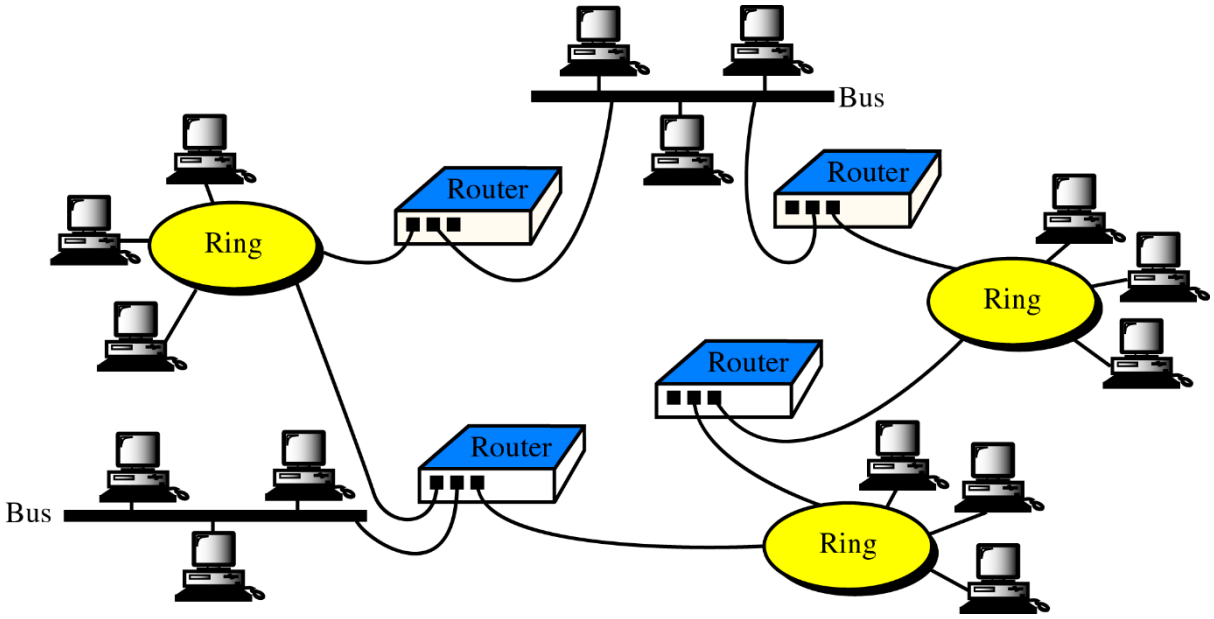
وقتی که A می خواهد با X ارتباط برقرار کند در فاز اول، یک درخواست می فرستد کـه مـی خواهـد بـا X ارتباط برقرار کند. که این درخواست به صورت یک Packet می باشد و این Packet از هر Routerای که عبور می کند در جدول Router قرار می گیرد چون Packet در حالت عادی یک مشخصه واحد دارد کـه شماره را در داخل Router درج می کند که هر Router مسیر بعدی را تشخیص می دهـد و نشـان مـی دهد که باید از کدام مسیر برود. در این حالت که Packet به مقصد می رسد، ارتبـاط بـین مبـدا و مقصـد برقرار می گردد که این ارتباط لاجیکی است و رزرو ما نیسـت. در خواسـت دوم، تمـام Packetهـای یـک اطلاعات از همان مسیر خاص عبور کرده اند چون همه Packetهـا مشخصـات همـان اطلاعـات را دارنـد و Router وقتی به مشخصات نگاه می کند و می بیند که همان مشخصات است، دوباره از همان مسیر ارسال می کند. در این روش چون همه Packetها از یک مسیر ارسال می شود در گیرنده به ترتیـب Packetهـا دیده می شود که احتیاج به مرتب شدن هم ندارد. در فاز سوم ارتباط قطع می گردد. این روش خیلی شبیه به Circuit Switching می باشد. به همین دلیل Circuit همسازی به آن گفته نمی شود و خط رزرو ما نیست و ویژگی این روش این است که اطلاعات در گیرنده مرتب شده اسـت چـون اگـر مرتـب نبـود زمـان زیادی برای Sort کردن نیاز داشت. یکی دیگر از ویژگی های این روش این است که Router هـا از پیـاده سازی عمل مسیریابی خلاص نمی شوند و برای تمام Packetهای مربوط به یک اطلاعات تنها یکبار عمـل مسیریابی صورت می گیرد. همچنین این روش مشکلات خاص خود را دارد. چون اگر یکی از Router ها از کار بی افتد، دوباره فاز ارتباط (فاز اول) باید طی شود و عمل مسیریابی صورت گیرد.

اگر حجم اطلاعات کم باشد منطقی تر است که از روش Data gram استفاده شود چون زمان فاز ارتباط از دست نخواهیم داد.

در این لایه دو نوع سرویس دهی مطرح می شود (طبق شکل) که سرویسهای Connection oriented ، سرویسهای مبتنی بر ارتباط است که ارتباطی بین مبدا و مقصد برقرار می شود که در این ارتباط بحث تبادل اطلاعات مطرح می شود و عموماً این سرویسها Virtual Circuit استفاده می کنند و کند می باشند ولی در مقابل در این سرویس ها اطمینان وجود دارد. همچنین در این سرویس ها Overhead کمتر می باشد.

در سرویسهای Connection Less که عموما از Data gram استفاده می کنند، سرعت تبادل اطلاعات بالا می باشد. Overhead به نسبت بیشتر است و در این سرویس ها بدون اینکه مسیر مشخص شود اطلاعات ارسال می گردد و دارای Router هایی می باشد که هر Router بحث مسیریابی را دارد. در این نوع سرویس های Connection Less سرعت را داریم ولی در مقابل صحبت را از دست می دهیم چون ممکن است اطلاعات در حین راه از بین برود. به عنوان مثال از این دو نوع سرویس، می توان گفت، سرویس تلفن یک سرویس از نوع Connector oriented و سرویس پشت یک سرویس Connection Less می باشد.

گفتیم که در شبکه های مبتنی بر سوئیچ، احتیاج به یک سری عناصر راه گزین داریم چه در شبکه های Circuit Switching به آنها سوئیچ و در شبکه های Packet Switching به آنها Router یا مسیریاب گفته می شود. هر دو بحث مسیریابی و راه گیزنی دارند. همانطور که در شکل دیده می شود در بحث ارتباطات بین شبکه ای ما می خواهیم شبکه هایی را که از لحاظ معماری متفاوتنند به یکدیگر وصل کنیم.



شکل ۱۴۰: Router in an Internet

مثلاً یک شبکه Token Ring، شبکه دیگر Token Bus و بعدی FDDL می باشد که با هم متفاوتند برای وصل کردن این شبکه ها به یکدیگر احتیاج به عنصری به نام Router یا مسیریاب داریم. عنصری که از دید آن شبکه ما در قالب یک شبکه یکپارچه دیده می شود. که این عنصر قطعاً در لایه سوم قرار دارد. (NETWORK)

پس این شبکه های متفاوت از دید لایه Data Link با همدیگر فرق دارند ولی از دید لایه Network در قالب یک شبکه یکپارچه دیده می شود. پس بایه Network باز می گردد به بحث ارتباطات بین شبکه ای. پس وقتی که می خواهیم با شبکه های دیگر ارتباط برقرار کنیم از لایه Network استفاده می کنیم. در غیر این صورت یعنی زمانی که با ارتباطی با شبکه های دیگر نداریم ، لایه Network نیز مطرح نخواهد شد.

گفتیم که عنصر مسیریاب در لایه سوم قرار می گیرد و هیچ لایه ای بطور مستقل نمی تواند ارتباطی با محیط داشته باشد. پس Router، دارای لایه Phisical می باشد، تا سیگنال را روی خط قرار دهد دارای لایه Data Link می باشد تا چک Frame و Flow Control و Error Control را داشته باشد و لایه Network برای مسیریابی بهتر.

پس در لایه Network که وظیفه مسیریابی را بر عهده دارد، بحث تصمیم گیری روی مسیرهای متفاوت نیز مطرح می شود.

Network همچنین از دید لایه Network که بحث آدرسهای لاجیمی در آن مطرح می شود، همه شبکه در قالب یک شبکه یکپارچه دیده می شود، آدرس های یکنواخت و ثابت که هنگام ساخت Packet آدرس های لاجیکی گیرنده و فرستنده در آن درج می شود. Packetای در لایه شبکه با استفاده از آدرس های لاجیکی ساخته شده برای فرستاده شدن باید به لایه Data Link فرستاده شود. از دید لایه Data Link، آدرس های لاجیکی نا مفهوم هستند و به دنبال آدرس کارت شبکه می گردد. این سرویس را باید لایه Network به لایه Data Link بدهد که همزمان که Frame را به لایه Data Link می دهد آدرس کارت شبکه فرستنده و گیرنده را هم بدهد. لایه Network، آدرس کارت شبکه خود را می داند. آدرس کارت شبکه گیرنده را باید به دست آورد. چون شبکه های ما در اینجا LAN هستند (طبق شکل) کافی است که یک اطلاعات Brand Cast بفرستد که مثلاً پس آدرس لاجیکی مربوط به آدرس کارت شبکه کیست؟ که این درخواست را همه دریافت می کنند و کسی که آدرس لاجیکی مربوط به آن است، پاسخ خواهد داد و آدرس کارت شبکه اش را می دهد. پس لایه Network به راحتی آدرس کارت شبکه گیرنده

را بدست آورده و در اختیار لایه زیرین (Data Link) قرار می دهد و Data Link هم با استفاده از آدرس های فرستنده و گیرنده Packet را می سازد و به لایه Physical می دهد.

هر Routerای که به شبکه وصل می شود، جزئی از آن شبکه می شود. مثلاً Router شماره ۳ جزئی از ۳ شبکه Ring ، Bus ، Rig می باشد. قطعاً وقتی جزئی از شبکه است، یک آدرس لاجیکی و یک آدرس کارت شبکه دارد و وقتی Frameای را دریافت می کند وقتی که آدرس آن با آدرس کارت شبکه خودش هم خوانی نداشته باشد DisCard می کند. Routerها به عنوان یک رابط عمل می کنند و اگر Frameای که دریافت می کنند با آدرس شبکه داخلی خودشان هماهنگی نداشت، آنرا به رابط بعدی می دهد. چون Routerها هم دارای پورتی می باشند که هر Port دارای یک آدرس لاجیکی و یک آدرس کارت شبکه می باشد. در هر Clint set شده است که اگر آدرس ها مطابقت نداشت، برو به شبکه بعدی با آدرس مشخص. که این آدرس هم همان روش Board Cast بدست می آید.

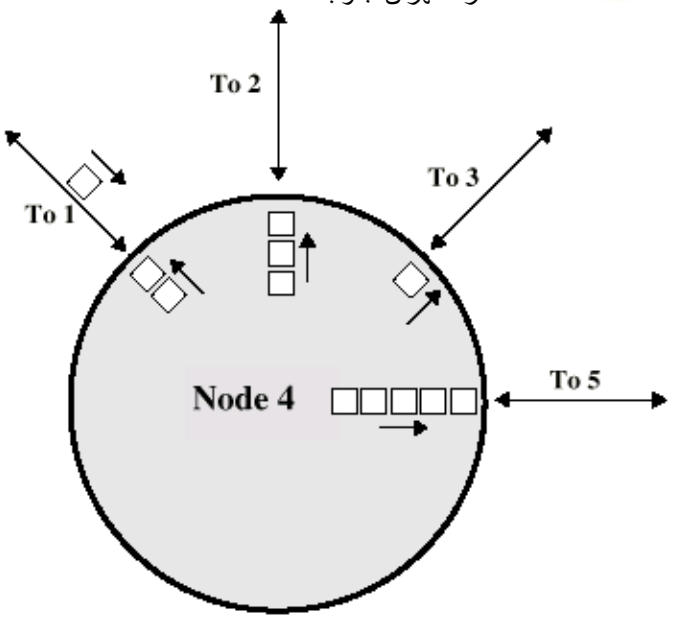
وقتی که در بین شبکه ها بحث مسیریابی مطرح می شوند، Router طبق جدولی که آن را Set کرده است مقایسه کرده و مسیر بهتر را از جدول بدست می آورد. پس از اینکه Frame به Router ارسال می گردد. Router چک می کند تا ببیند که شبکه هایی که به آنها وصل شده، Packet ارسال شده با آدرس nodeهای موجود در شبکه های آن مطابقت دارد یا خیر. برای بدست آوردن آدرس کارت شبکه نیز Broad Cast

صفحه ۹۰

۲- روش ایزوله (Isolated):

در این روش، هر کس الگوریتم خاص خود را اجرا می کند. از جمله روش هایی که د اینجا، می توان به روش HOT Potato و روش Flooding اشاره کرد.

روش HOT Potato:



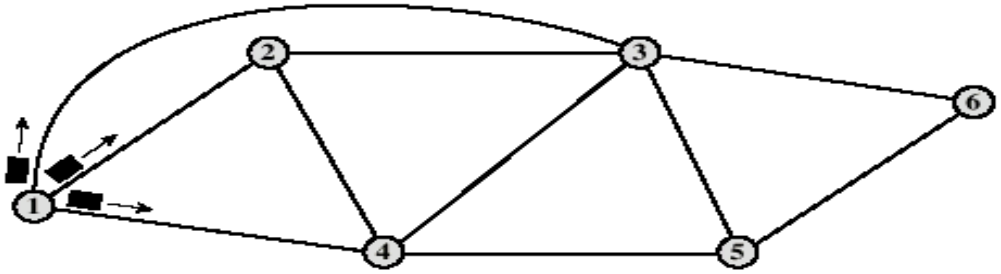
Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0

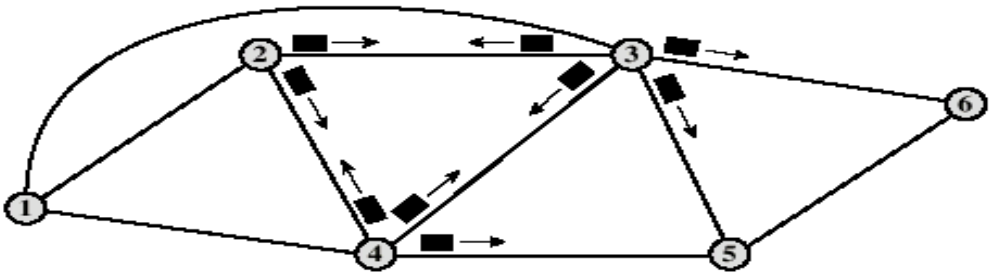
شکل ۱۴۲: مثال HOT Potato

در این روش، اطلاعاتی که از ورودی دریافت می شود، بدون توجه به مقصد در کوتاهترین صف خروجی قرار گرفته و بلافاصله خارج می شوند. در حقیقت، Router ها هم به همین ترتیب عمل می کند که وقتی ورودی را می گیرد به دنبال مسیر گشته و ورودی را روی آن مسیر قرار می دهد. که می توان به عنوان یک روش از این روش استفاده کرد. (زمانی که می خواهیم چیزی را از حالت بن بست بیرون آوریم).

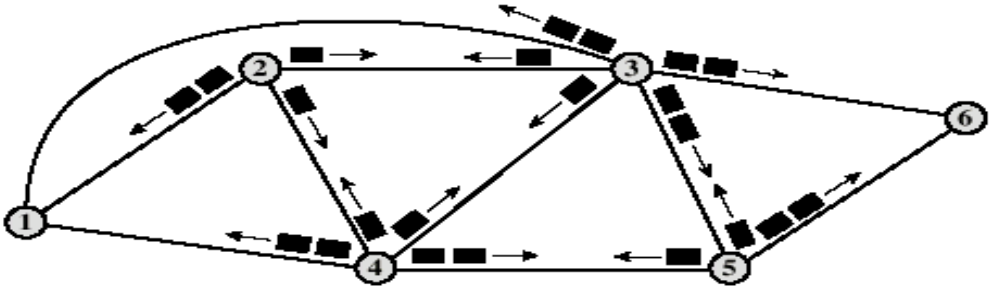
روش Flooding:



(a) First hop



(b) Second hop



(c) Third hop

شکل ۱۴۳: مثال Flooding

در این روش که به نسبت کاربرد زیادی دارد، هر Packet ای که از ورودی می آید، در تمامی خروجی ها ارسال می گردد که همان طور که در شکل دیده می شود در مرحله اول، یک Packet به سه Packet تبدیل شده و در مرحله بعد به همین ترتیب. که مرتب رشد کرده و تعداد این Packetها را در شبکه زیاد می شود. ویژگی این روش این است که ما مطمئن هستیم که در کوتاه ترین زمان Packet به مقصد می رسد و ارسال آن به مقصد ۱۰۰٪ می باشد و در مقابل ترافیک را بسیار زیاد افزایش می دهد ولی در شبکه ها هنوز از این روش استفاده می گردد. پس باید به دنبال راه حلی باشیم که وقتی Packet مورد نظر به مقصد رسید، بقیه Packetها که در شبکه در حال زیاد شدن هستند از خود خارج شوند. برای این کار باید چک کنیم و ببینیم که Packet ما، ماکزیمم بعد از چند پرش یا HOP باید به مقصد برسد؟ می توان به عنوان سیمبلیک این تعریف را داشت که از یک Router به Router پورتی یک HOP یا پرش را داریم به

عنوان مثال فرض می کنیم که Packet در بهترین شرایط باید نور از ۲۰ تا HOP بـه مقصـد برسـد چـون میدانیم که الگوی Network ما چیست و چند Router در آن قرار گرفته است.

کافی است که یک فیلد در اطلاعات کنترلی Packet قـرار دهـیم و فیلـد TTL بنـامیم و Counter آن را روی عدد ۲۰ Set کنیم. از هر Router عبور می کند، Router چک می کند مه این فیلد صفر اسـت یـا نه. اگر صفر بود و به مقصد نرسیده بود یعنی Packet شر گردان است و آن را DisCard می کنـد وگرنـه یکی از Counter کم می کند و ادامه می دهد مرتب از این شماره کم می شود تا به مقصد برسد و یا صفر شود در مقصد نرسد. قطعاً بعد از مدتی DisCard شده و از شبکه خارج می شود.

در جاهایی که Router ها می خواهند اطلاعات را با یکدیگر رد و بدل کننـد از روش Flooding اسـتفاده می کنند (از زمانی که می خواهند جدول هایشان را با یکدیگر Set کنند). فرم جدول ها در Router ها به صورت شکل مقابل می باشد که این جدول یک جدول کلی است که کل اطلاعات در کل شبکه را در بر می گیرد.

Network ID	Cost	Next Hop
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .

شکل ۱۴۴: Distance vector routing table

اندازه HOP یا پرشی که برای Packetها در نظر گیریم نه باید زیاد باشد کـه موجـب بـالا رفـتن ترافیـک شبکه گردد و نه کم باشد که Packet به مقصد نرسد.

۳- روش (Distributed):

در مورد این بعداً بحث خواهد شد.

روش های استراتژی Non_Adaptive:

گفته شد که این استراتژی یک استراتژی ثابت و انعطاف ناپذیر است و دارای دو روش می باشد. روش Shortest path و روش Multi path.

روش Shortest path:

در این روش که یک روش Static می باشد، یک گراف از شبکه رسم می شود (از Router که وجود دارند).

CENTRAL ROUTING DIRECTORY

To Node \ From Node	1	2	3	4	5	6
1	—	1	5	2	4	5
2	2	—	5	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

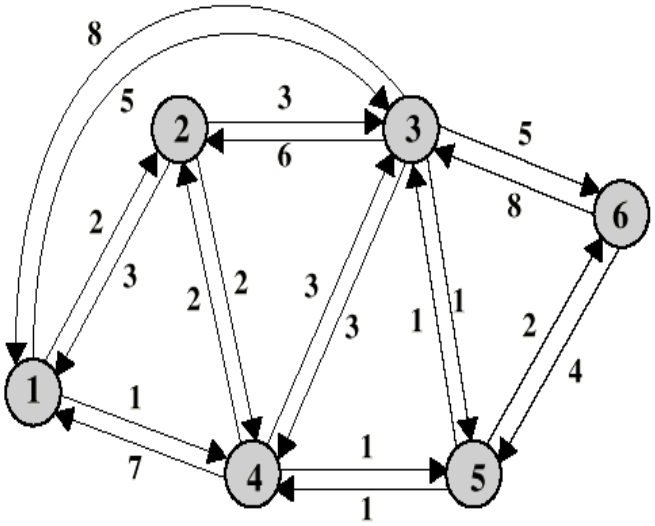
Destination	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5



شکل ۱۴۵: Shortest path

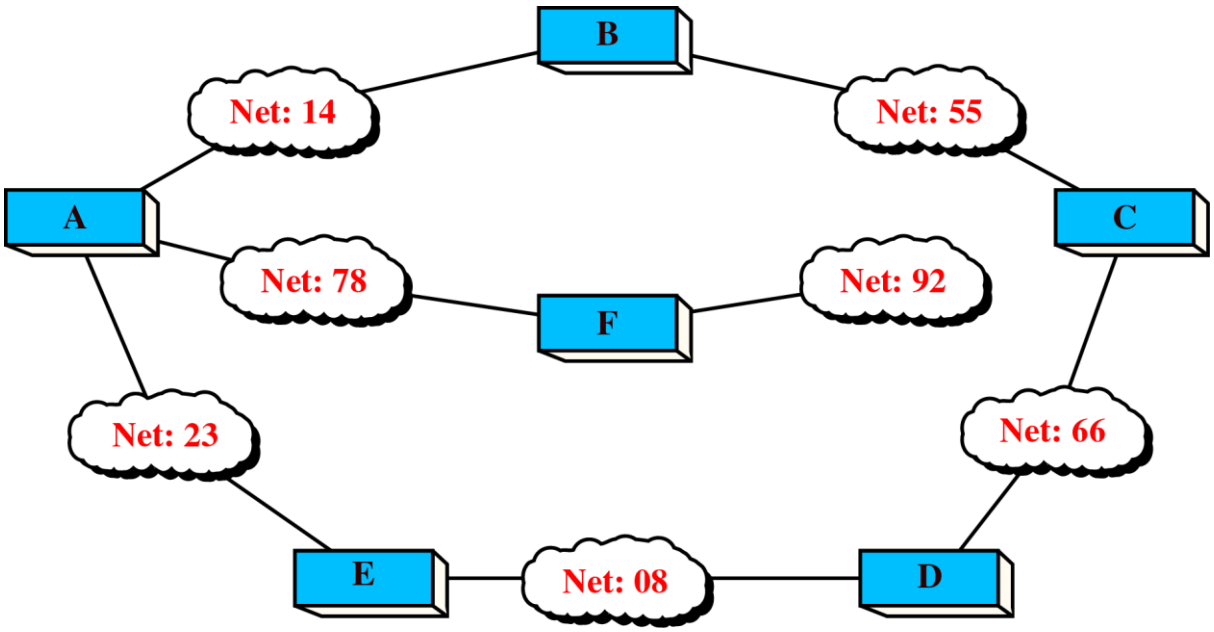
هر Link ارتباطات بین شبکه را نشان می دهد و به هر link وزنی می دهد که این وزن می تواند بر مبنای هزینه و عیناً می باشد. فاصله و ... باشد و یا کل این پارامتر ها را در نظر گرفته و وزن Link مربوطه را مشخص می کنیم و گراف را با این وزن ها رسم می کنیم.

حال برای هر Node، Table مربوطه را رسم می کنیم. به عنوان مثال یک packete از node شماره ۱ به node شماره ۳ می خواهد برود. طبق این جدول و با توجه به گراف اگر از مسیر node شماره ۴ به ۳ برود، دارای وزن کمتری خواهد بود. طبق همین روش، برای تمام nodeها یک جدول به دست می آوریم و بر مبنای این جداول، routerها را به صورت دستی setup می کنیم. مشکل این روش این است که اگر یک node به شبکه اضافه شود، کل این مسیرها دارای وزن متفاوتی با حالت قبل خواهند شد و مسئول شبکه دوباره باید این جداول را به دست آورد. این روش static است و انعطاف ندارد. همیشه از یک مسیر برای رفتن به node بعدی استفاده می کند و اگر ترافیک مسیر هم زیاد بود، باز از همان مسیر استفاده می کند. برای انعطاف دادن به این روش بحث Multi path working را مطرح می کنند که در شبکه های Circuit switching استفاده می گردد. که پارامتر زمان را در این روش اثر می دهند که در زمان های متفاوت مسیر های متفاوت را در نظر می گرفتند.

روش Multi path:

برای انعطاف دادن بیشتر به روش Shortest path این روش مطرح شد که در واقع همان روش Shortest path می باشد ولی از پارامتر زمان برای انعطاف پذیری بیشتر استفاده نموده است. ولی باز هم این روش، یک روش استاتیک است و درست به ؟؟؟ Circuit switching از این روش استفاده می شود.

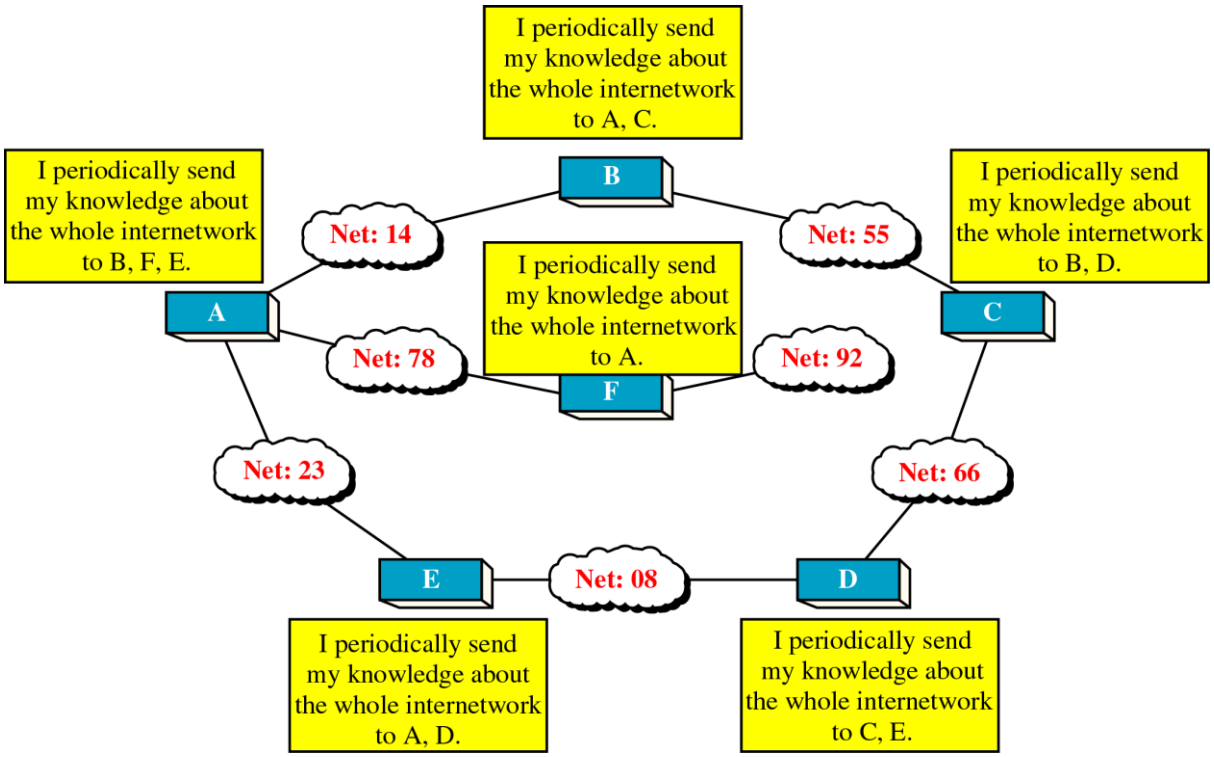
روش Distributed:



شکل ۱۴۶: Example of an Internet

در این روش که از روش های استراتژی Adaptive می باشد، مسیر یابی ها در زمان هـایی خـاص جـدول های Routering خود را با یکدیگر مبادله می کنند و روش بسیار پر کاربرد است. از روش های موجـود در Distributed، روش Distance vector Routing و روش LS یـا Link State Routing کـه بسـیار بسیار الگوریتم های معروفی هستند و مورد استفاده زیادی دارند جدولهای Routing معمولاً ۳۰ ثانیه یکبار بین Routerها معادله می شود.

روش Distance vector Routing:



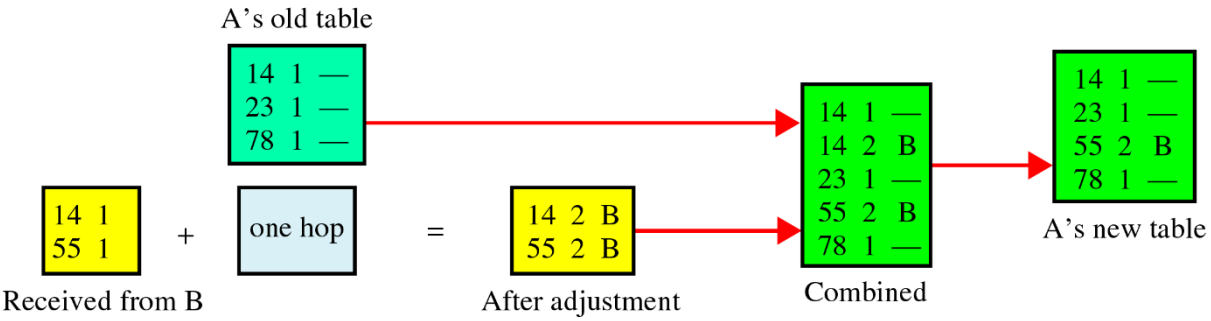
شکل ۱۴۷: The Concept of Distance Vector Routing

هر شبکه یک شماره دارد. مثل شبکه ۱۴،۵۵،۹۲،... که اینها از ارتباطات بین شبکه ای را برقرار می کننـد و به هر Router یک نـام داده مـی شـود. مـی خـواهیم ببینـیم کـه طبـق الگـوریتم Distance vector Routing نحوه پر کردن اطلاعات به اول به چه صورت می باشد در این روش، هر Router به صورت دوره ای به اول خود با به Routerهای همسایه ارسال می کند. جدول خود را برای Router هـای A و C مـی فرستد و به همین صورت همه Router ها به اول خود را برای همسایه های خود ارسال می کنند.

جدول Routerها طبق شـکل شـماره ۱۹_۲۱ در صـفحه ۶۱ دارای ۳ قسـمت مـی باشـد. شـماره شـبکه (nework ID), Cost و next Hop (Router بعدی)

Castها در Distance vector Routingها بر مبنای پارامتر زمانی Hop تنظیم می شود و یا بر مبنـای تاثیر زمانی می باشد. برای بدست آوردن تاخیر زمانی کافیست A یک packete بـرای B ارسـال کنـد کـه شمانی که بدست می آید، تاخیر می باشد.

Setup کردن جدول Router ها :



شکل ۱۴۸: بروز رسانی جدول مسیریابی برای مسیریاب A

این روش یک روش Dynamic می باشد پس وقتی کـه مـا یکبـار آنـرا Setup کـردیم، خـودش خـود را Update می کند.

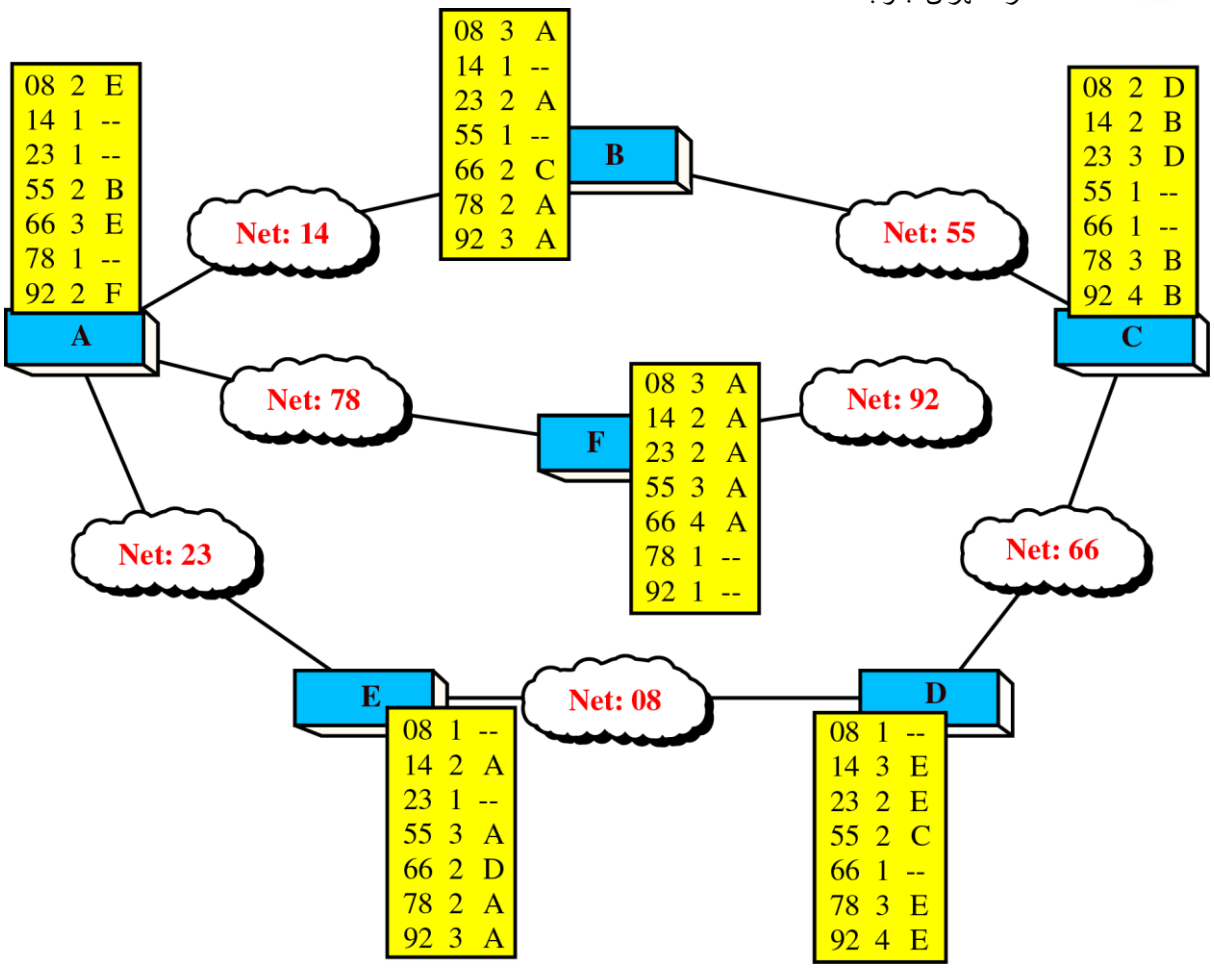
در این مثال، Costها را بر مبنای پارامتر زمانی Hope تنظیم می کنیم.

اگر بخواهیم جدول Router A جزو شبکه A است پس Router های A و B از یکدیگر با خبرند چون هر دو شبکه ۱۴ می باشند.

اولین پارامتری که در جدول Set می شود Routerهای همسایه یک Router است. A به شبکه ۱۴ وصل است که دارای ۱ Cost می باشد، بر شبکه ۲۳ با ۱ Cost و به شبکه f۸ هم با ۱ Cost وصل است.

پس اولین فاز برای Setup جداول، Setup مسیریابی می باشد، به همین صورت بـرای تمـام Routerهـا این کار صورت می گیرد.

در فاز دوم هر Router جداول خود را به صورت حوزه ای برای همسایه خود ارسال می کنـد. (فـرض مـی کنیم که B می خواهد اطلاعات جدول خود را برای A بفرستد وقتـی B اطلاعـات را مـی فرسـتد از طریـق شبکه ۱۴ دارای یک Hop است که چون Cost خودش هم یک بوده، با یک جمع شده و Cast آن ۲ مـی شود.) فرستنده B است پس در جدول آدرس فرستنده را B قرار می دهـیم. جـدول after adjustment طبق شکل بدست می آید که این جدول به A و B با یکدیگر ترکیب شـده و جـدول Combioecl را مـی دهند. اما در این جدول با رکورد های تکراری مواجه می شویم پس برای حـذف رکـورد هـایی کـه تکـراری هستند رکوردی را در جدول باقی می گذاریم که دارای هزینه کمتری می باشد. فرض کنیم که Router A می واهد جدول خود را ارسال کند. جدول به صورت زیر خواهد بود:



شکل ۱۴۹: Routing table Distribution

پس فقط کافی است که در مرحله اول ما فاز Setup را انجام دهیم، Update کردن آن به عهده خود Routerها می باشد.

در اینجا اگر یک Node اضافه شود، پس از مدتی دوباره همه جداول Update می شوند و مشکل را در روش استاتیک Shortest path وجود داشت، در اینجا وجود ندارد.

این منطق نیز منطق بسیار ساده و پرکاربردی می باشد. مثال کاربردی از این الگوریتم در پروتکلی به نام RIP استفاده می شود. پروتکل RIP ماکزیمم ۱۵ تا Router یا Hote دارد که در دهه ۸۰ استفاده می شد. Update آن هر ۳۰ ثانیه است. یعنی Router ها هر ۳۰ ثانیه جداول خود را برای همسایه ها ارسال می کنند. بعد از هر ۳ دقیقه اگر از Routerای جوابی نیاید، آن Router به صورت اتوماتیک از شبکه حذف می شود و آن پروتکل بیشتر در شبکه های کوچک استفاده می شود.

در این پروتکل شبکه ها را به شبکه های کوچکتری تقسیم می کنند و مستقل هستند و داخل هر شبکه یک پروتکل RIP است و بین این شبکه ها هم بی تفکر یابی برای مسیر یابی وجود دارد که ارتباطات بین

شبکه های کوچکتر را برقرار می کنند. شبکه های کوچک (خودمختار) بوسیله یک شبکه میانی به یکدیگر وصل می شوند و این موضوع باعث می شود که هم رکورد های جداول بالا نرود و هم ترافیک یک شبکه به شبکه دیگر افزایش نداشته باشد.

اگر شبکه ها را تقسیم کنیم به شبکه های کوچکتر، تعداد Routerها افزایش یافته و ترافیک زیادی خواهیم داشت پس شبکه را تا حدی کوچک می کنند که ۱۵ تا ۲۰ Router بیشتر داشته باشد. شبکه ها عدی و عد هم به همین ترتیب در این شبکه ها به وسیله Routerهای میانی به هم وصل می شوند. که به این Routerها، Gate way یا دروازه گفته می شود.

کنترل ازدحام:

رفت و آمد اگر از حدی بیشتر شود، ازدحام به وجود می آید و ما باید این ازدحام را کنترل کنیم.

یکی از روش های کنترل ازدحام، افزایش پهنای باند است ولی در بسیاری مواقع این روش امکانپذیر نیست. دیگر بحث فشرده سازی اطلاعات می باشد که اطلاعات را محدود می کنیم و یا روی تعادل اطلاعات شبکه مدیریت می کنیم که به برخی از Packete ها اجازه عبود داده شود و به بقیه خیر.

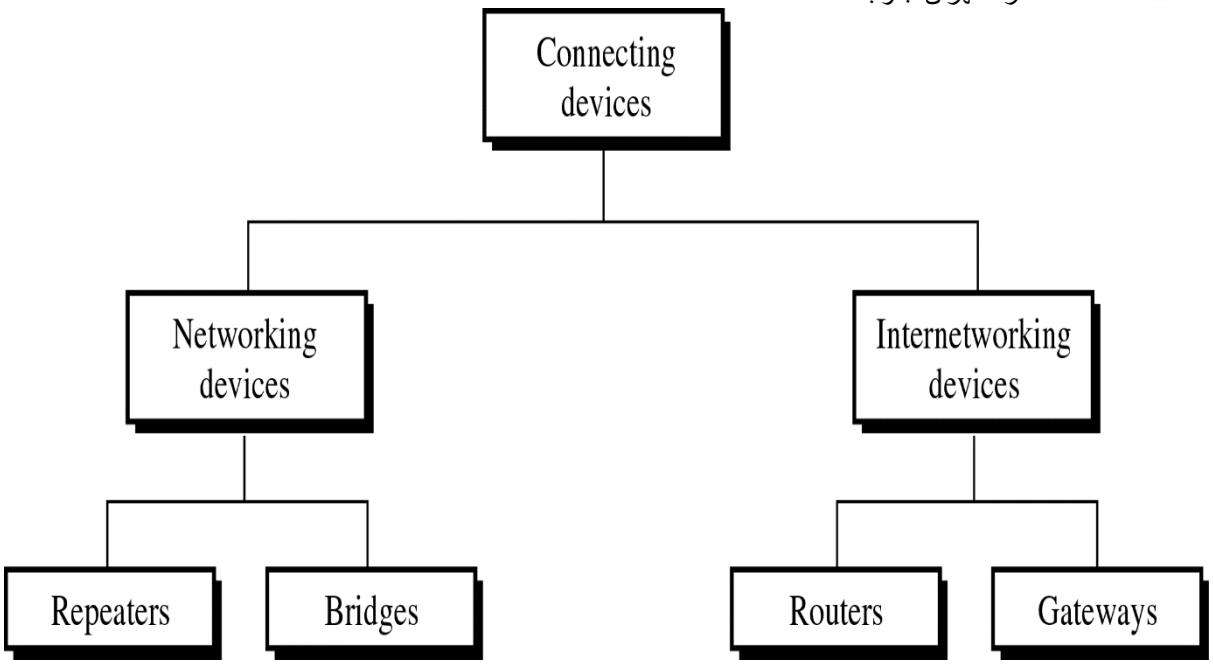
روش دیگر زمانی مطرح می شود که ترافیک بیش از همه زیاد و یا بافر گیرنده پر است و عملاً Packete ها خود انداخته می شوند.

روش دیگر استفاده از chalk packete یا Packete انسداد می باشد که در این روش وقتی ورودی های یک Router خیلی زیاد می شود، Packete انسداد را ارسال می کنند تا ورودی را کاهش داده و ترافیک پایین بیاید.

اگر قبلاً در مسیری ازدحام زیاد باشد و از روش Virtual Circuit استفاده می شود، عملاً ازدحام بالاتر رفته و ترافیک افزایش می یابد و یا TTL را کم و زیاد کردن را اگز کم کنیم به مقصد نمی رسد و اگر زیاد کنیم ازدحام بالا می رود.

Networking and Internetworking Device:

این بخش ابزار های سه لایه physical و Data link و Network را در داخل و خارج از شبکه معرفی می کند (ابزار های سخت افزاری برای ارتباطات بین شبکه ای)



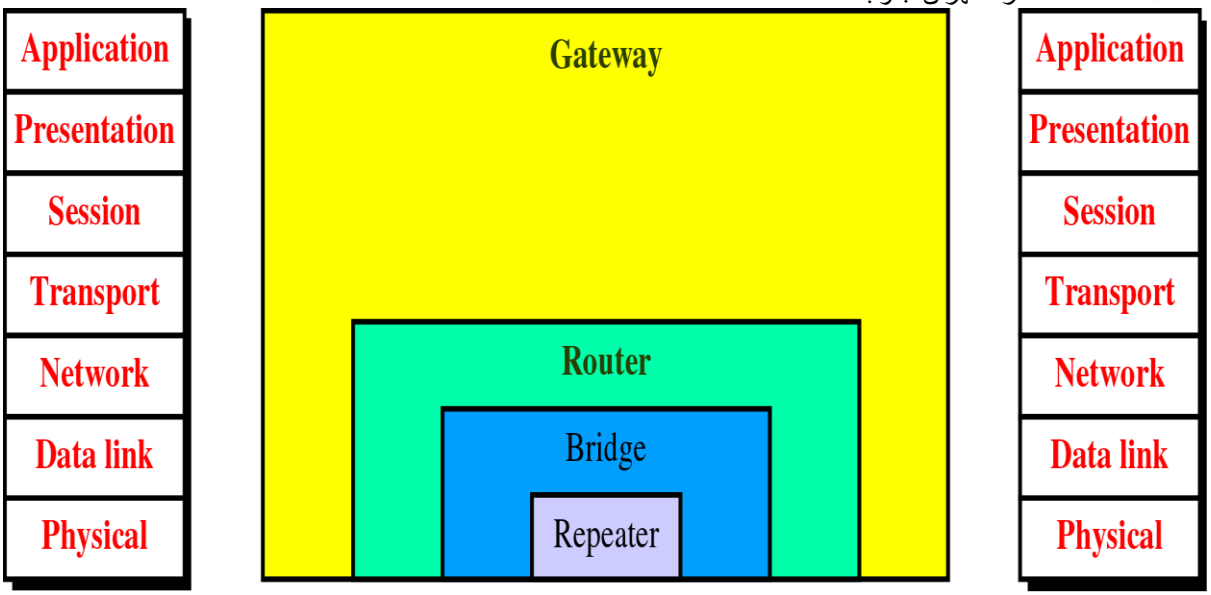
شکل ۱۵۲: وسایل ارتباط

Connecting Devices:

ایزار های Repeater و Bridge به عنوان ابزار های Networking و ابزار های Route و Gateway به عنوان ابزار های Internetworking در نظر گرفته می شوند.

محل قرار گیری Device ها در مدل OSI:

Repeater در لایه physical قرار دارد و همه کارهایی را لایه physical انجام می دهد Repeater هم انجام می دهد.



شکل ۱۵۳: Device های ارتباط و مدل OSI

Bridge در لایه Data link قرار دارد. عنصری که در لایه Data link قرار می گیرد، قطعاً لایه physical را هم دارد چون Data link کستقیماً به خط ارتباط ندارد و باید به لایه فیزیکی اطلاعات را روی خط قرار می دهد.

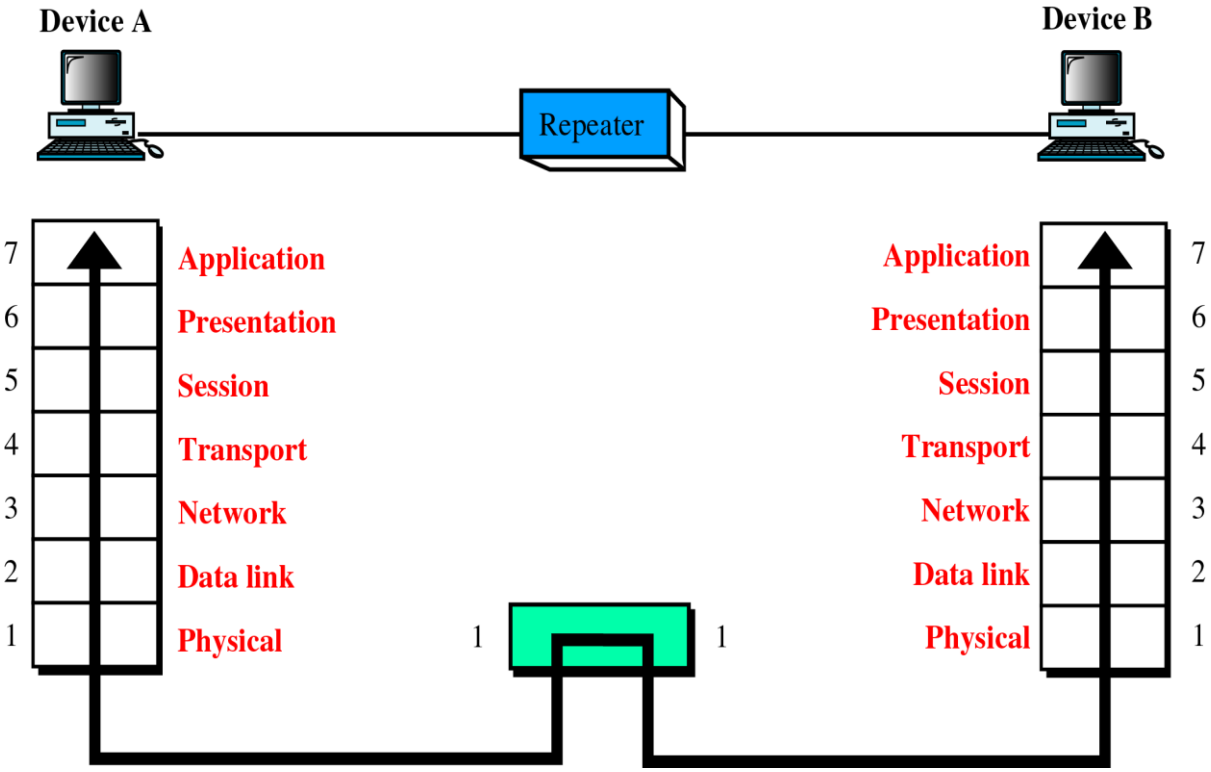
Router ها در لایه Network می باشند و قطعاً لایه physical دارد که سیگنال را از محیط می گیرد و Data link دارد تا یک Frame را انجام دهد و در Network عمل مسیریابی بر عهده Router می باشد. Gateway در بالاترین لایه قرار دارد و قطعاً تمام لایه های زیری را دارا می باشد (کل لایه ها).

Repeater:

در لایه physical برای اینکه دو شبکه را به یکدیگر وصل کنیم می توانیم از روی بنام Repeater استفاده کنیم. در انتقال اطلاعات با دو شبکه روبرو هستیم:

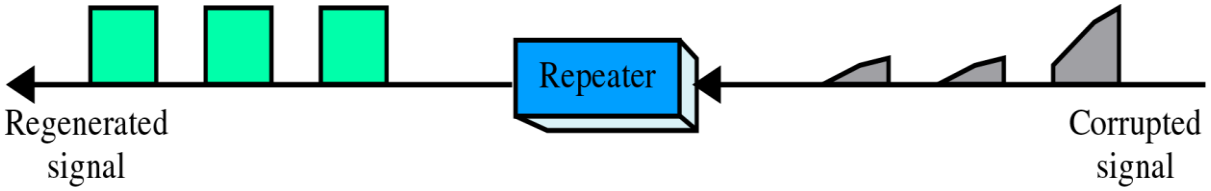
۱- تضعیف

۲- نویز

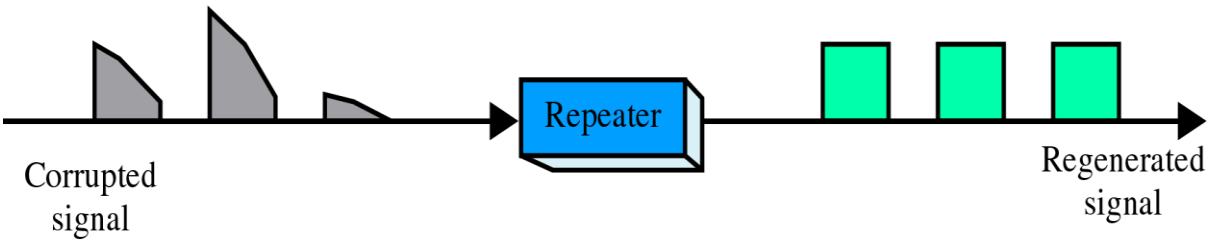


شکل ۱۵۴: تکرار کننده در مدل OSI

وقتی که سیگنال روی خط می رود، هنگام تضعیف شدن، مرتب دامنه اش کم می شود و به اصطلاح میرا می شود.



(a) Right-to-left transmission.



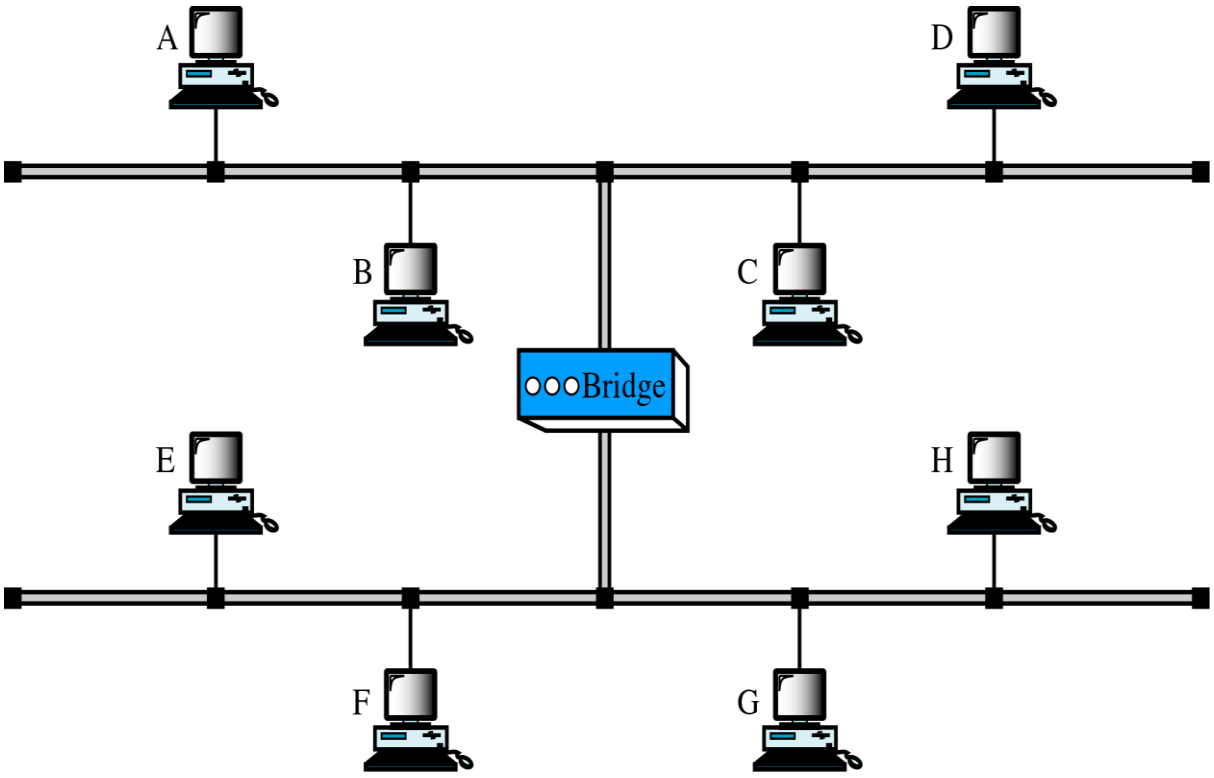
(b) Left-to-right transmission.

شکل ۱۵۵: Function of a repeater

بحث نویز سهم یکی دیگر از عوامل محدود کننده در گسترش شبکه می باشد.

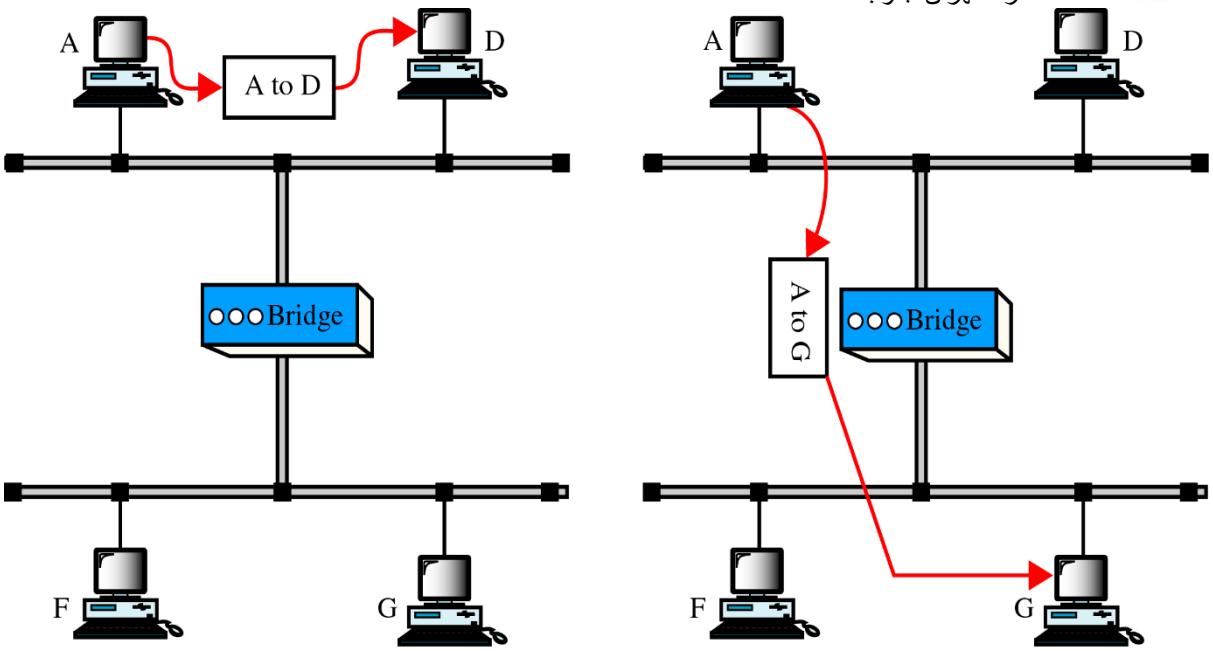
در هنگام تضعیف سیگنال ها، نیاز به ابزاری می باشد که اینکه سیگنال به عوامل تضعیف برسد، آنرا تقویت یا تکرار کند که Repeater این کار را انجام می دهند و از اینکه سیگنال و عوامل مقدار (از نظر ولتاژ) برسد، آنرا دوباره تولید و یا تکرار می کند. (Repeater تکرار کننده است نه تقویت کننده یا Amplifier وقتی که سیگنال تقویت می شود عملاً نویز را هم تقویت می کند ولی در اینجا توسط Repeater نویز از بین می رود.)

صفحه ی ۱۰۰



شکل ۱۵۸: Bridge

مثلاً در شکل ۱ Port به شبکه های A , B, C ,D وصل است و ۲ Part به شبکه های E, F, G, H وصل می باشد که آدرس کارت شبکه آن Nodeها در این جدول قرا می گیرد و به این ترتیب جدول Bridge، Setup می شود. در این مثال A می خواهد اطلاعاتی برای D بفرستد. وقتی که اطلاعات را روی خط قرار می دهد همه Nodeهای موجود در شبکه که در اینجا D و Bridge می باشند آنرا دریافت می کنند.



a. A packet from A to D

b. A packet from A to G

شکل ۱۵۹: عملیات یک Bridge

Bridge در جدول خود نگاه می کند و می بیند که D در همان شبکه می باشد پس اطلاعات را از خود عبور نمی دهد و این اطلاعات وارد شبکه بعدی نمی شود.

در حالت بعدی A به یک port وصل است و C به port دیگر. پس اطلاعات را عبور می دهد Simple Bridge به این صورت عمل می کند.

در واقع ما با Bridge شبکه ها را ایزوله می کنیم و هر اطلاعاتی را که می خواستیم از شبکه عبور می دهیم. و از این طریق ترافیک را کم می کنیم.

از Bridgeهایی نیز در شبکه هایی با معماری یکسان استفاده می گردد شبکه هایی را دارای لایه physical و Data link یکسان می باشد.

مشکل Simple Bridge ها این است که اگر کارت شبکه ای بسوزد و یا تغییر آدرس کارت شبکه داشته باشیم، مسئول شبکه باید دوباره جداول Bridgeها را Setup کند. برای حل این مشکل Learning Bridgeها را مطرحی کردند. Learning bridgeها خودشان آدرس ها را یاد می گیرند. یکبار اطلاعات را می فرستند و برای بقیه دفعات تمام آدرس ها را در اختیار خواهند داشت.در حالت عادی جدول Bridge خالی است. A می خواهد اطلاعات را برای D بفرستد. پس همه Nodeهای موجود در شبکه (در لایه D و Bridge) دریافت می کنند جدول Bridge خالی است پس نمی داند که D کدام Port آن وصل است. بنابراین یکبار اطلاعات را در کل شبکه می فرستد و آدرس ها را یاد می گیرد. بار دوم D می خواهد

اطلاعات را برای A بفرستد. A چون قبلاً به عنوان فرستنده اطلاعاتی را به Bridge فرستاده بود در جداول Bridge قرار گرفته است. پس Bridge می داند که A در همان شبکه است و اطلاعات را به شبکه بعدی منتقل نمی کند. D هم به عنوان فرستنده از همان شبکه می باشد پس آدرس D را هم در جدول خود ثبت می کند.

(پس Learning bridge های یکبار اطلاعات را می فرستند و تمام آدرس را یاد می گیرند.)

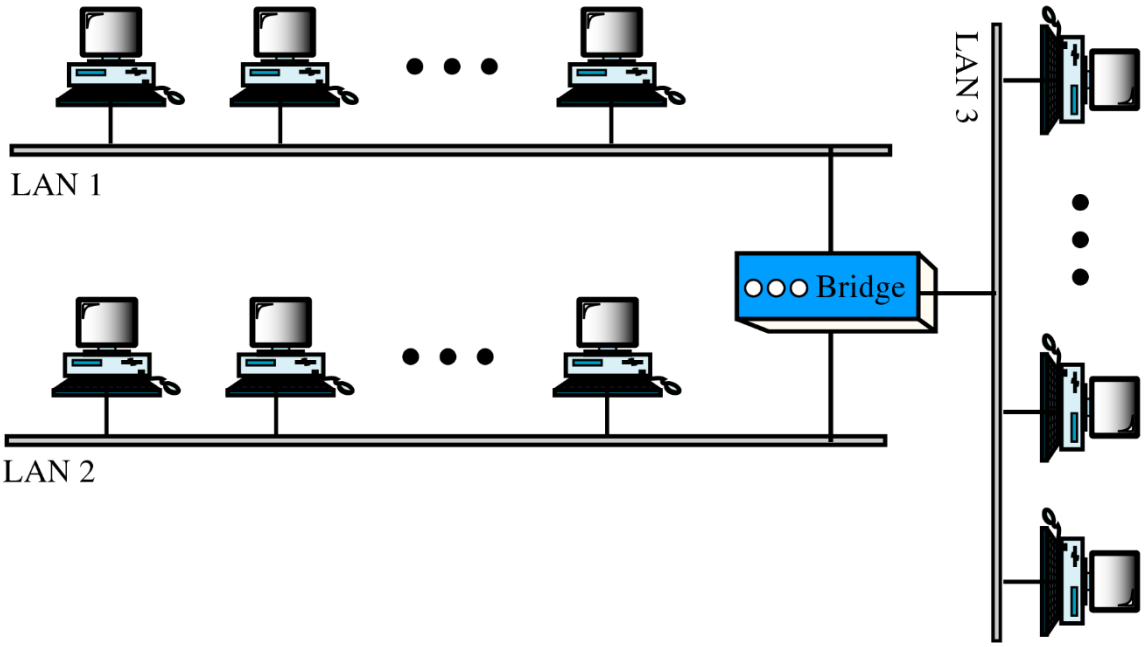
در این Bridgeها یگر شکل سوختن کارت شبکه وجود ندارد چون می توان گفت که اگر پس از مدتی کارت شبکه ای Update نشد، از جدول حذف شود و وقتی که دوباره شبکه وارد شد Bridge با فرستادن اطلاعات، آدرس را یاد می گیرد.

مشکل این Bridge ها در تشخیص Nodeها هنگام استفاده همزمان دو یا چند Bridge می باشد. فرض کنیم در شکل ۸-۲۱ ، دو شبکه A و B به یکدیگر متصل اند. زمانی که Bridge ها اطلاعات را برای یاد گرفتن آدرس ها می فرستند، مثلاً برای D، Bridgeها اطلاعات را برای یادگرفتن آدرس روی کل شبکه می فرستد که در اینجا Bridge روی شبکه A اطلاعات خود دریافت می کند و می رسد که اطلاعاتی آمده از فرستاده آن A است. پس آدرس A را روی Port شماره ۱ خود درج می کند در صورتیکه A در این Port وجود ندارد و جدول Bridge شبکه B به صورت اشتباه Setup می گردد.

برای جلوگیری از این اشتباه از Bridgeهایی استفاده می کنند نام Spaning Tree که یک Tree از Nodeهای خود درست می کنند که آن Loop به وجود نیاید. یعنی هر Bridge یک ساختار درختی از Nodeهای موجود در شبکه خود دارد تا از بروز این مشکل جلوگیری کند.

Multi Bridge:

این Bridgeها، Bridgeهایی هستند که چندین شبکه را به یکدیگر وصل می کنند که هم امکانات Simple Bridgeها و هم امکانات Learning Bridgeها را دارند یعنی می توانند به صورت Simple، Config شوند و یا به صورت Learning هر دو مکان برای آنها وجود دارد.



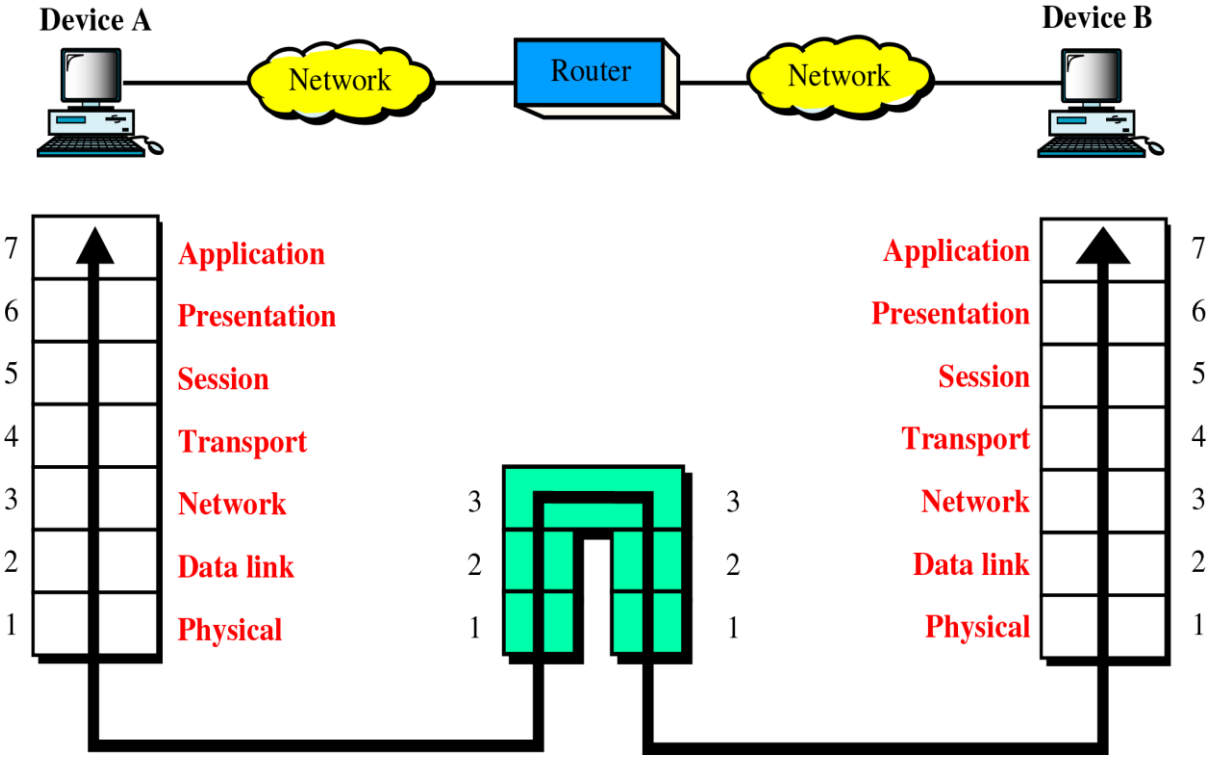
شکل ۱۶۰: Multiport Bridge

Internet working Device:

این Repeater و Bridge به عنوان ابزارهای داخل شبکه نام بردیم. از ابزارهایی بین شبکه ای می توان به Router و Gateway اشاره نمود.

Router:

وقتی بخواهیم شبکه هایی با معماری ها و ساختارهای متفاوت را به یکدیگر وصل کنیم که در لایه سوم (Network) می باشد.



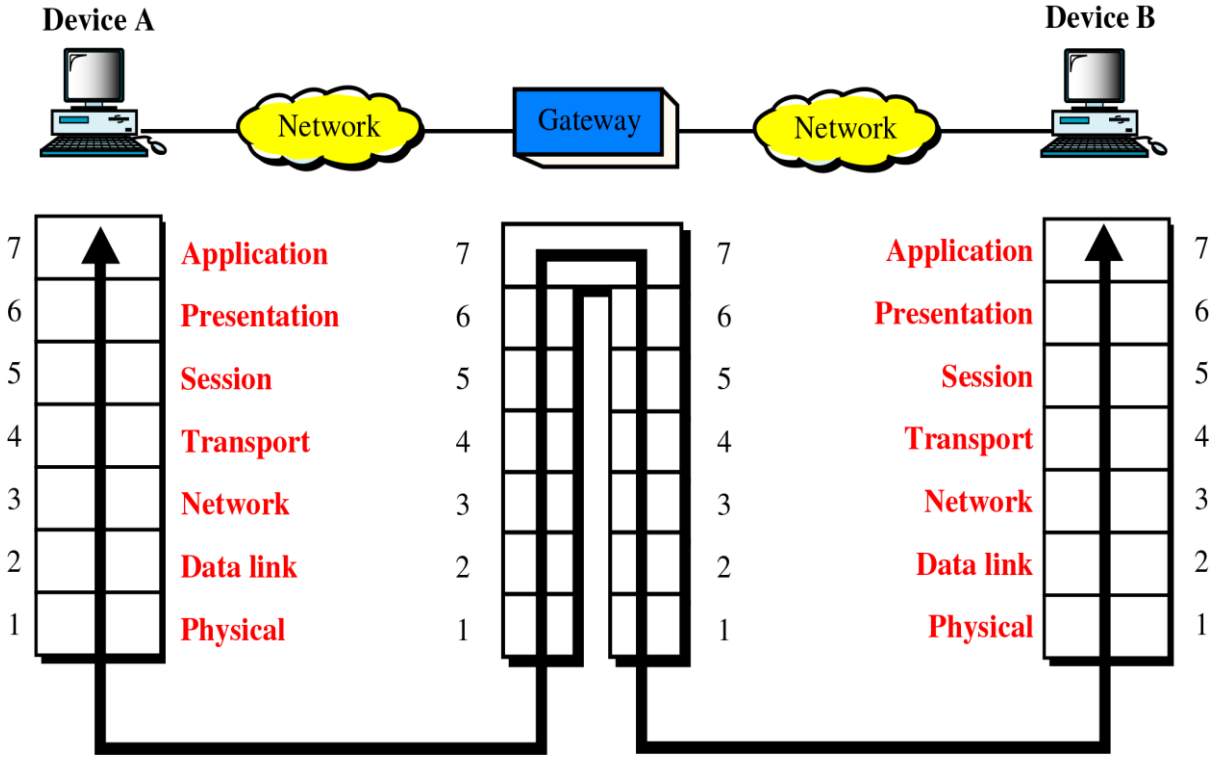
شکل ۱۶۱: Router در مدل OSI

همانطور که قبلاً گفته شد تحت مسیریابی، کنترل ازدحام و ... در Router مطرح می شود. شبکه هایی با معماری متفاوت را می توان از طریق Router به هم وصل کرد اما از قطر رو شکلی در لایه Network هم در پروتکل باشند.

در مبحث Repeater گفته شد که شبکه ها باید از نظر ساختاری کاملاً یکسان باشند حتی لایه physical آنها باید یکی باشد تا توان از Repeater استفاده کرد. در مبحث Bridge گفتیم که شبکه ها باید در لایه Data link شیشه هم باشند که قطعاً وقتی در Data link مشابهند Physical آنها هم یکی است در Router می گوییم که لایه physical و Data link می تواند در شبکه های مختلف متفاوت باشد اما در لایه Network هم با هم مشابتی نداشتند یعنی حتی با یکدیگر هم پروتکل هم نبودند و نه در لایه های Transport و Session و ... مشابه نبوده اند از یک Device دیگر به نام Gateway استفاده نمود.

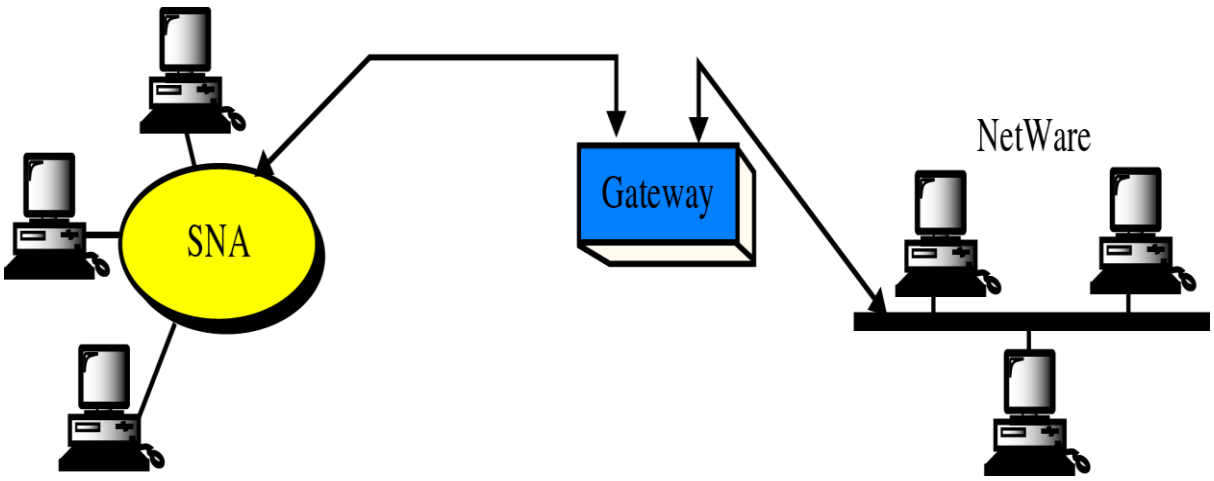
Gateway:

Gateway اصطلاحاً Protocal convertor گفته می شود که از کل معماری هر دو شبکه را می خواهند با یکدیگر تبادل اطلاعات کنند با خبر است و عمل تبدیل پروتکل را انجام می دهد.



شکل ۱۶۲: Gateway در مدل OSI

در این مثال، دو شبکه کاملاً متفاوت مانند SNA و Network از طریق یک Gateway به یکدیگر متصل شده اند.



شکل ۱۶۳: Gateway

Gateway از یک طرف جزیی از شبکه SNA و از طرف دیگر جزیی از شبکه Network می باشد. پس قطعاً با هر دو شبکه به طور کامل آشنا می باشد. در اینجا دیگر Router نمی تواند ارتباط دو شبکه را با یکدیگر برقرار کند و فقط Gateway می تواند این کار را انجام دهد.

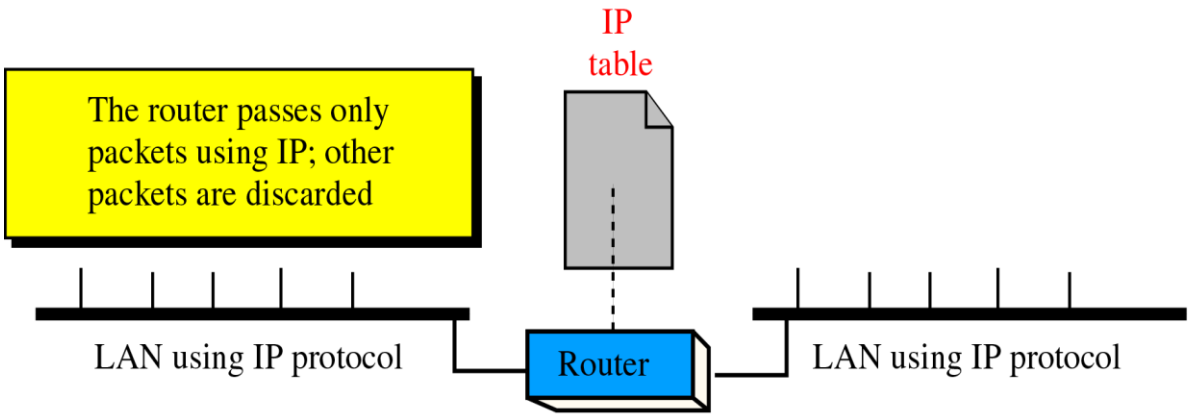
Other Device:

از وسیله های ارتباطی دیگر می توان اشاره کرد:

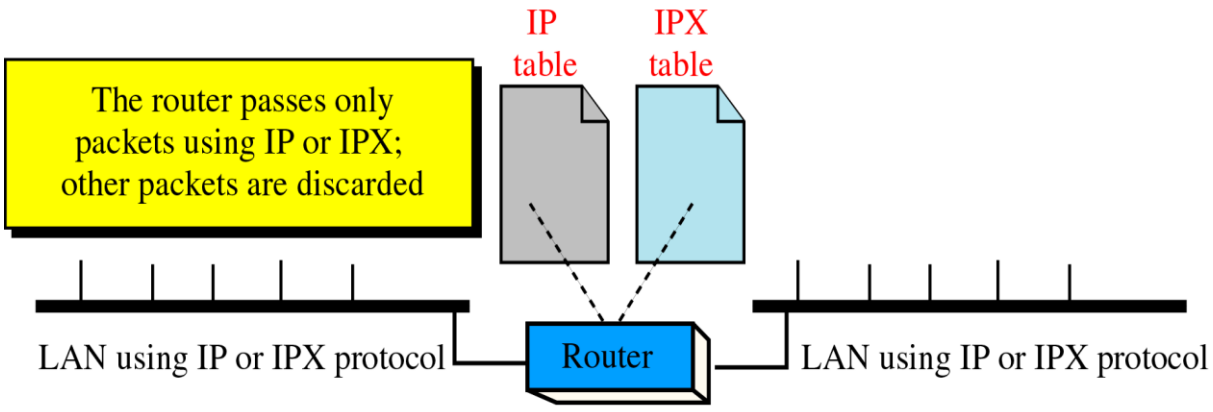
۱- Multiprotocol routers
۲- Brouters
۳- Switches

Multiprotocol Routers:

گفتیم که Router در لایه Network قرار دارد و شبکه های از ساختار متفاوت را به یکدیگر وصل می کند.



a. Single-protocol router



b. Multiprotocol router

شکل ۱۶۴: Single Protocol versus Multiprotocol Router

IP هم یک پروتکل بسیار معروف در لایه Network است.

اگر به Packete ای وارد شود از پروتکل IP استفاده کرده باشد، قطعا Router عمل مسیریابی را انجام می دهد ولی در غیر این صورت آنرا Discard می کند. چون پروتکل آن با پروتکل لایه Network سازگاری ندارد این موضوع در Routerهای Singel protocol دیده می شود.

در مقابل Multi protocol Routers هستند را در این Routerها، IPX هم به عنوان یک پروتکل معتبر در لایه Network مطرح است (IPX مربوط به شرکت Novell می باشد.)

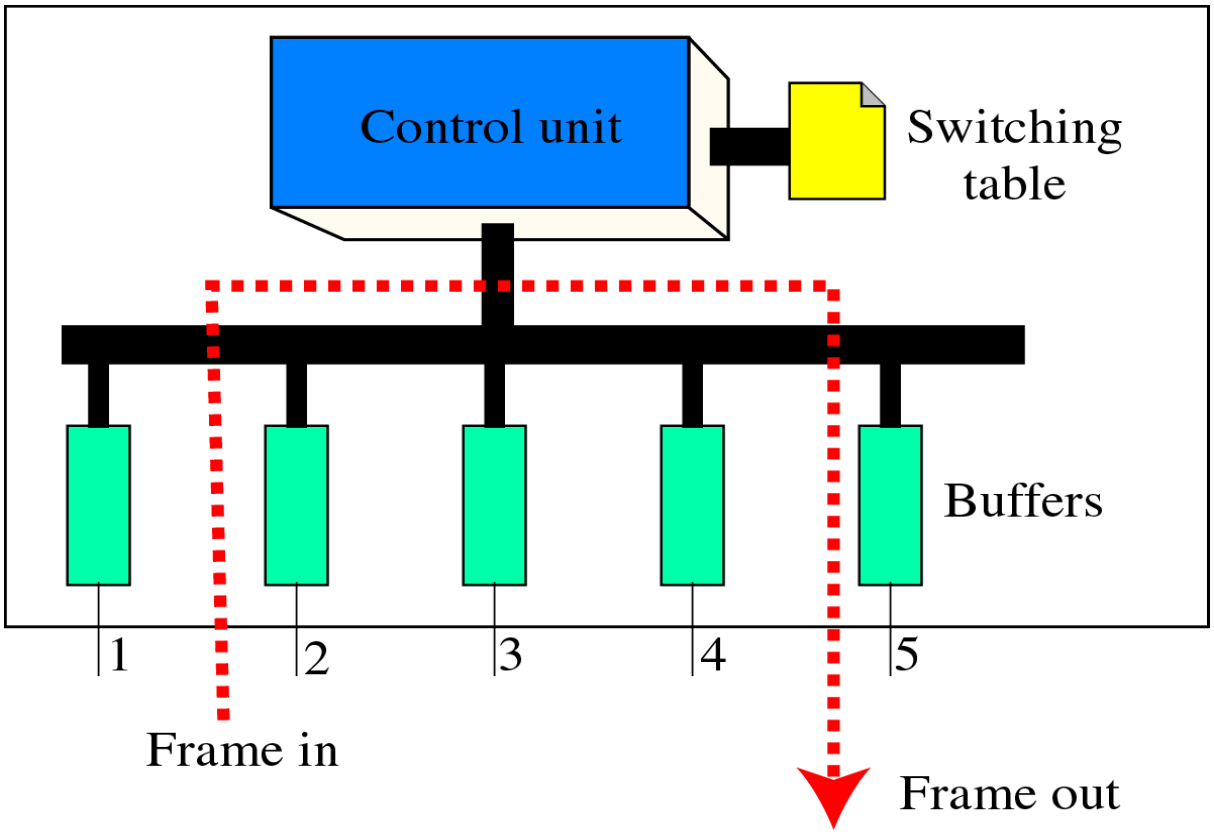
که هر پروتکل IX و IPX را پشتیبانی می کند. در واقع در Multi protocol Routers از چندین پروتکل می توان استفاده نمود.

صفحه ۱۰۵

Switch:

یکی دیگر از ابزار های بسیار رایج در داخل شبکه Switch است.

ما نیاز به وسیله ای داریم که اطلاعات را روی کل شبکه پخش نکند و فقط به شبکه مورد نظر ما اطلاعات را برساند که به آن Switching لایه ۲ گفته می شود و دقیقاً مثل Bridge عمل می کند و برخی اطلاعات را عبور داده و برخی را عبور نمی دهد و می داند که چه اطلاعاتی را روی کدام Port قرار دهد.



شکل ۱۶۶: Switch

فرض می کنیم که در شکل بالا یک Hub داشته باشیم وقتی که ۲ می خواهد اطلاعات را به S بفرستد، Hob وسیله ای قرار می دهیم که اطلاعات را فقط به ۵ فرستد نه به همه Nodeها و قطعاً چنین Divice ای در لایه Physical معنی ندارد و باید در لایه Data link باشد که به آن Switch گفته می شود.

اگر شبکه ای داشته باشیم که ۱۰ ایستگاه و از Hub استفاده می کند، خدمات آن شبکه تقسیم بر ۱۰ می شود یعنی هر ایستگاه حق استفاده از یک دهم خط را دارد و یک دهم از بخش زمانی را می تواند در اختیار داشته باشد. اما در Switch این ویژگی وجود دارد که بطور همزمان نصف ایستگاه ها می توانند با ۳ ارتباط داشته باشند.

وقتی از Switch استفاده می کنیم، زمانیکه بخواهیم ۲ اطلاعات را به ۵ بفرستد، ابتدا اطلاعات ۲ وارد یک Control unit می شود و توسط جدول Switching مقصد آن تشخیص داده می شود و اطلاعات دقیقاً روی ۵ قرار می گیرد و روی ایستگاه های دیگر کپی نمی شود. در این زمان، دیگران خط را آزاد می بینند و می توانند اطلاعات خود را ارسال کنند.

Switch های دیگر وجود دارد به نام Store & Forward Switch که در این سوئیچ ها هر Port یک Baffer دارد. وقتی اطلاعات وارد این Portها می شود، ابتدا بافر می شود و بعد چک می شود. که آیا Frameای که آمده سالم است یا نه. اگر خراب باشد Discard می شود.

گفتیم بحث سوئیچ ها در شبکه اترنت مطرح می با شد که شبکه اترنت Flow Control ندارد و به محض اینکه لایه Data Link تشخیص می دهد Frame خراب است آن را Discard می کند و از طریق این سوئیچ ها سالم و یا ناسالم بودن Frame تشخیص داده می شود. این کار باعث می شود Performance (کارایی) خیلی افزایش پیدا کند چون Colligion کم می شود.

گفتیم که شبکه ها ، اترنت هستند. مشکل شبکه های اترنت هم Colligion آنها بود. وقتی که Frameای به صورت خراب در ارسال می شود باعث می شود که Colligion افزایش یابد. پس در این سوئیچ ها Colligion کاهش می یابد و با کاهش آن Performance افزایش می یابد.

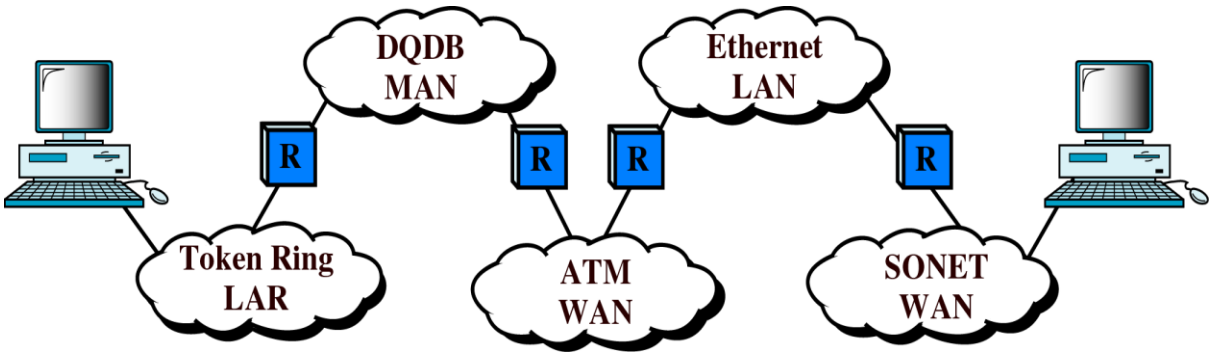
سوئیچ های دیگری هستند به نام کاردترد که در این سوئیچ ها فقط چند سایت اول Frame چک می شود و آدرس مقصد را نگاه می کند و بعد فاصله روس خروجی قرار می دهد.

از نظر سرعتی چون پردازشی انجام نمی دهد، بسیار سریعتر است ولی دارای هزینه کمتری می باشد. چون کشف خطا خودش هزینه بردار است و وقتی نیازی به کشف خطا نباشد سپس هزینه پایین می آید. (سخت افزار مورد نیاز کشف خطا را ندارد.)

پس از سوئیچ ها برای ایزوله کردن ترافیک ها و بالا بردن Performance شبکه استفاده می شود.

Transport Layer:

لایه Network لایه امنی برای ما نیست و ممکن است Packet در انتقال اطلاعات بین Router از بین برود و اشتباه برسد. لایه Network لایه مطمئنی نیست. لایه Data link داخل شبکه را کنترل کرد. ما مشکلی از این نظر در شبکه نداریم. مشکل در خارج از شبکه است، به همین دلیل لایه Transport مطرح می شود؟؟؟ مدل OSI لایه Transport می باشد.

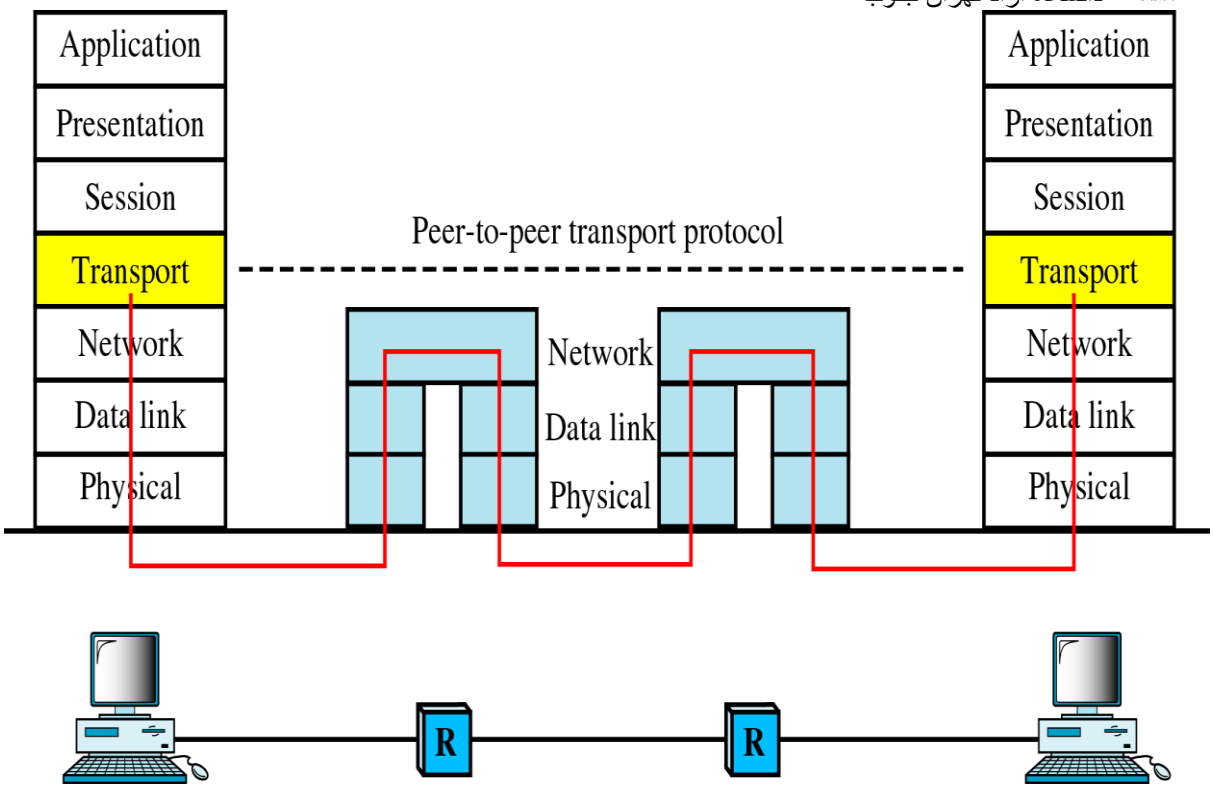


شکل ۱۶۷: an Internet

همانطور که در شکل نشان داده شده، یک شبکه ممکن است مجموعه ای از شبکه های Ethernet، Sonet Token Ring و ... باشد. Routerها شبکه ها را به معماری های متفاوت به هم وصل کرده اند.

همانطور که گفته شد ما در داخل شبکه مشکل نداریم. بحث کنترل خطا، چک Frame و ... همه در داخل شبکه صورت می گیرد. مشکل بر سر Router می باشد که دومی راه ممکن است اطلاعات از بین برود و یا دچار مشکل شود و خراب برسد به همین دلیل لایه Network لایه مطمئنی نمی باشد.

لایه Transport چهارمین لایه از مدل OSI می باشد و لایه Transport به بالا به صورت نرم افزاری یاده می شود و مطمئناً وقتی که Transport یک ماشین می خواهد با Transport ماشین دیگر ارتباط برقرار کند، باید هم Protocol باشند تا بتوانند با یکدیگر تبادل اطلاعات انجام دهند.



شکل ۱۶۸: Transport Layer

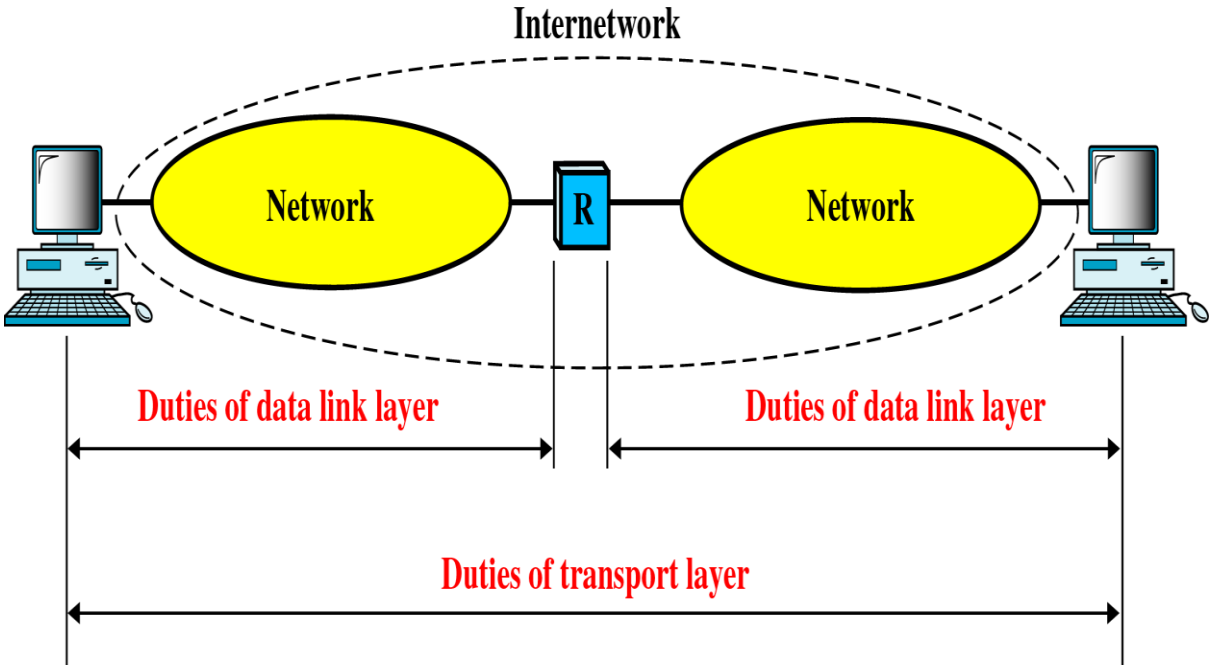
گفته می شود وظیفه لایه Transport، نظارت بر تحویل اطلاعات از Application در ماشین مبدا Application در ماشین مقصد می باشد.

لایه Network اطلاعات را از ماشین به ماشین تحویل می دهد ولی هیچ نظارتی روی آن ندارد. تنها چیزی که در لایه Transport اضافه شده، نظارت و تحویل اطلاعات از Application تا Application می باشد.

شباهت زیادی بین لایه Data link و لایه Transport وجود دارد.

مقایسه لایه Transport و Data link:

طبق شکل دو شبکه را توسط یک Router به یکدیگر وصل کرده ایم.



شکل ۱۶۹: مقایسه لایه Transport و Data link

وظایف لایه Data link از قبیل:

Flow control, Error Control, Line Dicipline و ... در داخل شبکه صورت می گیرد.

سپس در داخل هر دو شبکه، Data link کار کنترل را انجام می دهد ولی در خارج از شبکه نمی تواند کاری انجام دهد چون بر لایه بالاتر از فروش نمی تواند نظارتی داشته باشد. پس در خارج از شبکه ها، Router را لایه Transport را روی Network قرار می گیرد کنترل می کند.

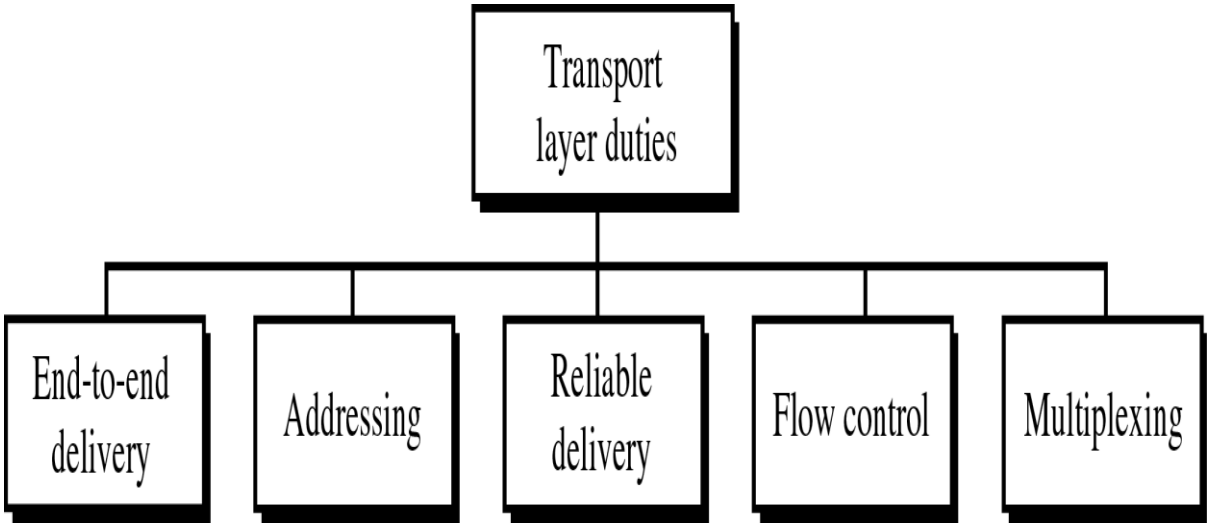
وقتی که لایه physical اطلاعات را به لایه Data link تحویل می دهد، لایه Data link کنترل ها مورد نیاز از قبیل کشف خطا، Flow control و ... را روی آن انجام می دهد و اطلاعات کنترلی لازم را به آن اضافه نموده و لایه Network تحویل می دهد. لایه Network وقتی که اطلاعات را دریافت می کند، اگر اطلاعات را به صورت ناقص دریافت کند و یا اشتباهی در آنها رخ داده باشد، تشخیص نخواهد داد چون فقط وظیفه مسیریابی را به عهده دارد و نه کشف خطا. به همین دلیل همان اطلاعات نادرست را با اضافه کردن یکسری اطلاعات کنترلی به گیرنده می فرستد. در گیرنده لایه Data link چک خطا را اینجا می دهد. (با توجه به اطلاعات کنترلی و چون خطایی مشاهده نمی کند، آن را به Network می دهد. بنابراین اطلاعات نادرست ارسال و دریافت می شود. لایه Transport برای نظارت بر این مشکل مطرح شد که اگر خطایی در لایه Network صورت گرفت آنرا تصحیح کند.

*چرا لایه Transport نظارت از Application تا Application را بر عهده گرفت؟

پروتکل های استاندارد معروف قبل TCP/IP، SPXITPX و ... که از مدل OSI پیروی می کند. مثلاً از لایه Transport، Application قرار دارد که گفته می شود Transport بر آن در تحویل اطلاعات نظارت می کند.

وظایف لایه Transport:

پنج وظیفه مهم بر عهده لایه Transport می باشد:



شکل ۱۷۰: وظایف لایه Transport

- End-to-end Delivery
- Addressing
- Reliable Delivery
- Flow Control
- Multiplexing

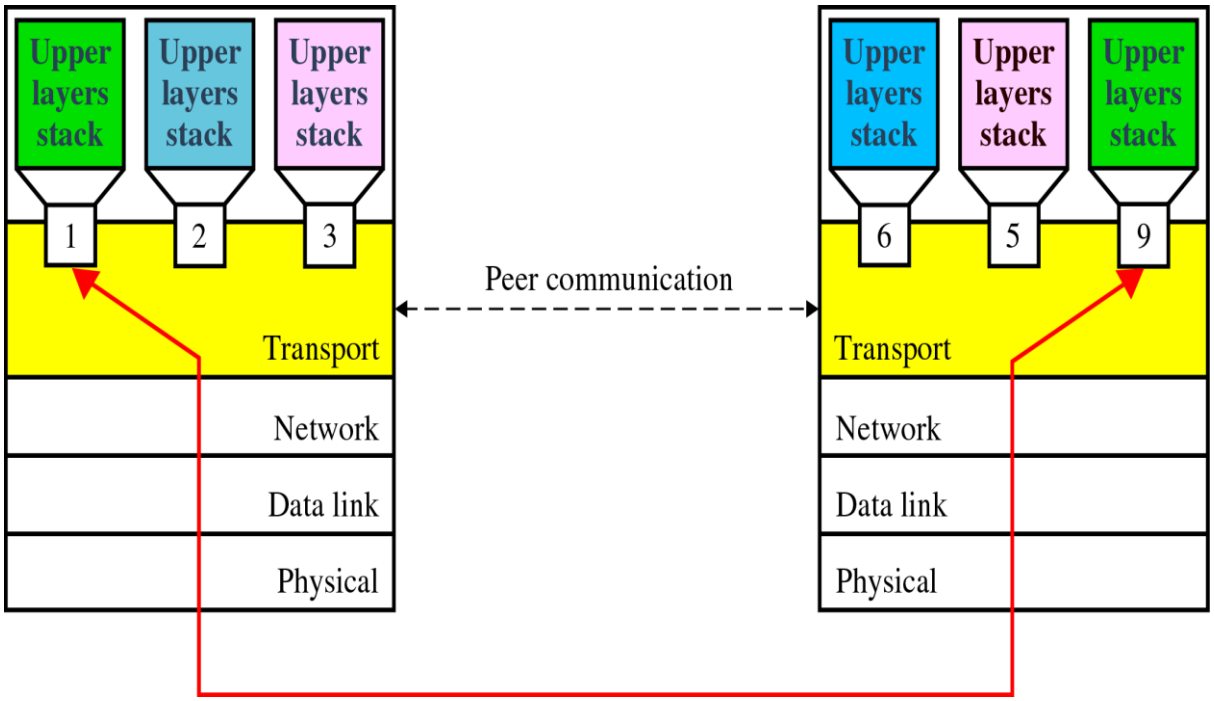
End-to-end Delivery:

به نظارت بر کل اطلاعات فرستاده شده از Application در ماشین مبدا به Application در ماشین مقصد End-to-end Delivery گفته می شود. چون ما یک بسته بزرگ اطلاعاتی را تبدیل به Packetهایی می کنیم که به طور جداگانه روی خط قرار می گیرند. مبحث end-to-end Delivery به

نظارت بر تمام این Packet ها از زمان ارسال اطلاعات تا رسیدن به لایه این Application گیرنده اشاره می کنیم.

Addressing

همانطور که بحث آدرس دهی در لایه Data link به عنوان آدرس کارت شبکه و در لایه Network به عنوان آدرس لاجیکی مطرح بود، در اینجا ه بحث آدرس دهی را خواهیم داشت. وقتی که لایه Transport می خواهد اطلاعاتی ارسال کند باید آدرس Application را داشته باشد در غیر این صورت Packet در مسیر صحیح هدایت نخواهد شد. به همین دلیل در لایه Transport هم نیاز به آدرس دهی حس می شود. آدرسی که در لایه Network مطرح می شود را اصطلاحاً IP و آدرسی را که در لایه Transport مطرح می شود را اصطلاحاً Port می گویند.



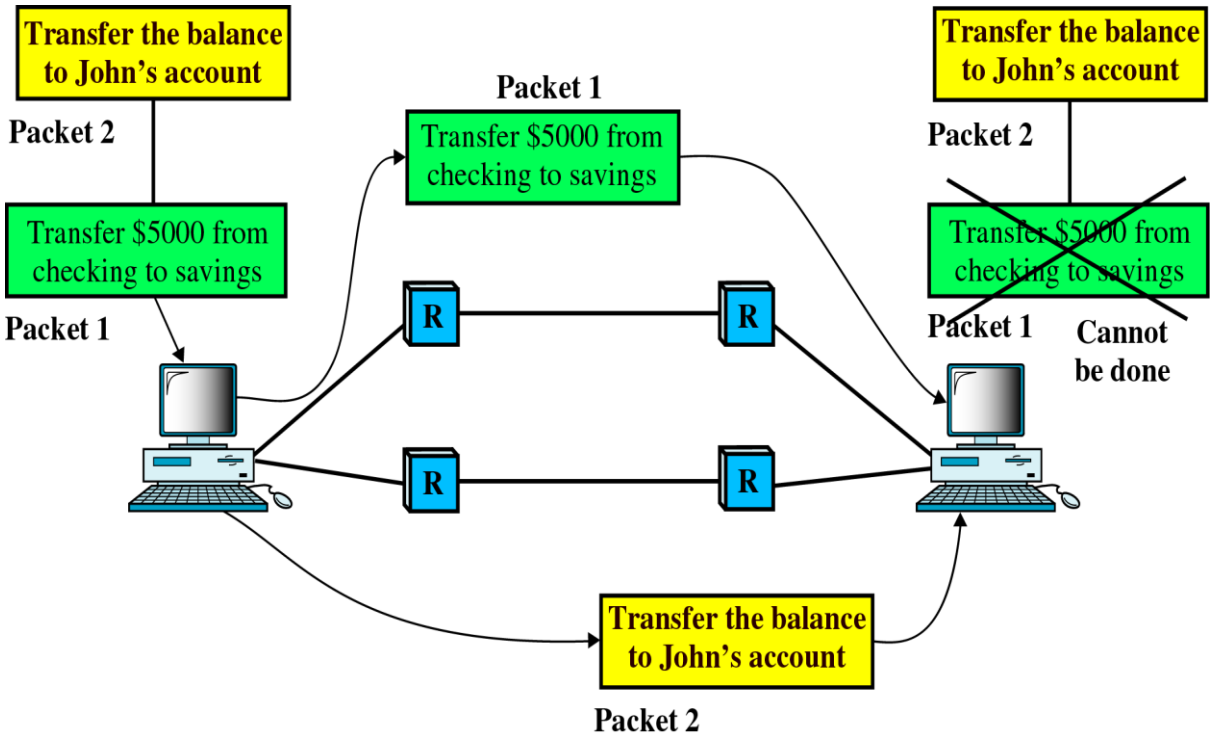
شکل ۱۷۱: Addressing

در لایه Network که با آدرس لاجیکی کار می کند، انتقال اطلاعات از ماشین به ماشین صورت می گیرد. اما زمانی که ما می خواهیم اطلاعات را از Application به Application تحویل دهیم، باید آدرس Application یا Port را داشته باشیم. به تلفیق یک

اینجا متوجه خطایی که در Router رخ داده نخواهد شد. پس برای کنترل Router ها که توسط لایه Data link نمی توان آنها را کنترل کرد، لایه Transport مطرح می شود تا این قسمت را کنترل کند.

Sequence Control:

طبق این مثال، وقتی که حجم اطلاعات ما زیاد باشد، در لایه Transport تبدیل به ۲ بخش می شود. که اینجا Packet ۱ و Packet ۲ ساخته شده است و واحد اطلاعاتی در لایه Transport و Data link ها را وقتی بسته بندی کرد تحویل لایه شبکه می دهد.



شکل ۱۷۴: Sequence Control

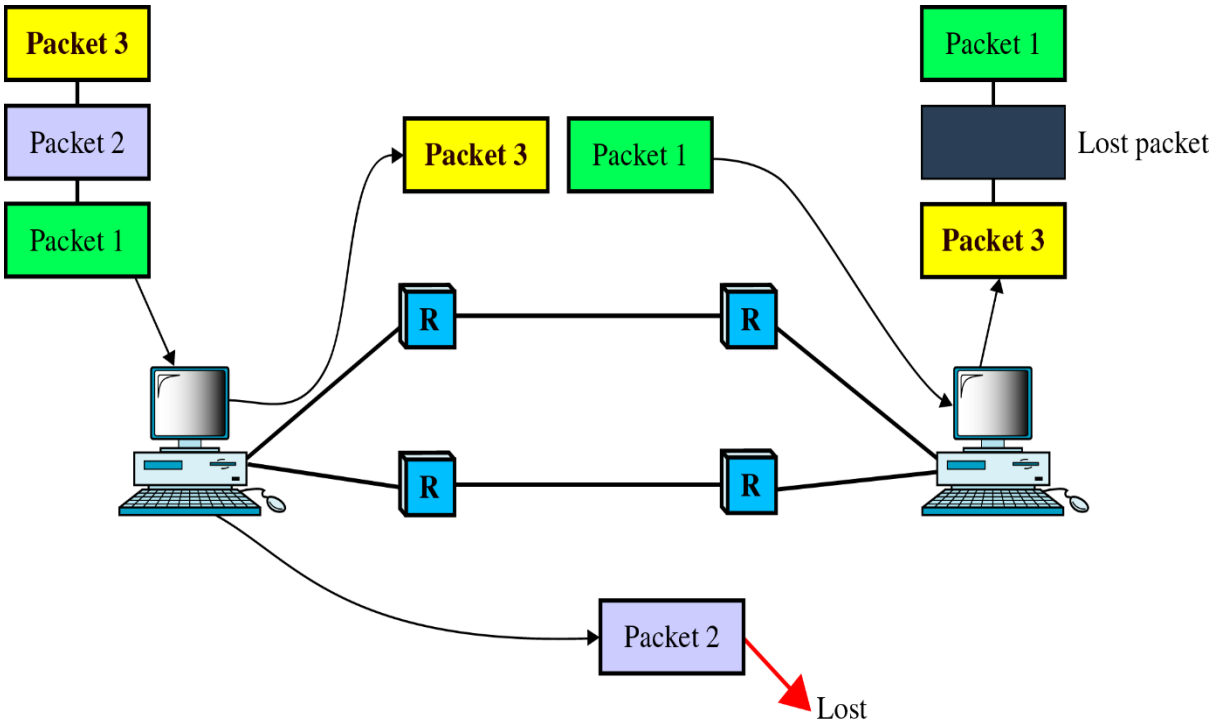
لایه Network هم دوباره بسته بندی خاص خودش را روی اطلاعات دریافتی انجام می دهد و Packetها را می سازد. این Packetها در لایه Network با مسیرهای متفاوت مواجه می شوند که در این صورت امکان جا به جا رسیدن Packetها وجود دارد وقتی که Packetها جا به جا می رسند نیز امکان رخ داد خطا خیلی زیاد می شود و ممکن است Packet مربوط اصلاً اجرا نشود چون نیاز به اطلاعات Packet قبلی داشته که آن Packet هنوز نرسیده است.

این وظیفه لایه Network نیست که ترتیب رسیدن Frameها را کنترل کند و لایه Transport باید این کار را انجام دهد. به این ترتیب که برای هر واحد اطلاعاتی (Data unit) یک شماره در نظر بگیرد در

برگیرنده کافیست که بر مبنای این شماره ها تشخیص دهد که آیا اطلاعات جا به جا رسیده یا نه و یـا گـم شده یا سالم رسیده است. این موضوع برای بعضی از کاربرد ها بسیار حیاتی می باشد و اگر Packetهـا جـا به جا برسند ممکن است خطای بزرگی رخ دهد.

Loss Control:

لایه Transport می خواهد اطلاعاتی را بفرستد و چون حجم اطلاعات زیاد است آنها را بـه ۳ بخـش مـی شکند که در Packet ها جای بگیرند. آنها را به لایه Network می دهد و لایه Network هم هر یک را از طریق یک مسیر حمل می کند. Packetها جا به جا می رسند و یکی از آنها هم در بین راه گم می شود. در گیرنده از طریق همان شماره ای که برای هر واحد اطلاعاتی در نظر گـرفتیم، (sequence number) چک می کند تا ببیند که ترتیب Packetها رعایت شده یا نه.

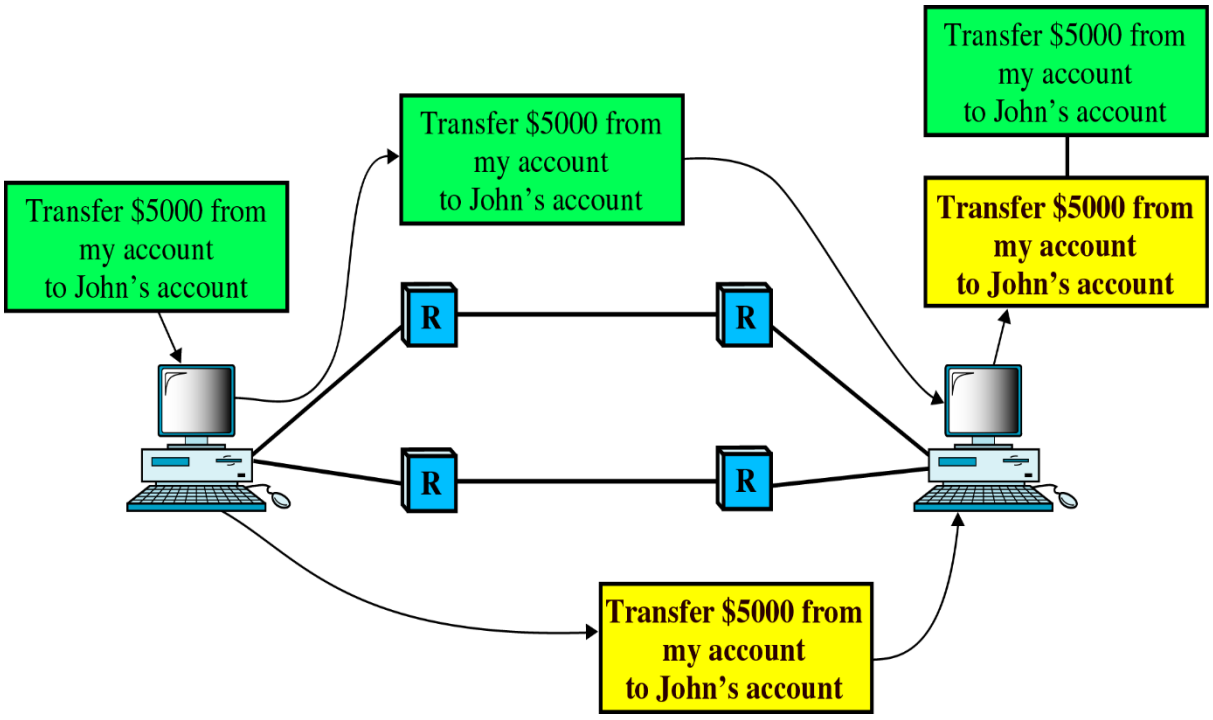


شکل ۱۷۵: Loss Control

پس متوجه شدیم که در واحد های اطلاعاتی (data unit) باید یـک بیـت بـرای Sequence number هم در نظر گرفته شود. چون اگر حجم واحد های اطلاعاتی زیاد باشد، باید آنهـا را در واحـد هـای کـوچکتر بشکنیم و بنابراین باید به هر یک از این واحد ها یک شماره ترتیب بدهیم که این شماره ترتیب یک فیلد در Data unit است.

Duplication Control:

لایه Transport باید قادر باشد بتواند اطلاعاتی را که به صورت دوبل (چند تـایی) مـی رسـند، Discard کند.

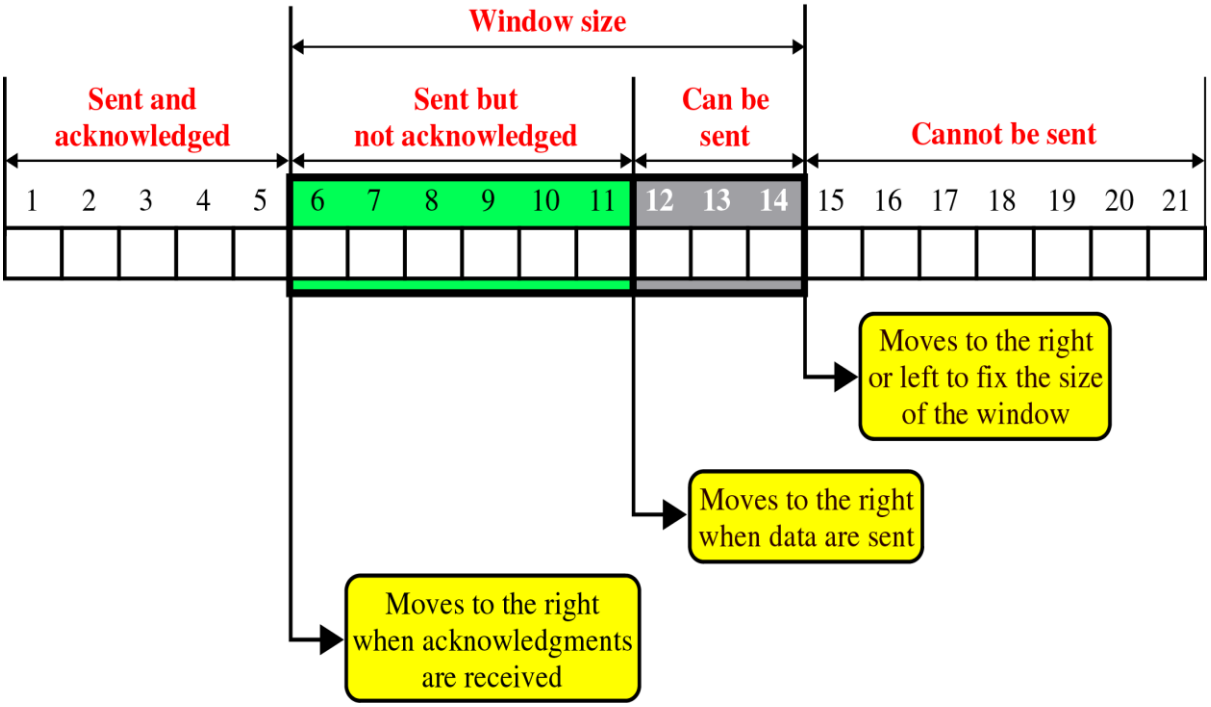


شکل ۱۷۶: Duplication Control

اگر یک Packet دوباره تولید شد و از مسیر های متفاوت به گیرنده رسید. گیرنده باید Packet تکراری را Discard کند که این کار هم از طریق همان Sequenceهایی در نظر گرفته شد به عنـوان یـک فیلـد در Data unit صورت می گیرد و Packetهای تکراری را حذف می کند.

Flow Control:

باعث Flow Control و Error Control را در لایه Data link داشتیم و حالا در لایه Transport هم داریم. وقتی واحد های اطلاعاتی گم می شود باید ACK یا NAK فرستاده شود. پـس همـان بحـث، پنجـره های نفرون را در لایه Transport هم داریم.

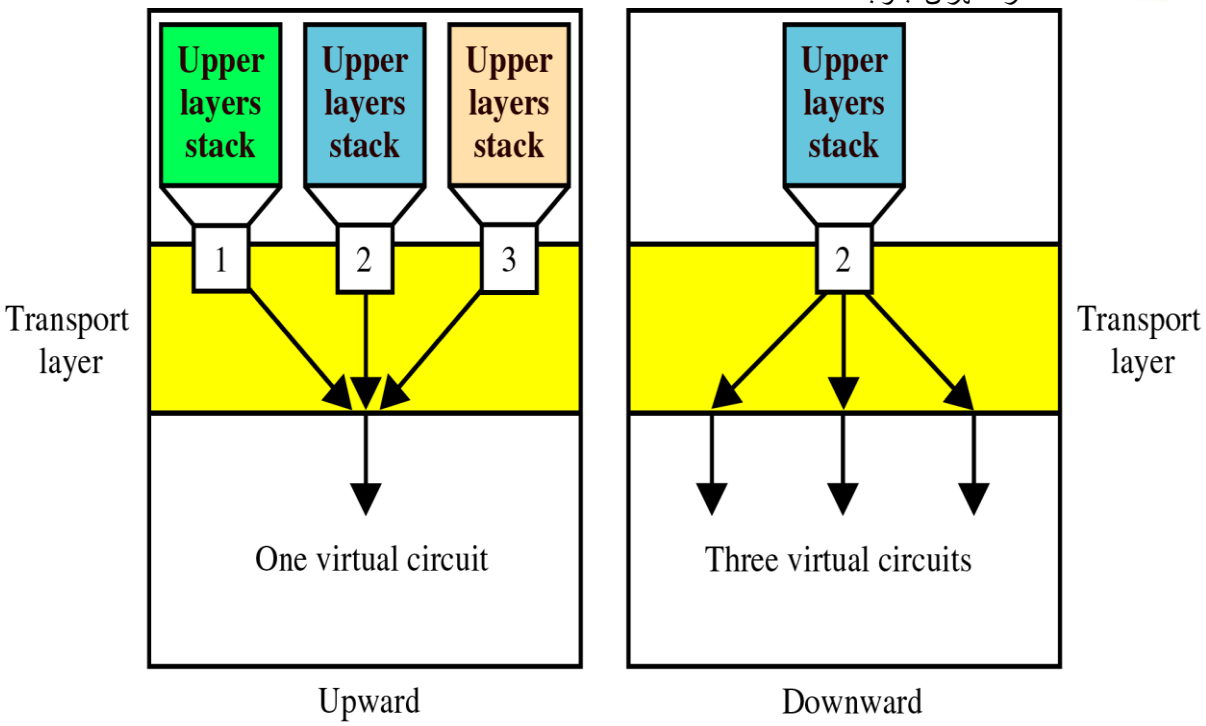


شکل ۱۷۷: Sliding Window

بحث Sliding win در لایه Transport دقیقاً مشابه لایه Data link می باشد. وقتی فرستنده، اطلاعات را می فرستند و گیرنده ACK را ارسال می کند، پنجره سمت چپ به سمت جلو حرکت کرده و پنجره سمت راست هم به همان مقدار به جلو حرکت می کند و ...

Multiplexing:

همان طور که در شکل دیده می شود، چندین Application از طریق یک Connection یا واحد اطلاعاتی شروع به ارسال اطلاعات می کنند. چندین Connection برقرار نکرده اند و تنها از طریق یک Connection این ارتباط برقرار می شود. به دلیل این که در لایه Transport، Virtuad Circuit دارای پهنای باند زیادی می باشد و می تواند اطلاعات زیادی را حمل کند و چندین Application در قالب یک Connection کار خود را انجام می دهند تا سرعت آنها افزایش یابد.

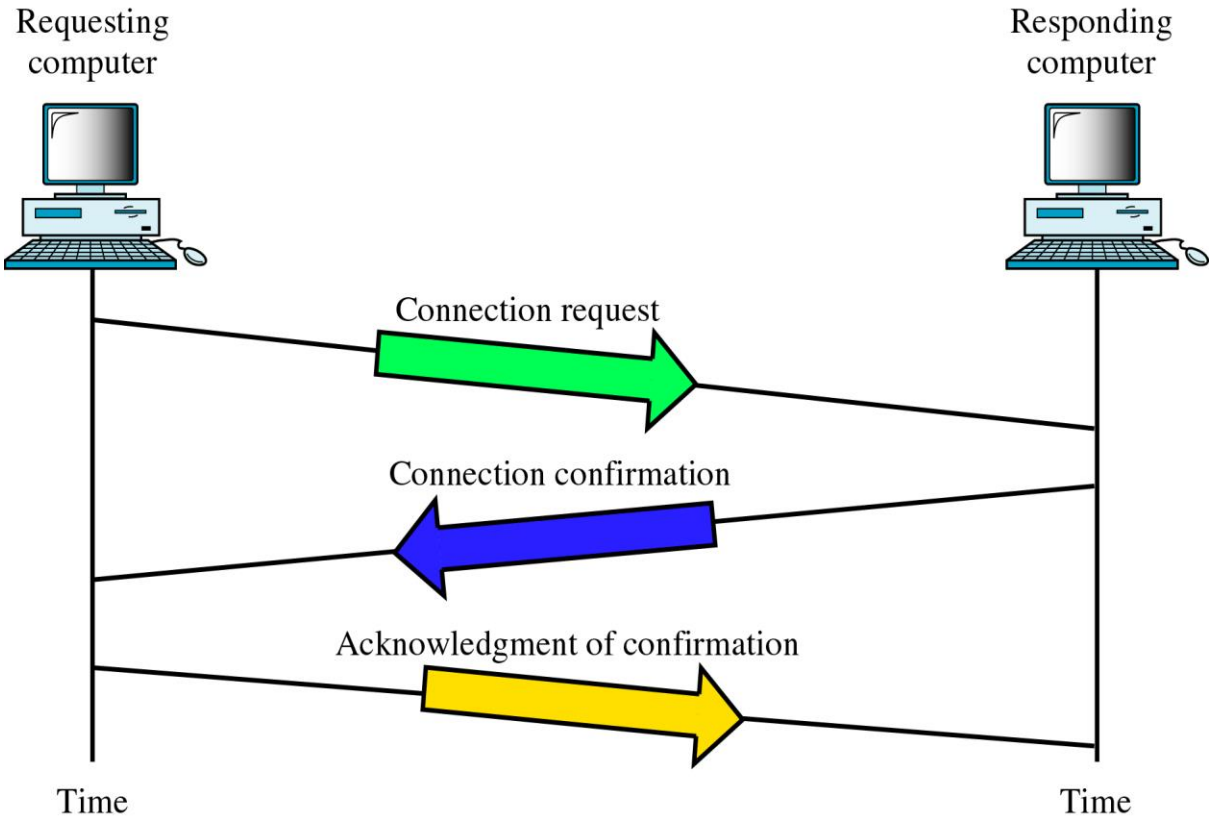


شکل ۱۷۸: Multiplexing

اگر مقصد این Applicationها یکی باشد این Applicationها می توانند از طریق یک واحد اطلاعاتی ارتباط برقرار کنند. این روش Upward از Multiplexing است و روش Downward هم وجود دارد که بر عکس است. در روش Down ward یک Application به دلیل محدودیت خطی که وجود دارد، ۳ تا Connection مجزا باز کرده تا سرعت خود را از این طریق افزایش دهد. این کار به دلیل پهنای باند کم می باشد. این Connectionها ممکن است دارای یک مقصد باشند و یا در ۳ مقصد جداگانه باشند.

برقراری ارتباط در لایه Transport:

وقتی لایه Transport مبدا می خواهد با لایه Transport مقصد ارتباط برقرار کند، یکی از راه های برقراری این ارتباط استفاده از ۳wat hand check می باشد. از طریق این ۳way hand check فرستنده یک Connection Rey می فرستد برای درخواست برقراری ارتباط.



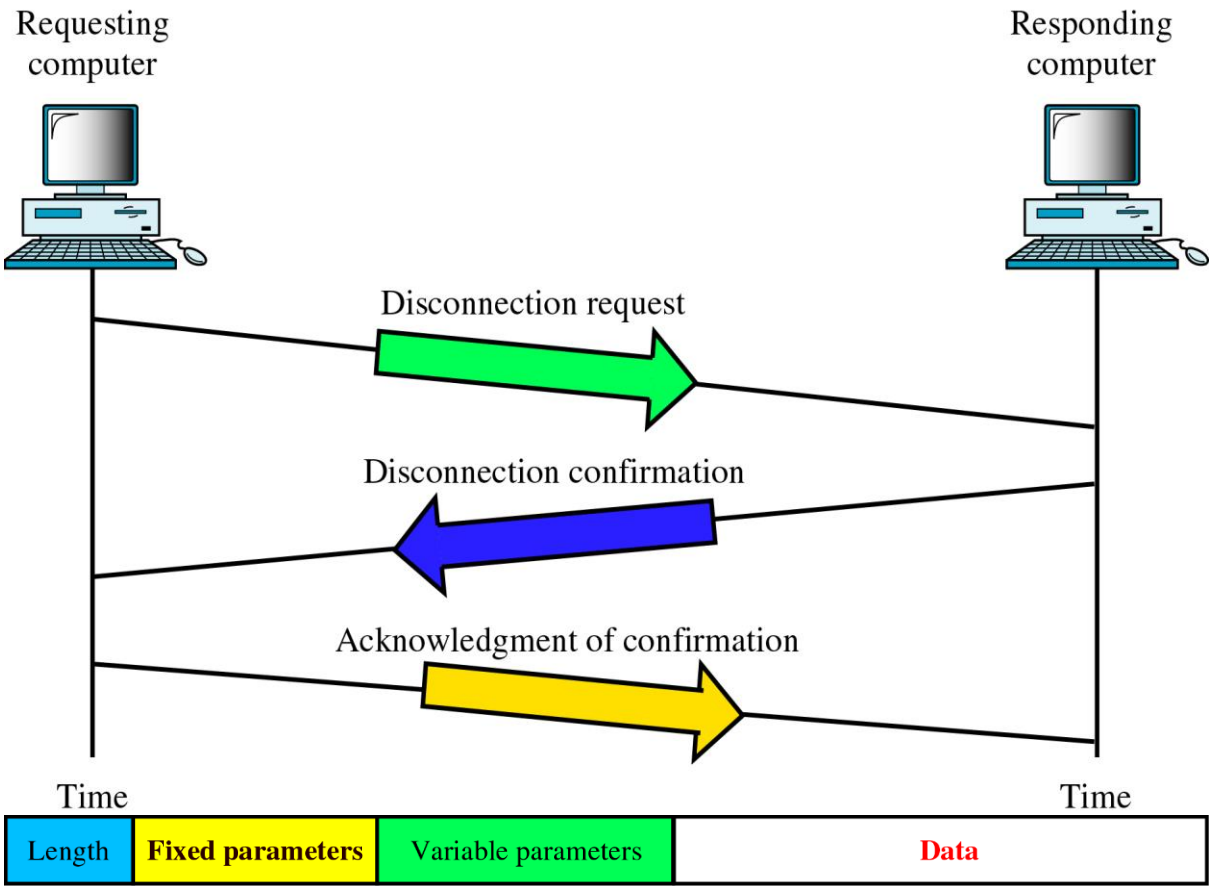
شکل ۱۷۹: Connection Establishment

گیرنده هم تاییدیه برای برقراری ارتباط می فرستد و باز فرستنده ACK را برای تایید دریافت تایید گیرنـده می فرستد. پس روش ۳way دارای ۳ مرحله زیر است:

۱. Connection Request
۲. Connection Confirmation
۳. Acknowledge of Confirmation

همچنین برای قطع ارتباط از این روش استفاده می گردد. در روش ۳wat hand shake برای قطع ارتباط باز هم ۳ مرحله زیر را داریم:

۱. Disconnection Request
۲. Disconnection Confirmation
۳. Acknowledge of Confirmation



Fixed Parameters

**** Code:**

- **CR: Connection Request**
- **CC: Connection Confirm**
- **DR: Disconnection Request**
- **DC: Disconnection Confirm**
- **DT: Data Transmit**
- **Ak: Data Acknowledge**
- **RI: Reject**
- **ER: Error**

- **ED :Expedited Data**

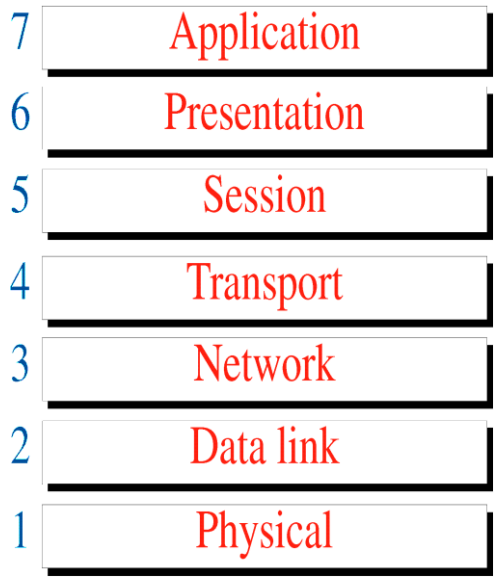
** Source and Destination Refrence

** Sequence Number

** Credit allocation

Session layer:

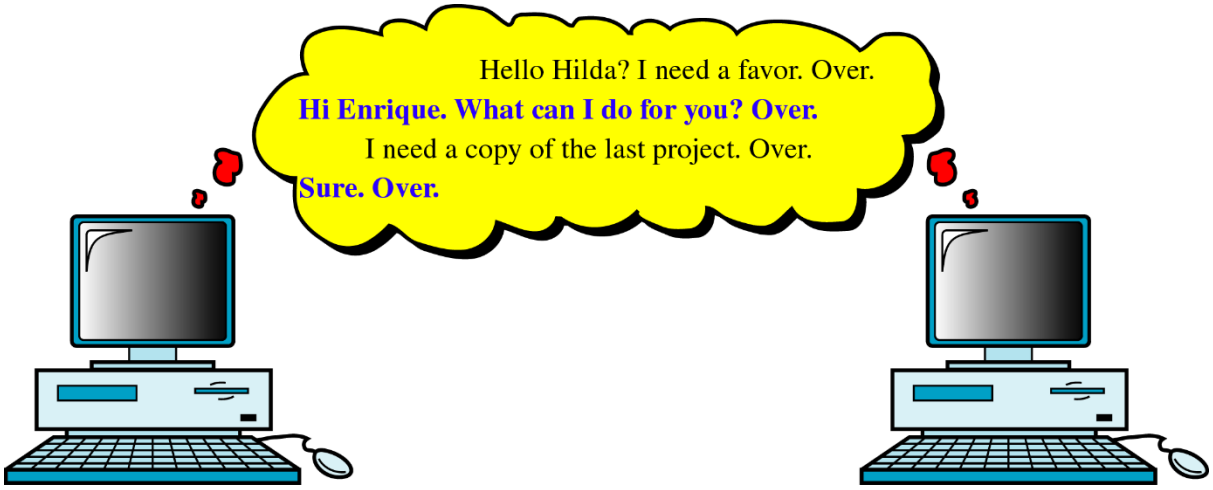
لایه Session پنجمین لایه از مدل OSI می باشد. در پروتکل های معروف گفتیم که لایه ۵ بـه بـالا را در قالب یک لایه به نام Application داریم.



شکل ۱۸۲: جایگاه Session layer در مدل OSI

یعنی لایه Session و Presentation و Application در قالب یـک لایـه مـی باشـد. لایـه Session وظیفه مدیریت جلسات و انجام جلسات و هماهنگی جلسات را بر عهده دارد که اینهااز وظـایف اصـلی لایـه Session می باشد. می توان گفت لایه Session یک Diaiog است. یعنی مجموعه ای از اطلاعاتی که رد و بدل می شود. ما اطلاعات را از لایه Application (مجموع ۳ لایه بالا) برای فرستنده ارسال مـی کنـیم. این اطلاعات وارد لایه Transport می شود و در آنجـا بسـته بنـدی مـی گـردد. لایـه Transport مـا و Transport طرف مقابل با یکدیگر Hand check انجام می دهند و فریم ها را کنترل می کنند که سالم رسیده باشد و اگر اطلاعات اشتباه ارسال شده باشد، لایه Transport طـرف مقابـل پیغـام مـی دهـد کـه اطلاعات را دوباره بفرست (چون لایـه Transport، Flow control دارد). سـپس اطلاعـات کنترلـی آن

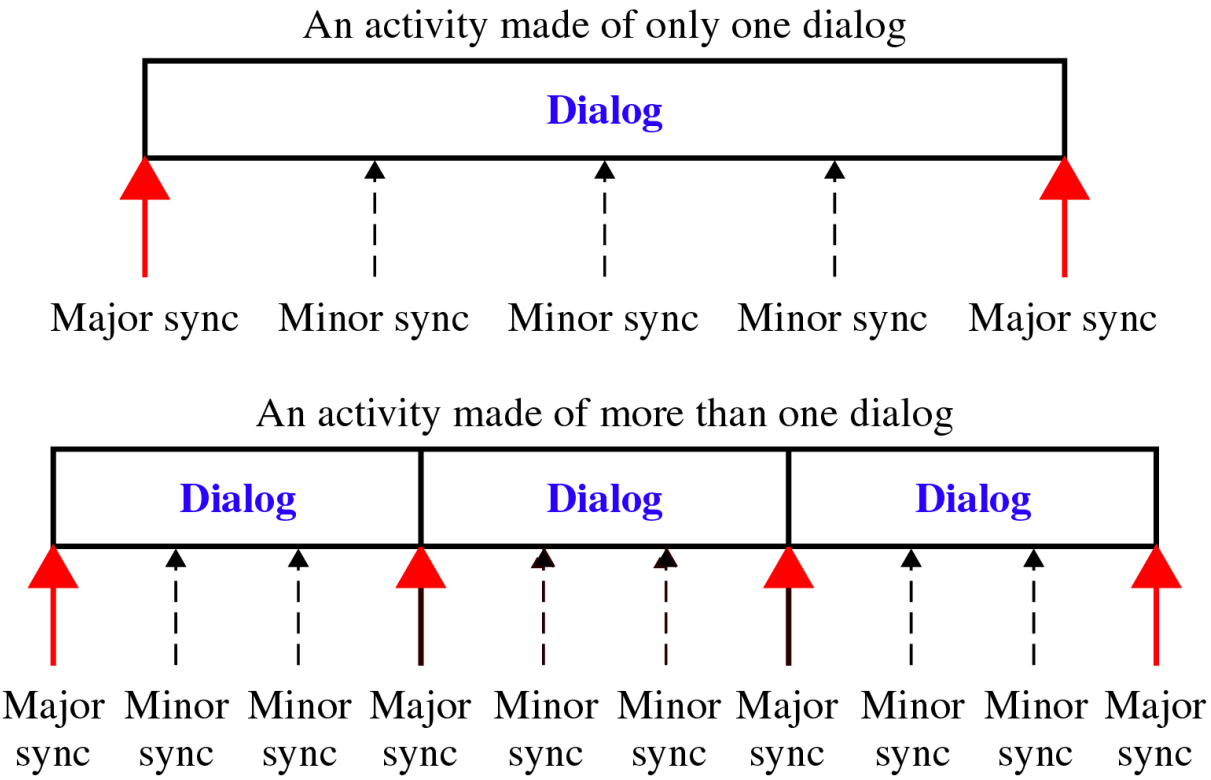
اضافه می گردد و به لایه Network می رود و .. پس از این که Transport طرف مقابل اطلاعات را به صورت سالم دریافت کرد، اطلاعات کنترلی آنرا جدا کرده و تحویل لایه Session می دهد.



شکل ۱۸۳: Session layer Dialog

تا اینجا اطلاعات به طور صحیح ارسال و دریافت می شوند. مشکلی که پیش می آید این است که ما می خواهیم این Dialogای که اجرا می شود یا همه آنها انجام شود و یا هیچ کدام آنها انجام نشود. اگر در حین انجام یک Dialog برق قطع شد، اطلاعات به صورت نیمه کاره ارسال می شوند. این مشکل را لایه Transport طرف مقابل با صادر کردن Error اعلام می کند ولی ما می خواهیم همه این اعمال انجام شوند و یا هیچ کدام (در دو صورت بروز خطا) انجام نشوند.

نمی توان گفت که این شکل مربوط به لایه Transport است چون این سرویس در لایه Transport وجود ندارد و نمی تواند این کار را انجام دهد. این سرویس در لایه Session که در پروتکل های معروف در قالب Application است قرار گرفته است. که اگر ما بخواهیم از این سرویس استفاده کنیم، باید از چنین پروتکل هایی بهره بگیریم. وقتی که حین انجام یک Dialog اتفاقی رخ می دهد دیگری که لایه Sesssion انجام می دهد این است که زمانی که ما از اینترنت فایلی را Download می کنیم، اگر در بین کار، ارتباط قطع شود، دوباره باید از اول اطلاعات را Download کنیم. لایه Session برای سرویس دادن به این مشکل نیز طراحی شده است و ما با نصب یک برنامه Download Acceptor می توانیم از این سرویس لایه Session استفاده کرده و از ادامه جایی که Download کردن قطع شده بود، به ادامه Download بپردازیم.



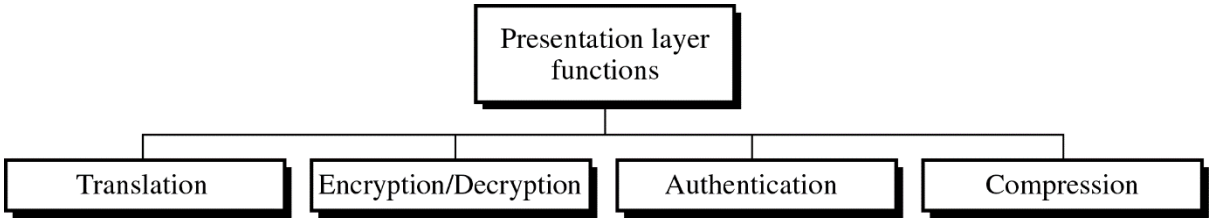
شکل ۱۸۴: Synchronization Points

کاری که لایه Session برای ارائه این سرویس انجام می دهد در دو قالب است (از طریق نقاط یا Synchronization points) که یک Majar sync و یک Minor sync دارد که در حالت Major، Dialog را به صورت بعدک تقسیم بندی می کنید و پس از ارائه هر بعدک از گیرنده یک Confirm می گیرد. مثلاً به ازای هر K ۱۰، یک Confirm از گیرنده دریافت می کند. حال اگر ارتباط قطع شود، باز می گردد به آخرین Confirmای که از گیرنده دریافت کرده ایت و ادامه را از آن نقطه شروع می کند. در حالت Minor، دیگر Confirm ای از گیرنده دریافت نمی کند. پشت سر هم اطلاعات را می فرستددو فقط علامت می گذارد. اگر ارتباط در بین کار قطع شد، کافی است که کمی به عقب باز گردد و از آن نقطه شروع به ارسال نماید.

قطعاً حالت Minor از حالت Major سریعتر عمل می کند چون از گیرنده Confirmای دریافت نمی کند. می توان از یکی از حالتهای Major یا Minor و یا هر دو به صورت هم زمان استفاده نمود.

Presentation Layer:

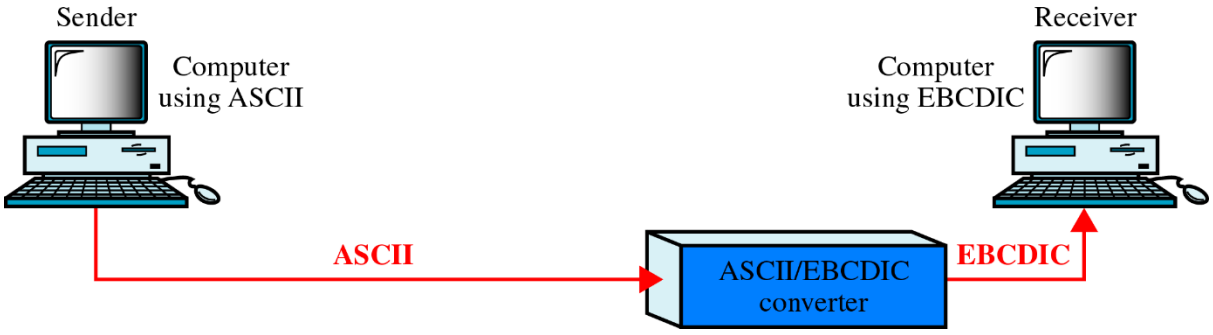
یکی از بهترین لایه ها در مدل OSI می باشد که ۴ وظیفه (Task) مهم و اصلی برای آن در نظر گرفته شده است که عبارتند از:



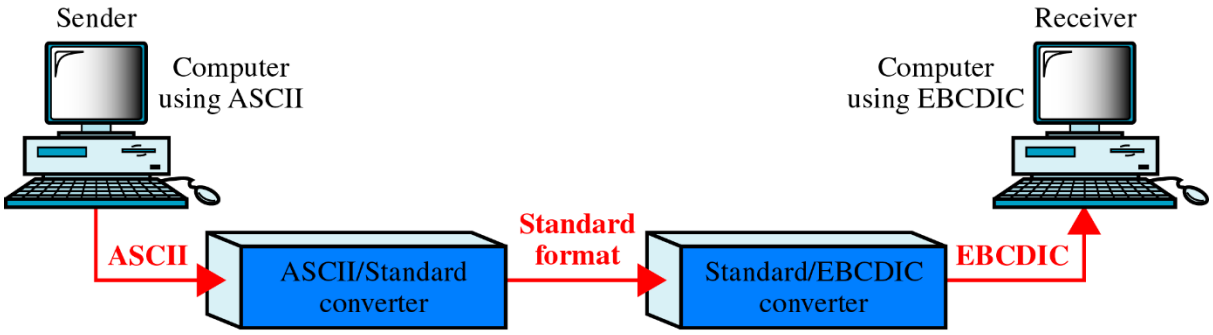
شکل ۱۸۵: وظایف لایه Presentation

۱-Translation (ترجمه)

ما در شبکه با تونع Cadingهای متفاوت مواجه هستیم. هر کاراکتر یک کد اسکی برای خود دارد که وقتی کاراکتر درج می شود کد اسکی آن روی خط قرار می گیرد. مثلاً حرف A دارای کد ۶۵ می باشد. فرستنده ۶۵ را می فرستد و گیرنده هم باید دارای همان کد اسکی بشد تا بتواند ۶۵ را به حرف A تبدیل نماید. حال اگر Cading طرف مقابل با Cading ما متفاوت باشد نمی توان این عمل تبدیل را انجام داد. بنابراین مشکل استفاده از یک Convertor می باشد. کافی است جدول ASCII خود و جدول طرف مقابل (مثلاً EBCDIC) را داشته باشیم و کد اسکی را به کد EBCDIC تبدیل کنیم. پس یک مبدل ASCII to EBCDIC داریم که کد ما را تبدیل به کد طرف مقابل می کند. این یک حالت است که به آن روش مستقیم (Direct) می گویند.



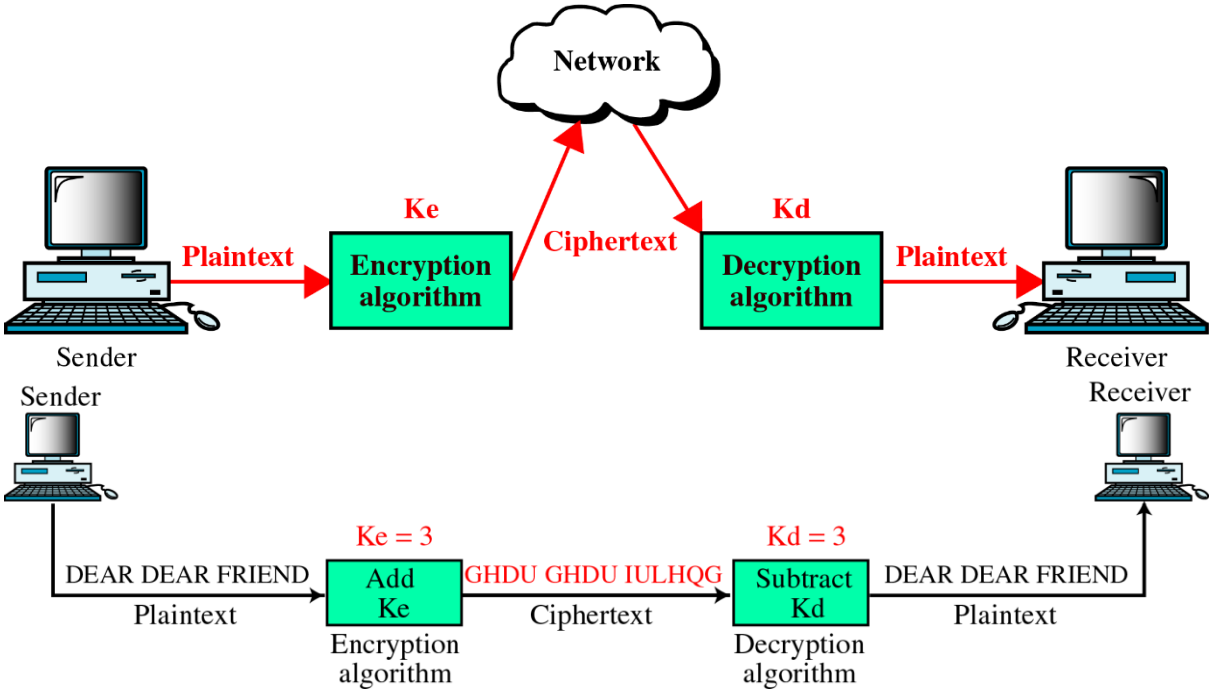
a. Direct method



b. Indirect method

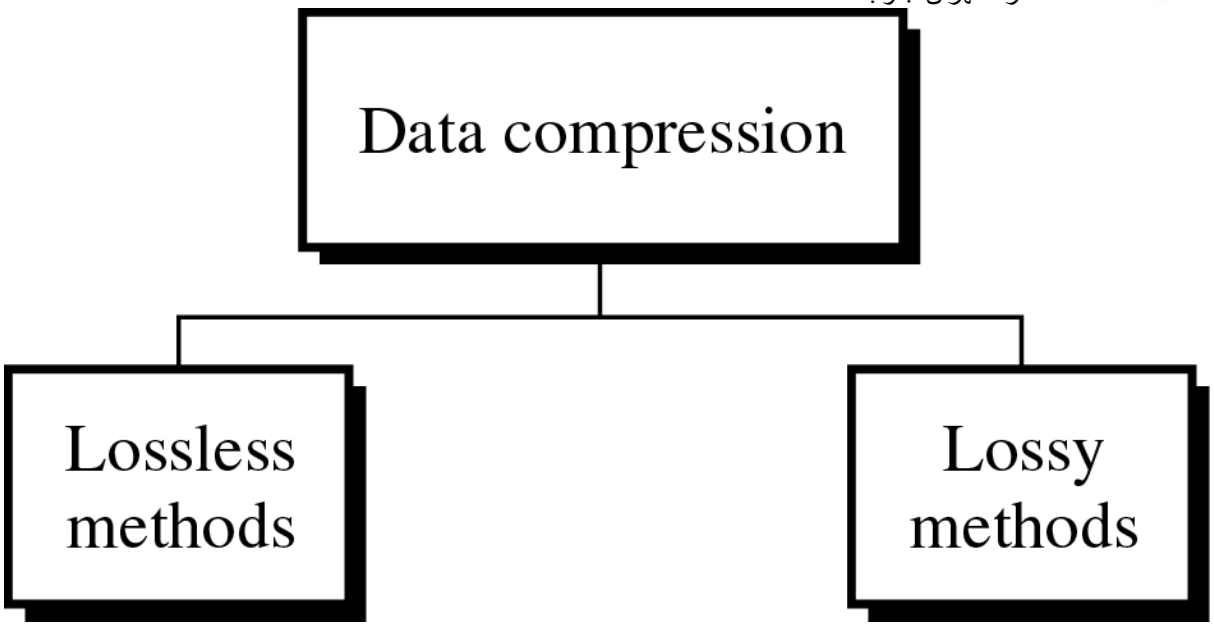
شکل ۱۸۶: روشهای مستقیم و غیر مستقیم ترجمه

روش دوم برای حل مشکل روش اول ارائه می شود. شکل روش اول این است که ما با تنوع Cadingهای زیادی مواجه هستیم و در روش اول ما به عنوان فرستنده باید مبدل تمام این کدها را داشته باشیم که قطعاً مشکل ساز خواهد بود. برای حل این مشکل در روش Indirect، یک Frame استاندارد (Standard Frame) تعریف می کنیم و کافی است که فرستنده و گیرنده هر یک مبدل کد خود را داشته باشند. مثلاً طبق شکل اگر فرستنده با ASCII Cade کار می کند باید مبدل ASCII را داشته باشد و گیرنده با EBCDIC کار می کند باید مبدل EBCDIC را داشته باشد تا به راحتی تبدیل Cadeها را انجام دهند.



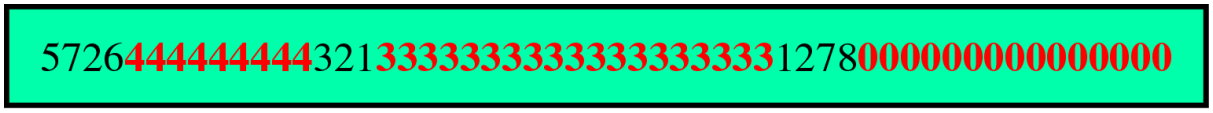
۳-Data Compression (فشرده سازی اطلاعات)

بحث فشرده سازی اطلاعات نیز از مهترین مباحث این بخش می باشد که در دو قالبریز مطرح می شود. در روش Less Lass اطلاعات زمانی که فشرده می شوند از دست نمی روند ولی در روش Lossy وقتی اطلاعات را فشرده می کنیم بخشی از این اطلاعات از بین می روند. JPEG و MP۳ از روش های فشرده سازی اطلاعات هستند که در آنها حجمی از اطلاعات از بین می روند (در زمان فشرده سازی). در MP۳ یک سری از فرکانسها کاملاً حذف می شوند ولی فرکانسهایی هستند که در گوش ما تاثیری به آن صورت ندارد. ما برای برخی اطلاعات می توانیم از روش Lossy استفاده کنیم. اطلاعاتی که به صورت Data بوده و حیاتی هستند را نمی توان از طریق فشرده کرد. چون اطلاعات در Data مورد نیاز نباید از دست برود.

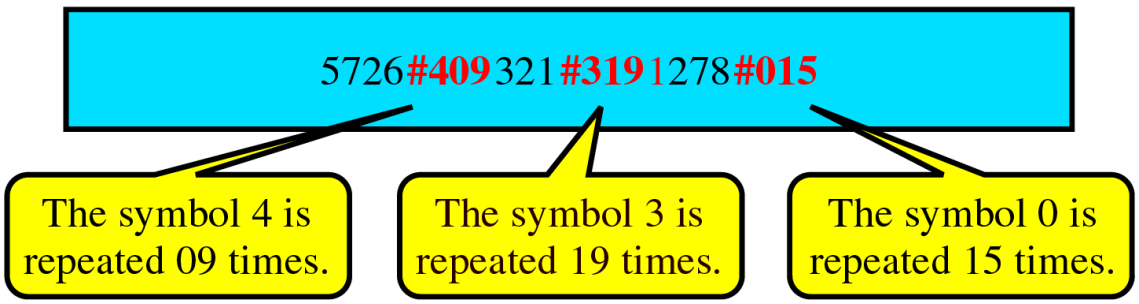


شکل ۱۹۰: روش های فشرده سازی اطلاعات.

در روش Less Lass روشی به نام Run-Length encoding وجود دارد که در این روش یک سمبل خاص برای رقم هایی که تعداد تکراری زیادی دارند در نظر می گیرد سپس خود آن عدد و بعد تعداد تکراری آن عدد را می نویسد.



a. Original data



b. Compressed data

شکل ۱۹۱: Run-Length encoding

مثلاً عدد ۴ در رشته مورد نظر، ۹ بار تکرار شده پس داریم: ۴۰۹

و یا عدد ۱۹، ۳ بار تکرار شده که خواهیم داشت: ۳۱۹ و ...

به این ترتیب اطلاعات به مقدار خیلی زیادی فشرده می شوند. روش های متفاوتی برای فشرده سازی اطلاعات وجود ارد. روش دیگری که می توان مطرح کرد، روش Statistical یا روش های آماری است که شامل ۳ روش Morse و Huffman می باشد. در کد Morse برای حروف مختلف تعداد بیت های متفاوتی در نظر گرفته می شود. مثلاً برای حروفی که تعداد تکراری بیشتری دارند، تعداد بیت های کمتر و برای حروفی که تعداد تکراری کمتری دارند، تعداد بیت های بیشتر در نظر گرفته می شود. اگر بخواهیم تعداد بیت های همه، حروف را یکسان بگیریم، این کد خیلی طولانی و برای کاربر خسته کننده می شود همچنین سرعت را پایین می آورد.

بر مبنای این ایده که ما ضمن ایجاد شده است. که از روی این تکراری ها، یک Tree درست می کند. (از تعداد تکرار ها) از روی Tree به هر حرف یک کد اختصاص می دهد. سپس یک جدول بر مبنای این کد ها ایجاد کرده و در Header هر یک از اطلاعاتی که می خواهد ارسال کند قرار می دهد و برای گیرنده ارسال می کند. در گیرنده کافی است که از روی Header، عملیاتی Unzip را انجام دهد. Morse و Huffman یک ایده دارند با این تفاوت که در کد درس جدول ثابت است ولی در روش ضمنی، جدول بر اساس تکرار تغییر می کند.

روش های دیگری به نام Reiative Compression وجود دارد. در این روش اختلاف ها ارسال می شود. (اختلاف یک Frame با Frame دیگر.) مثلاً در فیلم های انیمیشن، اگر ما بخواهیم تمام Frameها را ذخیره کنیم، حجم بسیار زیادی اشغال خواهد شد. به عنوان مثال یک فیلم ۶۰ ثانیه ای با کیفیت بسیار پایین و رنگ کم چیزی حدود ۶GB فضا اشغال می کند که ذخیره آن روی یک CD ممکن نیست. حال اگر از روش Relative استفاده کرده برای فشرده سازی و فقط اختلاف Frameها را ذخیره کنیم مشکل تا حد زیادی برطرف می شود. مثلاً تمام Back groundها شبیه هم است. کافی است که ما تنها یکی از آنها را ذخیره کنیم.

روش های Lossy:

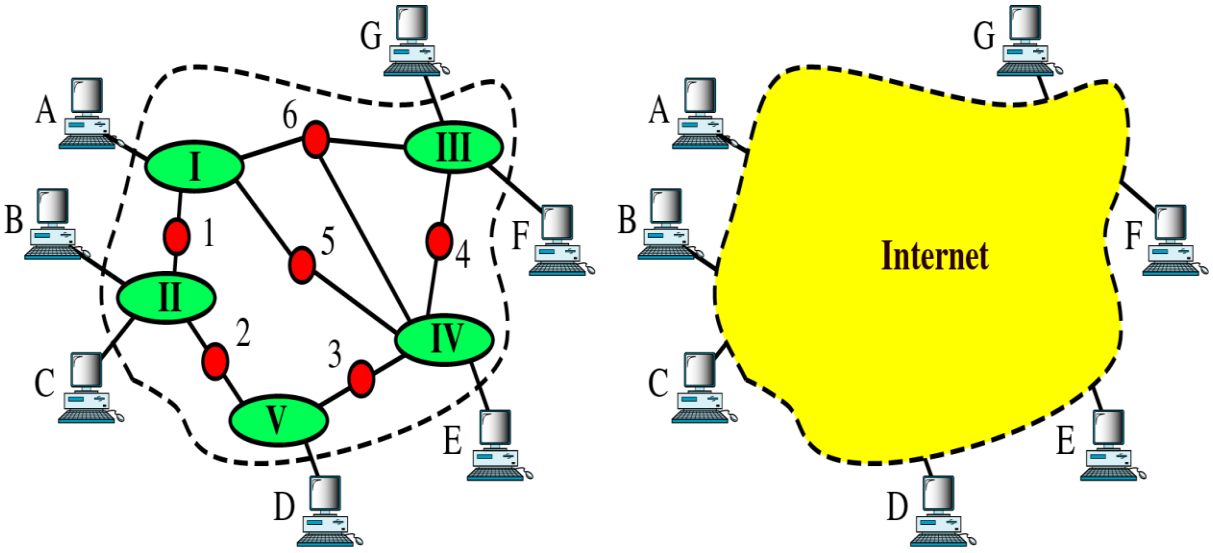
روشهایی هستند که وقتی اطلاعات Zip یا Unzip می شود، قسمتی از آنها از بین می رود. در این روش ها، سری های فوریه کاربرد زیادی دارند. مثلاً تصویر مورد نظر را به صورت یک ماتریس در می آورند. در یک سری فوریه ضرب کرده و تشخیص می دهند که حجم بیشتر اطلاعات در یک قسمت خاص است و این قسمت برای گیرنده ارسال می شود. گیرنده هم اطلاعات دریافت شده را در همان سری فوریه ضرب می کند و به تصویری مشابه تصویر مورد نظر دست می یابد. نمونه این روش ها، فایلهای PEG به y هستند که هر چند فشرده می شوند اطلاعات بیشتری از دست می رود.

۴-Authentication:

این بحث مربوط به اعتبار سنجی می باشد که یک بحث کلیدی و مهم است. که تشخیص دهیم آیا ایجاد سایت اینترنتی به سود ماست یا به ضرر ما.

TCP/IP:

پروتکل TCP/IP به عنوان یک پروتکل بسیار معروف و پرکاربرد مورد استفاده قرار می گیرد و حذف مقایسه این با مدل OSI است.در شکل زیر TCP/IP با یک شبکه internet (از لحاظ ارتباطات بین شبکه ای) مقایسه شده است که همانطور که دیده می شود. در شبکه a یک سری شبکه و یک سری Router دیده می شود که از هر یک از این شبکه ها nodeهایی نمایش داده شده است.



a. An actual internet

b. An internet seen by TCP/IP

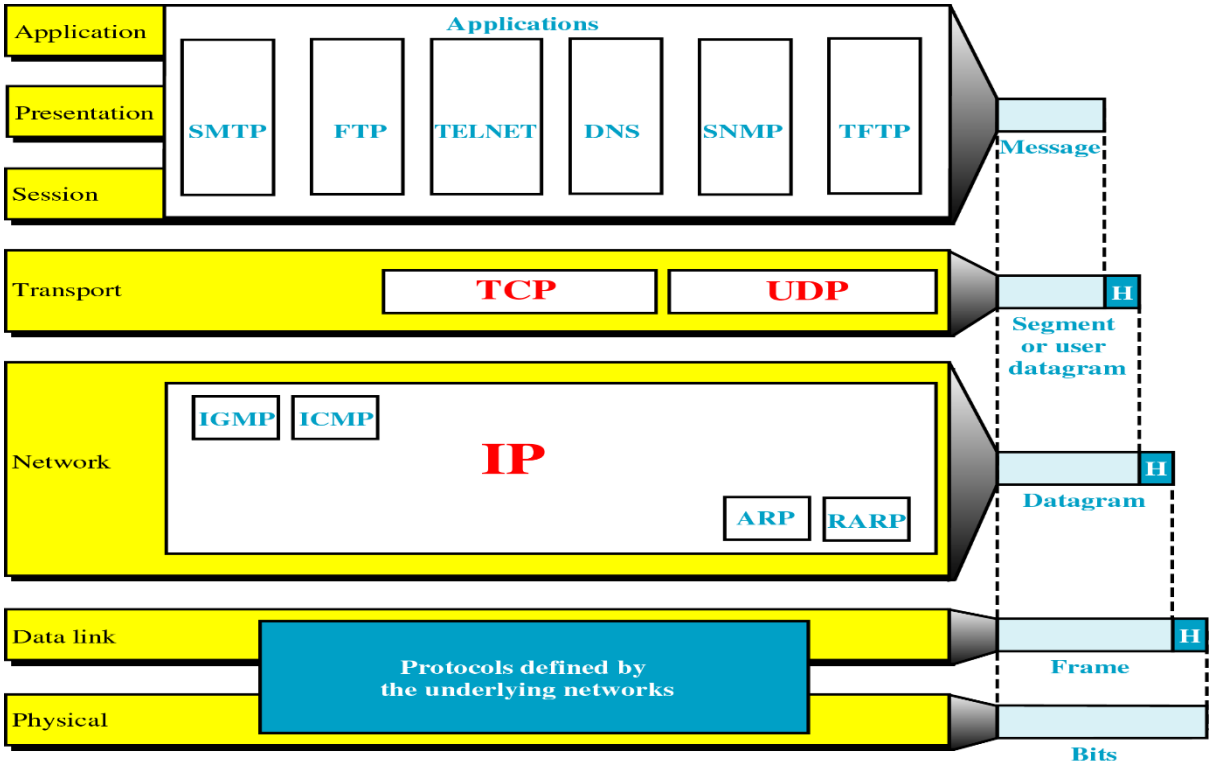
شکل ۱۹۲: An Internet Acoording to TCP/IP

این شبکه از دید یک کاربر هم شبکه ها هم Routerها و هم Nodeها را می بیند.

اما در مثال b از دید پروتکل TCP/IP کل شبکه در قالب یک ابر دیده می شود. ابری که node های a,b,c و ... جزئی از آن ابر هستند و کاری به جزئیات این ابر ندارند که یک Packet از کدام شبکه و از چه Routerای می گذرد و ...

مقایسه پروتکل TCP/IP و مدل OSI:

در این شکل مدل OSI که شامل ۷ لایه می باشد به همراه پروتکل TCP/IP نمایش داده شده است.



شکل ۱۹۳: TCP/IP و مدل OSI

دو لایه زیرین یعنی Physical و Datalink اصطلاحاً Host to network نامیده می شوند. که پروتکل هایی که در این بخش تعریف می شوند به شبکه های داخلی ما بر می گردد که از چه نوعی است، Ethernet و tokenring و ... که TCP/IP با این تست کاری ندارد و فقط اطلاعات را تحویل این لایه می دهد تا آنرا حمل کند. چون این دو لایه به معماری شبکه بر می گردد و بر مبنای این دو لایه شبکه ها به انواع مختلف تقسیم می شوند. پس می توان گفت اولین لایه از پروتکل TCP/IP، لایه host to network می باشد. همچنین می توان همان دو لایه physical و datalink را به عنوان اولین لایه های پروتکل TCP/IP در نظر گرفت (هر دو حالت ممکن است).

لایه سوم (یا دوم) از پروتکل TCP/IP لایه ای با نام لایه internet است که معادل با لایه network از مدل OSI است که لایه internet شامل ۵ پروتکل بسیار معروف با نامهای پروتکل IP,ARP,RARP,ICMP,IGMP.

لایه چهارم در پروتکل TCP/IP لایه Transport است که معادل می باشد کـه دارای دو پروتکـل معـروف TCP و UDP می باشد.

لایه سیم در پروتکل TCP/IP لایه Application است که معادل لایه های ۵ و ۶ و ۷ در مدل OSI است.

پس می توان گفت TCP/IP دارای ۵ لایه به ترتیب زیر است:

Physical, datalink, internet, transport, application

و یا بر مبنای host to network دارای ۴ لایه به ترتیب زیر می باشد:

Host to network, internet, transport, application

که ما مدل ۵ لایه ای را در نظر می گیریم.

چون در مورد لایه های physical و datalink بحث شده است، از لایه internet شروع به بحث در مورد پروتکل TCP/IP می نمائیم:

لایه internet:

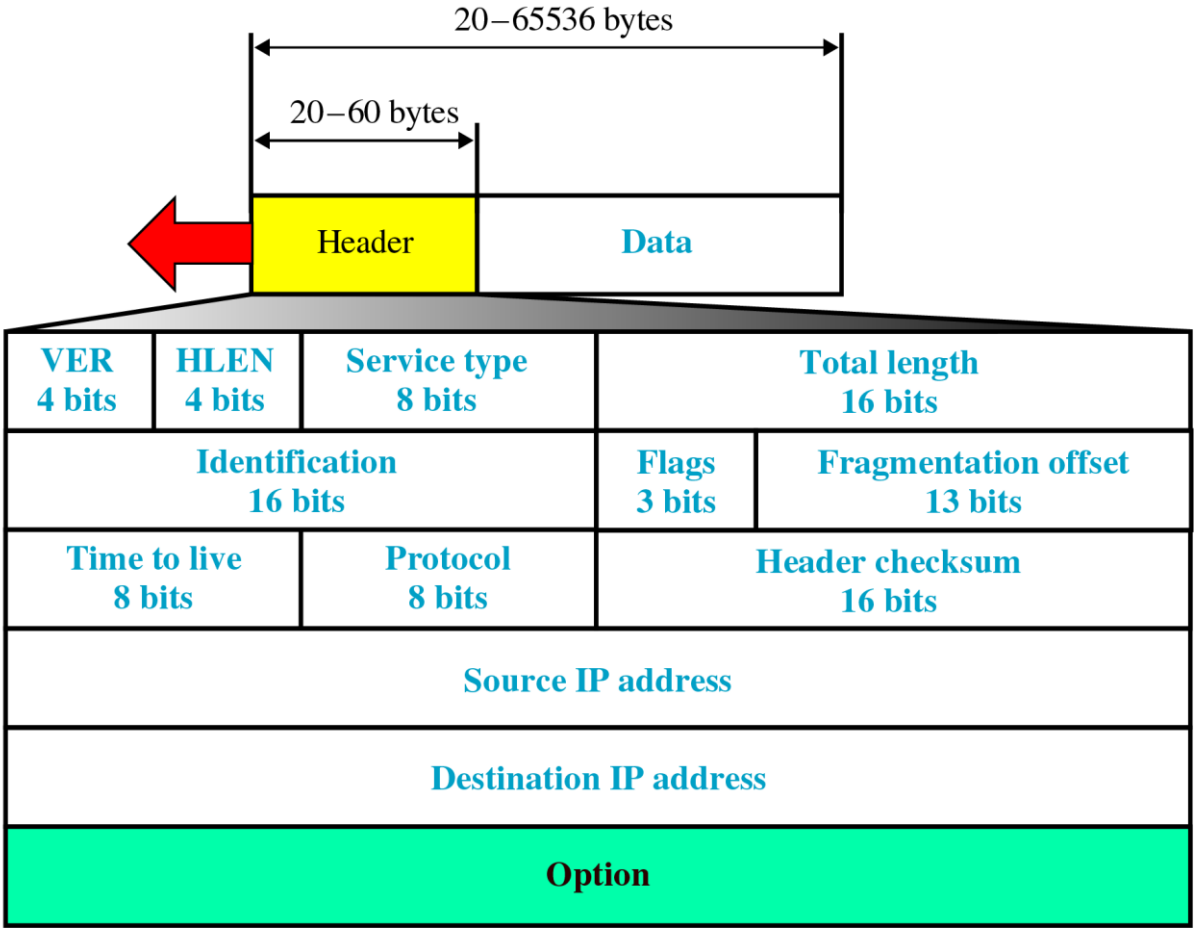
قطعاً لایه اینترنت (internet) دارای وظیفه مسیریابی، کنترل ازدحام و پیدا کردن آدرسهای لاجیکـی و ... می باشد. همه این کارها به پروتکلی به نام IP که مهمترین پروتکل در لایه internet اسـت محـول شـده است. internet protocol (ip) می باشد که مسیریابی داخل packetها و ... بر عهده دارد.

گفتیم که لایه Network در مدل OSI با Packetها سر و کار دارد و packetها راجا به جا مـی کنـد و اینکه چه چیزهایی در Packet وجود دارد گفتیم یک سری Data می باشد که اطلاعـات کنترلـی شـامل آدرس لاجیکی فرستنده، آدرس لاجیکی گیرنده و ... می باشد. حال باید دید به صورت واقعـی در پروتکلـی که روی آن کار می شود، در Packetها چه چیزهای وجود دارد و جزئیـات داخـل آن چیسـت کـه در IP Datagram توضیح داده شده است.

IP Datagram:

هر Datagram یا Packet در IP، دارای دو بخش می باشد. Data و header

Header بین ۲۰ تا ۶۰ بایت است و Data اطلاعاتی است کـه از لایـه بـالاتر مـی آیـد و قـرار اسـت کـه packet آنرا حمل کند. مجموع کل packet می تواند بین ۲۰ تا ۶۵۵۳۶ باید باشد که تقریبـاً معـادل ۶۴ کیلو می باشد.



شکل ۱۹۴: IP Datagram

جزئیات Header شامل فیلد های زیر است:

فیلد ver که ۴bit است و Version، IP را مشخص می کند.

فیلد بعدی، فیلد hlen یا header length است که ۴ بیتی است و طول header را مشخص می کند که می تواند بین ۲۰ تا ۶۰ بایت متغییر باشد.

فیلدهای Service type که ۸ بیت است که بیت شماره صفر تا بیت شماره ۲ آن مربوط به اولویت است که این امکان در Header یک IP رزرو شده است که بتواند packetها را اولویت بندی کند. بیت شماره ۳ از service type بیتی بنام d است که اگر set شود اعلام می شود که این packet باید تا کمترین تاخیر به مقصد برسد. (d به معنای delay) که این بیت بیشتر برای مسیریاب ها طراحی شده است که اگر دیدند این بیت set شده است خارج از نوبت packet را بفرستند تا برود. بیت شماره ۴، بیت t می باشد به معنای true put که اگر set شود اعلام می کند که packet باید از مسیر هایی بگذرد که بیشترین قابلیت گذردهی را دارد یا high true put است. بیت شماره ۵ بیت r است که اگر set شود بدین معنی

است که packet باید با بالاترین قابلیت اطمینان به مقصد برسد یا high reliability بیت شماره ۶و ۷ بیتهای unused هستند که استفاده نمی شوند.

مسیریابی های تجاری به فیلد service type نگاه نمی کنند.

فیلد بعدی total length است که ۱۶ بیتی می باشد که ماکزیمم طول datagram یا packet را مشخص می کند که می تواند تا ۶۴k را آدرس دهی کند.

سه فیلد بعدی Identifigation و flags و fragmentation می باشد که در اینجا بحث fragmentation مطرح می شود.

Fragmentation:

گفتیم که packetها می توانند ۶۴k حجم داشته باشند. فرض کنید که host to network شما اترنت است که packet باید از طریق frameهای اترنت حمل شود. می دانیم که ماکزیمم طول در اترنت ۱۵۰۰ بایت است. پس برای فرستنده اول packet به شکل بر می خوریم و بحث شکستن اطلاعات مطرح می شود که packetها به واحد های کوچکتری شکسته می شوند و قطعاً وقتی که packet می شکند هر یک از اجزا آن مهم یک packet است که خود یک ساختار packet شامل header و data و ... دارد.

Multi fragmentation:

وقتی packet می خواهد ارسال شود و این packet به routerای می رسد که حجم آن کمتر از حجم packet است، packet را به دو بخش یا بیشتر می شکند و آنرا عبور می دهد. این packetها به router بعدی می رسند که باز هم حجم این packetها را به واحدهای کوچکتری شکند تا بتواند آنها را عبور دهد. پس ممکن است که packet چنین عبور از routerهای مختلف مرتب شکسته شود (چندین بار) که در اینجا بحث multi fragmentation دیده می شود که یک packet چندین بار شکسته می شود.

دیگر اینکه عمل defrage فقط در مقصد انجام می شود و در هیچکدام از router ها عمل defrage انجام نمی شود.

فیلد بعدی که در شکل دیده می شود، فیلد Flag است که ۳ بیتی می باشد که یک بیت آن MF یا Don't Fragmentation است که اگر set شود اعلام می کند که حق شکستن اطلاعات را ندارید. وقتی که فرستنده می داند گیرنده امکان defrage کردن را ندارد این را اعلام می کند که وقتی packet به router ای رسید و نتوانست عبور کند، router آن را discard کند. به همین دلیل می توان گفت که

پروتکل IP یک پروتکل غیر قابل اطمینان است چون ممکن است packetها در بین راه از بین بروند. بیت سوم flag استفاده نمی شود و unused است. در بیت MF که اعلام می کند packetها می توانند شکسته شوند، وقتی یک packet به چندین packet تبدیل می شود در بیت MF همه آنها به جز آخرین packet، یک قرار می گیرد که اعلام می کند این packet که بیت MF آن صفر است، آخرین packet شکسته شده می باشد.

فیلد بعدی Fragmentation offset می باشد. گفتیم وقتی بحث شکستن اطلاعات مطرح می شود Packetها برای اینکه جا به جا نرسند از یک شماره ترتیب برای packet استفاده می کنیم. در packetهای IP به جای این شماره ترتیب، offset نگهداری می شود که این offset برای مرتب سازی استفاده می شود و position آن واحد اطلاعاتی از اطلاعات کل را نشان می دهد. مثلاً مشخص می کند که این packet از بایت هفتم (نسبت به کل اطلاعات) شروع شده است. سپس عمل data را در data اصلی نشان می دهد و گیرنده از روی این offset می تواند هر packet را در جای خود قرار دهد.

فیلد دیگر، Identification می باشد که ۱۶ بیتی است (بیت هویت). خر packet وقتی که می خواهد شکسته شود، یک فیلد هویت در داخل آن قرار می گیرد که فرستنده هنگام شکستن اطلاعات یک عدد random را در این فیلد قرار می دهد (به ازای هر packet).

وقتی که این packetها به routerها می رسند و در routerها شکسته می شوند در فیلد Identification همه packetهای شکسته شده همان عدد قرار می گیرد. مثلاً اگر ۴۷ packet به router رسید و به ۳ تا packet دیگر شکسته شد در فیلد Identification هر ۳ packet عدد ۴۷ قرا می گیرد و در گیرنده می تواند تشخیص دهد که این ۳ تا packet مربوط به یک پیغام هستند چون این اعدادی که در فیلد identification قرار می گیرند، Random هستند، ممکن است که در دو packet یک عدد قرار گیرد و اشتباه شود. سپس دلیل دیگری است که بتوان گفت پروتکل IP پروتکل مطمئن نمی باشد.

فیلد بعدی، فیلد Time to live است که ۸ بیتی می باشد و برای عمر یک packet استفاده می شود. برای اینکه از عبور packetهای سرگردان جلو گیری شود، packet از هر مسیر که عبور کند یک واحد از TTL آن کم می شود و router چک می کند که آیا TTL صفر شده و به مقصد نرسیده اگر این اتفاق افتاده باشد، packet سرگردان است و discard می شود. در غیر این صورت یک واحد از مقدار TTL کم کرده و آنرا به router بعدی می دهد. ماکزیمم مقداری که در TTL می تواند قرا بگیرد از ۰ تا ۲۵۵ است چون Time to live، ۸ بیتی است (در مسیریاب ها به ازای یک ثانیه تاخیر هم یک واحد از TTL کم می شود.)

فیلد بعدی Header check sum است که، ۱ بیتی می باشد. Check sum برای Header است نه برای Discard که کشف خطا را در Header به عهده دارد و از همان منطق Check sum استفاده می کند. این Check sum به ازای تمام packet ها هم در مبدا و هم در مقصد محاسبه می شود.علاوه بر آن در Routerها هم به دلیل اینکه یک واحد از مقدار TTL کم می شود، باز هم باید Check sum محاسبه گردد.

Original IP Packet

فیلد بعدی، protocol است که ۸ بیتی می باشد که مشخص می کند چه پروتکلی این اطلاعات را تولید کرده است که گفتیم پروتکل ها در لایه Transport قرار دارند (TCP یا UDP) که به ازای هر یک از این پروتکل ها یک بیت خاص قرار می گیرد.

فیلد بعدی IP Address است که لاجیکی در پروتکل Topip با عنوان IP Address مطرح می شود که یک Source IP Address و یک Distination داریم که یکی برای آدرس لاجیکی مبدا و دیگری آدرس لاجیکی مقصد می باشد که هر یک ۳۲ بیت هستند.

فیلد بعدی، فیلد option است که ۴۰ بایت می باشد. این فیلد به تنهایی ۴۰ بایت و بقیه فیلد ها روی همدیگر ۲۰ بایت می باشند که کلاً ۶۰ بایت Header را تشکیل می دهند. می توان فیلد option را حذف کرد چون حداقل مقدار برای header ۲۰ بایت در نظر گرفته شده است. Option برای حالتهای خاص استفاده می شود و کدهای متفاوتی می توان در آن درج کرد که هر یک معنای خاصی دارد. مثلاً کدی به نام (مهر زمان) که اگر این فیلد در option، Set شده باشد، Router آدرس و زمانی که packet از آن Router گذشته است را داخل فیلد option درج می کند.

و یا کدی برای مسیر در فیلد option در نظر گرفته می شود که مشخص می کند packet از کدام مسیر عبور کرده است و هر Router آدرس خود را در فیلد option درج می کند.

و یا فیلدی به نام (مسیریابی غیر دقیق) که مشخص می کنیم packet از مسیر های دیگری عبور کند.

به طور کلی این الگو، الگوی یک header بود که در مسیر یابها و پروتکل TCP/IP از آن استفاده می شود. نکته ای که در اینجا مطرح می شود این است که fragmentation offset، B بیتی است و می تواند تا ۸k را آدرس دهی کند در حالیکه کل ۶۴k Datagram می باشد و با B بیت نمی توان ۶۴k را آدرس دهی نمود. سپس با این ۱۳ بیت ۸k، ۸k آدرس دهی می کند یعنی ۸k اول، ۸k دوم، ۸k سوم، تا ... که به این ترتیب مشکل آن حل می شود.

صفحه ۱۲۸و ۱۲۹ و ۱۳۰

پس در کلاس A که ۷ بیت برای شماره شبکه قرار داده شده است می توان حداکثر ۷ شبکه داشت و در داخل هر شبکه می توان تا $۲^{۲۴}$ یعنی ۱۶ میلیون، Host داشت.

در کلاس B، دو بیت اول رزرو می باشد و دو بایت اول مربوط به شماره شبکه و دو بایت بعد مربوط به شماره Host می باشد. پس در کلاس B، می توان $۲^{۱۴}$ تا شبکه متفاوت داشت که در داخل هر شبکه $۲^{۱۶}$ تا Host قرار دارد یعنی ۱۶۰۰۰ شبکه متفاوت که در داخل هر یک می توان ۶۴۰۰۰، Node داشت.

در کلاس C سه بایت اول مربوط به شماره شبکه و بایت آخر مربوط به شماره Host می باشد. و ۳ بیت اول آنهم رزرو است پس در کلاس C می تواند $۲^{۲۱}$ شبکه داشت که در داخل هر شبکه $۲^{۲۸}$ Host وجود دارد.

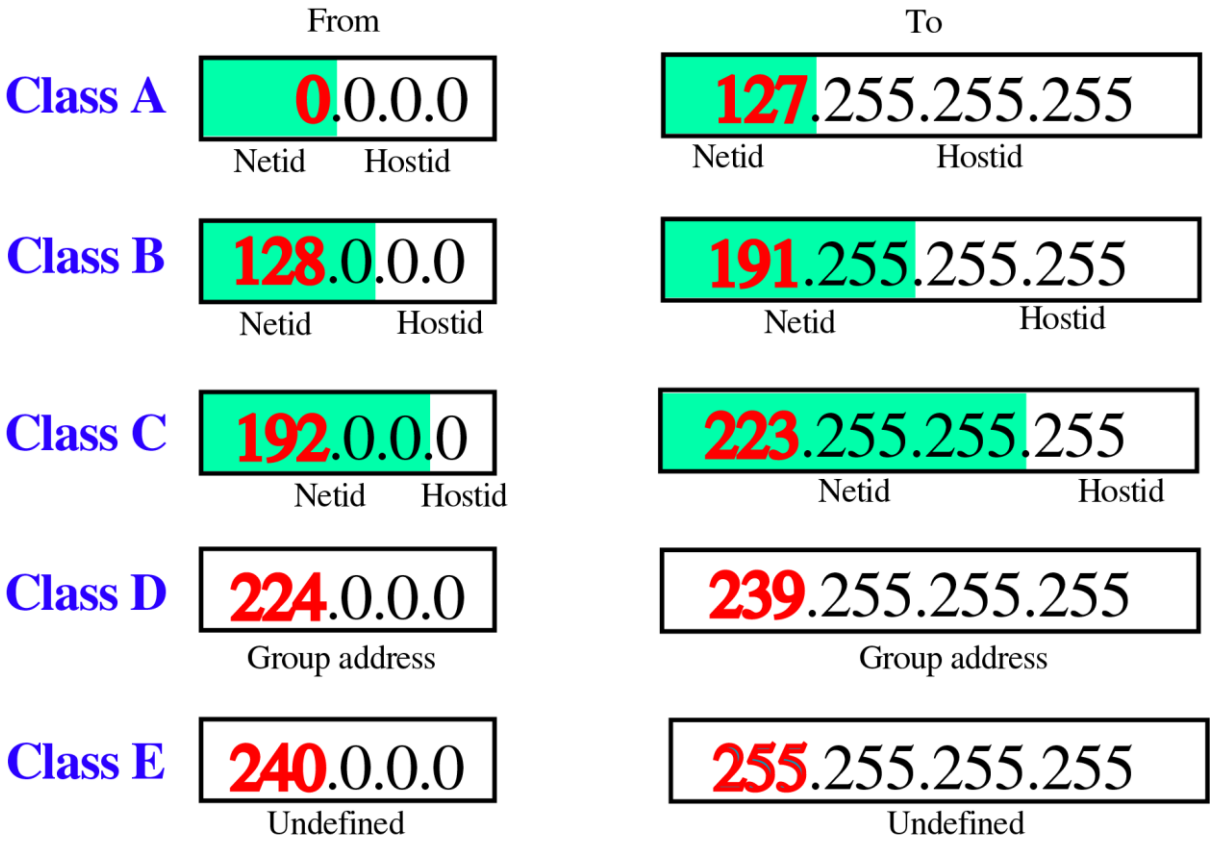
کلاس D یک کلاس multicast است کلاس های multicast، کلاس های با آدرس های گروهی می باشند که چند مدل دارد. Unicast آدرسی است که به یک شخص نسبت می دهیم، broad cast آدرسی است که به همه اعضا نسبت می دهیم. کلاس های multicast آدرسهایی هستند که یک گروه آنها را دریافت می کنند (یک گروه خاص) کلاسهای A و B اکثراً داده شده و تمام شده اند و در حال حاضر کلاس های C را به مشترکان عرضه می کنند.

محدوده کلاس های A از ۰,۰,۰,۰ تا ۱۲۷,۲۵۵,۲۵۵,۲۵۵ می باشد که برای تعیین این محدوده ها کافیست همه بیت ها را حداقل صفر و حداکثر یک کنیم که این محدوده ها به دست خواهند آمد. مثلاً در Byte ۱ داریم: ۰۰۰۰۰۰۰۰ که معادل صفر است و ۰۱۱۱۱۱۱۱ که معادل ۱۲۷ است. صفر انتهایی عدد رزرو شده توسط class type است که تغییر نمی کند.

کلاس B از ۱۲۸ شروع می شود. یعنی از ۱۲۸,۰,۰,۰ تا ۱۹۱,۲۵۵,۲۵۵,۲۵۵ را خواهیم داشت.

در کلاس C از ۱۹۲,۰,۰,۰ تا ۲۲۳,۲۵۵,۲۵۵,۲۵۵ را داریم.

در کلاسِ ۲۲۴,۰,۰,۰ تا ۲۵۵,۲۵۵,۲۵۵,۲۵۵ را خواهیم داشت.



شکل ۱۹۸: محدوده کلاس های آدرسی اینترنت

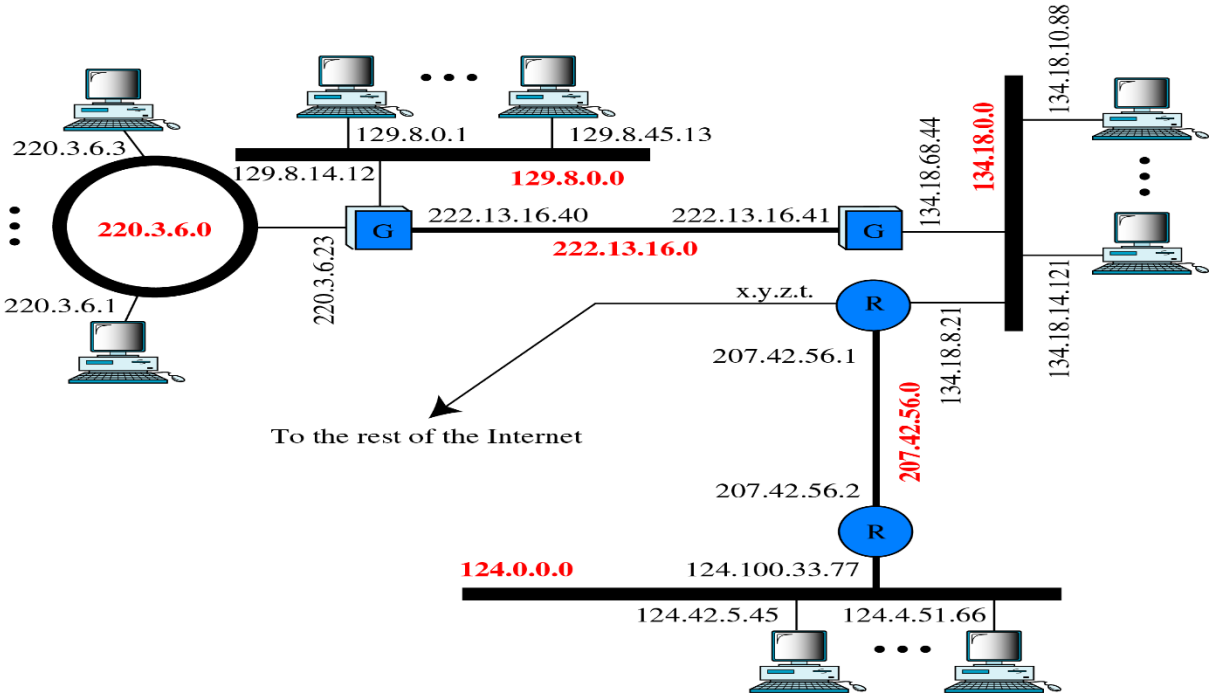
در ارتباط با کلاس D گفتیم که اطلاعات به صورت گروهی ارسال می شوند. در این کلاس یک مجموعه از Host ها آماده برای رسیدن اطلاعات خاص هستند. هر ماشینی که می خواهد وارد شبکه شود موقعیت آن بر اساس آدرس IP آن شناخته می شود که این آدرس ها Unic cast هستند و فقط برای یک نفر استفاده می شوند.

آدرس های Broad cast آدرسهایی است که همه دریافت می کنند و آدرسهای Multi cast متعلق به یک گروه است که قطعاً Routerها باید این موضوع را پشتیبانی کرده و اطلاعات را برای گروه خاص بفرستند.

کاربرد این آدرسها Multi cast در ویدئو کنفرانس ها می باشد که یک نفر می خواهد سرویس اطلاعات را به جمعی از کاربران بفرستد نه برای یک شخص خاص.

مثال:

فرض کندی شبکه ای داریم که در داخل آن نیز چندید شبکه وجود دارد که هر یک با یک توپولوژی خاص و با یک معماری خاص می باشند.



شکل ۱۹۹: Network and host Address

اولین بخشی که مطرح می شود، آدرسهای IP هستند و هر شبکه باید برای خود یک Class انتخاب کند و به هر Node خود یک IP اختصاص دهد. مثلاً یک شبکه کلاسی با آدرس ۲۲۰,۳,۶,۰ انتخاب می کند که این کلاس، کلاس C است. که این وظیفه مدیر شبکه است که یک کلاس انتخاب کند و بر مبنای آن به هر node خود یک آدرس IP نسبت دهد.

آدرس های IP، آدرسهای UNIC هستند و نباید دو شبکه از یک آدرس یکسیان استفاده کنند تا آدرسهای تکراری نداشته باشیم. پس برای اینکه آدرسهای تکراری نداشته باشیم، باید مدیر برای کل شبکه در نظر گرفته شود که ما از آن درخواست کلاس کنیم و از آدرسهای تکراری جلوگیری شود.

در شبکه اینترنت بحث آدرسهای Valid و Invalid وجود دارد. آدرسهای valid آدرسهای معتبرند که در شبکه اینترنت رایج می باشند ولی آدرس های Invalid آرسهای غیر معتبرند که در شبکه اینترنت وجود ندارند و کسانی که این آدرس ها را دارند در شبکه مخصوص خود آدرس دهی می کنند و به شبکه اینترنت کاری ندارند ولی از طریق مکانیزم هایی آنها را به آدرسهای اینترنت معتبر تبدیل می کند. پس تا اینجا هر شبکه یک کلاس انتخاب می کند و بر مبنای آن کلاس، به Nodeهای خود یک آدرس IP اختصاص می دهد.

از بین آدرسهایی که می توان اختصاص داد، چند آدرس وجود دارد که حق دادن این آدرسها به nodeها را نداریم:

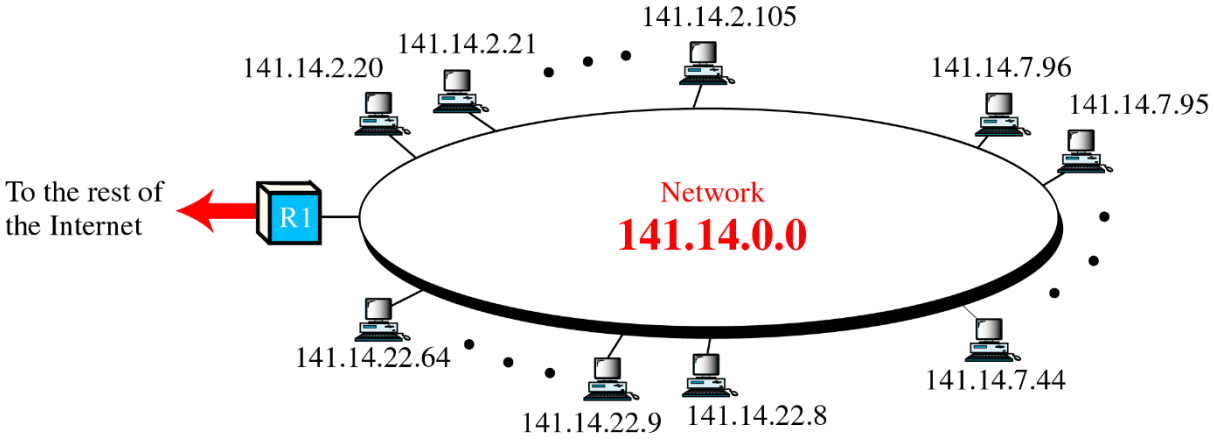
۱. اگر آدرس host صفر باشد، این آدرس مشخص کننده شماره شبکه است و حق نداریم این آدرس را به ماشین (node) نسبت دهیم. مثل ۲۲۰,۳,۶,۰
۲. اگر آدرس Host برابر ۲۵۵ باشد، حق دادن این آدرس به node را نداریم. این آدرس یعنی Broad cast داخل شبکه.

نکته هایی که باید در نظر داشت این است که Routerها آدرسهای Broad cast را از خود عبور نمی دهند چون می دانند که مربوط به داخل شبکه است.

۳. آدرس ۱۲۷,۰,۰,۱، آدرس loop Back یا local host می باشد که مربوط به خود ماشین است و packet ای که از این آدرس استفاده می کند به داخل ماشین باز می گردد و برای تست خود ماشین استفاده می شود.
۴. آدرس ۰,۰,۰,۰، آدرس hostای است که آدرس خود را نمی داند. بنابراین هر میزبانی که آدرس خود را نمی داند می تواند به عنوان آدرس فرستنده از آن آدرس استفاده کند و فقط اطلاعات را به آدرس مورد نظر می فرستد و اطلاعات برگشت پیدا نخواهد کرد.

Sub network

فرض کنیم شبکه ای داریم به صورت یک پارچه و یک کلاس از نوع B برای آن انتخاب می کنیم: ۱۴۱,۱۴,۰,۰



شکل ۲۰۰: a network with two levels of hicrarchy

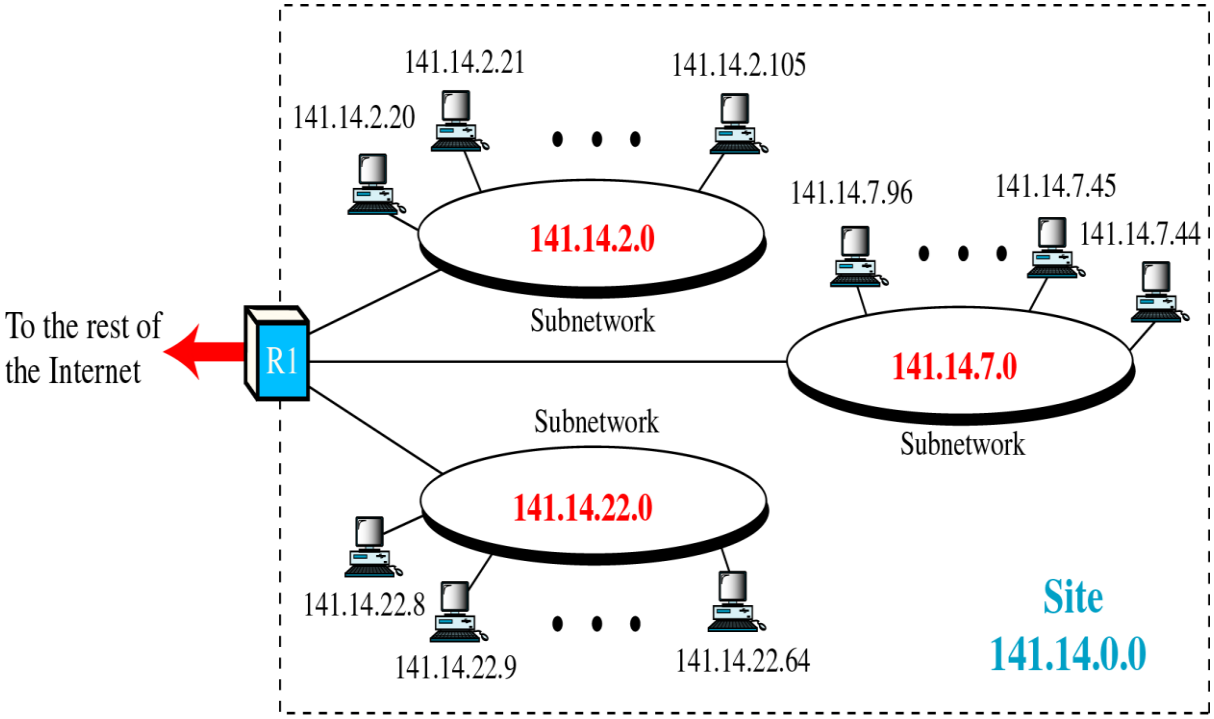
و طبق این کلاس به هر یک از nodeها یک IP داده می شود. در این کلاس می توان تا ۶۴k host داشت.

اگر این شبکه بخواهد با بیرون ارتباط برقرار کند از طریق یک Router بنام R این کار را انجام می دهد. این شبکه دارای Nodeهای زیادی می باشد که هر یک مربوط به یک شبکه خاص می تواند باشد. وقتی که یک packet به آدرس این شبکه (۱۴۰,۱۴,۰,۰) ارسال می شود، این packet را همه nodeها دریافت می کنند که با این کار اولا ترافیک شبکه خیلی بالا رفته است و ثانیا امنیت شبکه پایین می آید. لزومی ندارد که یک packet که مربوط به یک بخش خاص است را nodeهای همه بخش های یک سازمان دریافت کنند. بنابراین باید این مشکل را رفع نمود.

بخشی که برای حل این مشکل مطرح می شود، بحث Subneting می باشد. در این روش، کل شبکه به زیر شبکه هایی تبدیل می شود ولی کل شبکه در قالب همان آدرس ۱۴۱,۱۴,۰,۰ دیده می شود. در اینجا بایت سوم را هم به عنوان آدرس زیر شبکه ها در نظر می گیریم و شبکه را به بخشهای مختلف تقسیم نموده و از آدرسهای متفاوت برای آنها استفاده می کنیم.

بنابراین شبکه یکپارچه را به سه زیر شبکه با آدرسهای IP متفاوت تقسیم کردیم. حال اگر یک شخص Packet را به صورت همگانی (broad cast) بفرستد برای یک آدرس زیر شبکه خاص، فقط همان nodeهای مربوط دریافت می کنند در صورتی که در روش قبلی کل شبکه (۶۴۰۰۰) آنرا دریافت می کردند که ترافیک را خیلی بالا می برد و امنیت را پایین می آورد که این ۳ مشکل با این روش حل شده است. به این روش Sub netting گفته می شود که این بحث را می توان برای کلاس C انجام داد. در کلاس C می توان گفت که از ۸ بیت مربوط به host، ۳ بیت آنرا برای زیر شبکه در نظر می گیریم و ۵ بیت بعدی را برای host های داخلی آن شبکه در نظر می گیریم با این کار می توان ۶ زیر شبکه ایجاد کرد

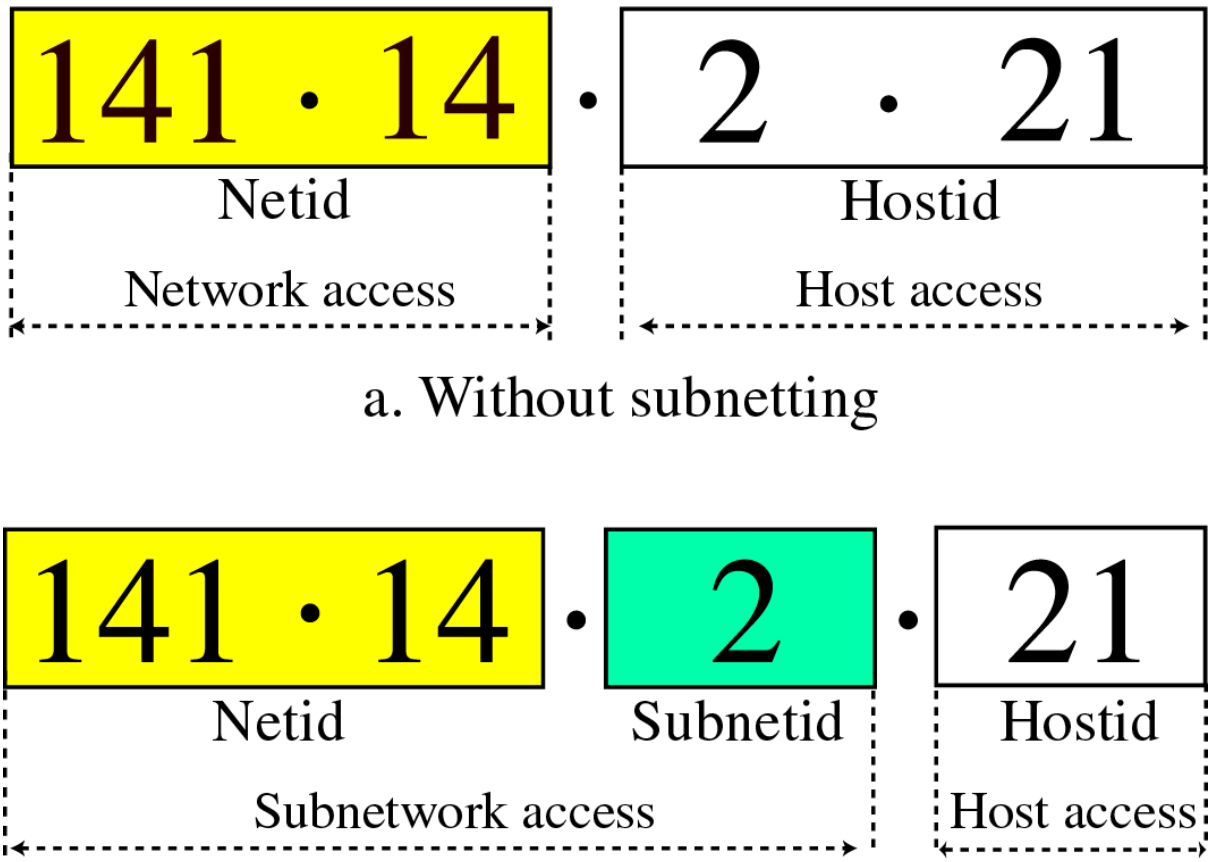
چون ۳ بیت برای زیر شبکه در نظر گرفتیم که یک حالت آن مربوط به Broad cast می باشد و یک حالت هم مربوط به شماره شبکه. (از ۸ حالت).



شکل ۲۰۱: a network with three levels of hicrarchy

با این دید که این ۶ زیر شبکه از بیرون به شکل یک شبکه می باشد و ترافیک ها را ایزوله می کند. وقتی از بیرون شبکه packetای فرستاده می شود، دقیقاً به آدرس node مورد نظر فرستاده می شود نه به آدرس مجازی که برای هر شبکه در نظر گرفته شده است.

همانطور که در شکل دیده می شود، در حالت a یعنی without sub netting، ۳ بایت اول به عنوان آدرس شبکه و ۳ بایت بعدی به عنوان آدرس host در نظر گرفته شده است.

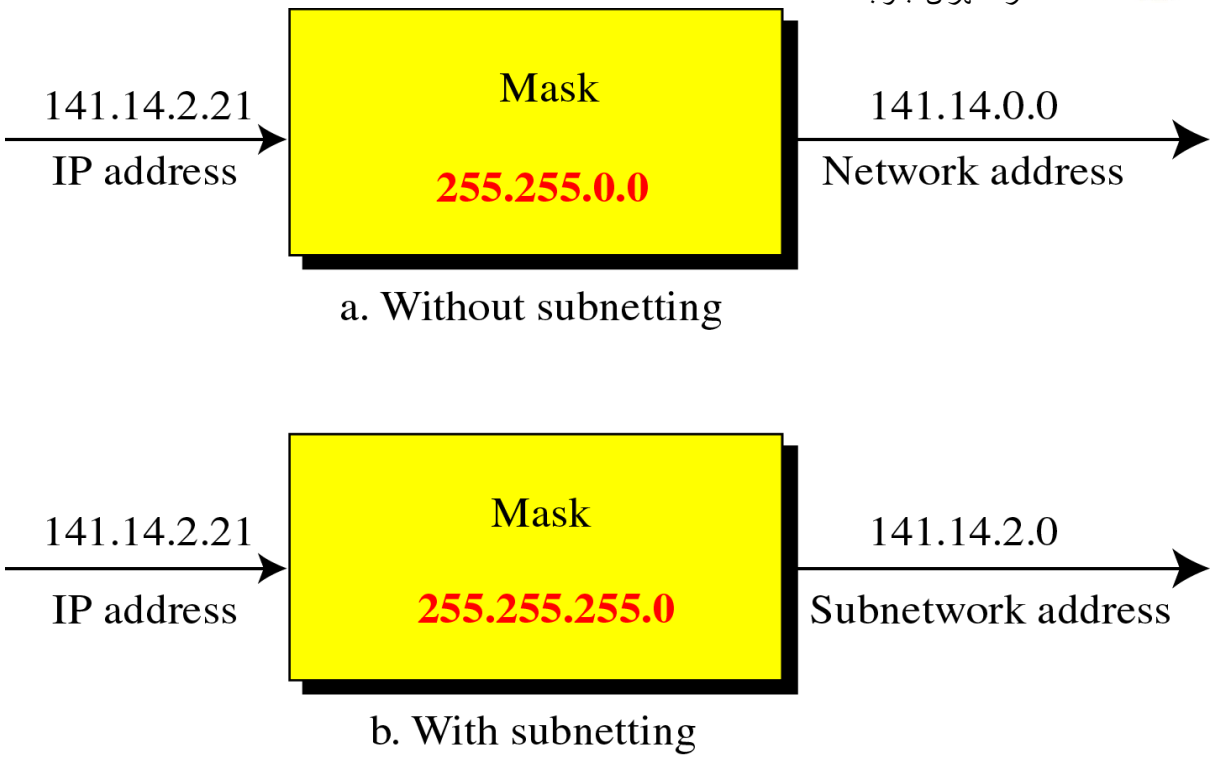


شکل ۲۰۲: آدرسها با و بدون subnetting

اما در حالت b یعنی with sub netting، ۸ بایت شوم به عنوان بیتهای sub netting در نظر گرفته شده است این حالت که سه بایت اول مربوط به شماره شبکه و بایت آخر مربوط به شماره host باشد در کلاس C دیده می شود. در حالیکه این کلاس، کلاس B است و خودمان به صورت مجازی بایت سوم را به عنوان زیر شبکه در نظر می گیریم که از بیرون به صورت کلاس B دیده می شود ولی در داخل زیر شبکه ها قرار دارند.

Subnet mask:

وقتی می خواهیم از پروتکل TCP/IP استفاده کنیم آنرا نصب کنیم. یک IP address از ما پرسیده می شود که باید یک کلاس را انتخاب کرده و IP address را از آن کلاس انتخاب کنیم و یک sub net mask پرسیده می شود. و همچنین در مورد Default Gateway پرسیده می شود Subnet Mask مشخص کننده Class و Subclass می باشد. اگر در یک کلاسی آدرس host را برابر صفر و آدرس شبکه را برابر ۲۵۵ قرار دهیم، آدرس Subclass Mask به دست می آید.



شکل ۲۰۳: Masking

برای اینکه تشخیص دهیم آدرس Packetای که ارسال شده است مربوط به شبکه هست یا نه کافیست آدرس packet مورد نظر IP Address را با Mask، AND کنیم که آدرس شبکه را به ما می دهد. اگر این آدرس در زیر شبکه وجود داشت مربوط به آن می باشد در غیر این صورت discard خواهد شد.

فرض کنیم آدرس فرستنده و گیرنده و subnet mask را به صورت زیر داشته باشیم:

فرستنده:

گیرنده:

Subnet mask: ۲۵۵,۲۵۵,۲۵۵,۰

اگر آدرس فرستنده را با آدرس subnet mask، AND کنیم آدرس ۱۹۳,۱۸۸,۲۰۴,۰ را خواهیم داشت که مربوط به شبکه داخلی خود فرستنده می باشد. در مورد گیرنده هم همین نتیجه به دست می آید. پس packet تولید شده مربوط به شبکه فعلی می باشد و آنرا دریافت می کند. ولی اگر packet مربوط به شبکه مورد نظر نبود، در این صورت به router فرستاده می شود که به آن default gatway گفته می شود و شبکه آن را قبول نمی کند. پس نکته شوم در پروتکل TCP/IP، Default gateway می باشد که به Routerای که packet را می گیرد گفته می شود.

در اینجا مشکلی وجود دارد و آن اینکه Router می داند که packet ای که ارسال شده است IP Address آن مربوط به زیر شبکه اش است. اما احتیاج به آدرس MAC دارد چون Data برای ارسال شدن باید به لایه Data link برود و لایه Data link هم برای فرستادن packet احتیاج به آدرس MAC فرستنده و گیرنده دارد (آدرس کارت شبکه باید مشخص باشد) که این مشکل باید بر طرف گردد.

Address Resolution:

برای حل این مشکل که در بالا مطرح شد، راه حل هایی ارائه شده است که از طریق این روش می توان مشکل آدرس MAC کارت شبکه گیرنده و فرستنده را برطرف نمود.

روش Look up Table:

یک router به دلیل اینکه به یک sub network متصل است (مثلا شبکه LAN) کافیست که یک جدول داشته باشد که در یک سمت آن IP Address را نگهداری کند و در سمت دیگر آدرسهای کارت شبکه را فقط این آدرسها محدود به داخل Sub netای می باشد که router به آن وصل است. که در این روش آدرس ها به راحتی قابل تشخیص می باشند.

مشکل این روش این است که اگر کارت شبکه بسوزد قطعاً Look up Table تغییر می کند که مدیر شبکه با تغییر هر آدرس، باید این جدول را تغییر دهد.

روش close from computation:

در این روش یک روش ریاضی مطرح می شود که از طریق این روش بتوان به راحتی از IP Address آدرس کارت شبکه را به دست آورد که مسلما این آدرس unic می باشد. بنابراین در این روش از طریق روشهای محاسباتی و از روی IP Address آدرس کارت شبکه به دست خواهد آمد که روش هایی نیز برای این کار مطرح شده است.

اما روش های پرکاربرد تری بنام ARP و RARP مطرح می شود.

Sending Message (ARP/RARP):

پروتکل های ARP و RARP مشکل ما را بر طرف می کنند. پروتکل ARP، مخفف Address Resolution Protocol می باشد و قادر است از IP Address آدرس MAC را به دست آورد.

در این روش Router یک ARP Request را به صورت Broad cast برای تمام اعضا Subnet می فرستد.

برای Broad cast فرستادن کافیست که وقتی Frame را تولید می کند در MAC Address گیرنده ۴۸ تا یک قرار دهد تا همه آنرا دریافت کرده و روی آن process انجام دهند.

یک واحد اطلاعاتی ARP تولید می کند و برای همه ارسال می کند و مثلاً آدرس ۱۲۸,۱۷۱,۱۷,۱۳ را اعلام کرده و می پرسد که این آدرس برای چه کسی است؟ که این در داخل پیام ARP Request وجود دارد. همه اعضا روی این Frame، process انجام می دهند چون آدرس Broad cast بوده است.

در مقابل، host ای که آدرس آن ۱۲۸,۱۷۱,۱۷,۱۳ می باشد، باید اعلام کند که این آدرس مربوط به اوست و یک ARP response تولید می کند که شامل آدرس MAC است که خود را معرفی کرده و آدرس کارت شبکه اش را ارسال می کند.

پروتکل ARP چون یک پروتکل است، واحد های اطلاعاتی آن دارای یک فرمت خاص می باشند ARP Requestها و ARP Responseها دارای فرمت مشخص و همچنین فیلد های مشخص می باشند.

بنابراین این پروتکل (RAP) به راحتی کار خود را انجام می دهد. چون هر host در داخل شبکه دارای یک IP واحد است و ممکن نیست که IP اشتباها ارسال گردد.

ویژگی خوب این روش این است که وقتی آدرس کارت شبکه ای تغییر کرد، دیگر نیازی به مدیر شبکه نیست که آدرس را update کند، بلکه خود آدرس در این روش update نگه داشته می شود.

علاوه بر این، این روش یک cache دارد که وقتی یکبار درخواست یک IP را کرد و Response آنرا دریافت نمود آدرس آنرا داخل cache نگهداری می کند و برای اینکه مشکل وجود رکورد بعد از سوختن کارت شبکه وجود نداشته باشد، یک Timer می گذارد و اگر بعد از مدتی هیچ مراجعه ای به این آدرس نشد، قطعاً این آدرس از شبکه حذف شده است و رکورد آنرا حذف می کند و اگر نیازی به آن آدرس داشته باشد، طبق همین روش یکبار می پرسد و دوباره یاد می گیرد.

بعد از اینکه Router آدرس MAC را به دست آورد آنرا به لایه Data link می دهد تا روی شبکه packet را بفرستد. بنابراین پیدا کردن آدرس کارت شبکه بر عهده لایه network آدرس کارت شبکه را پیدا کرده به لایه Data link می دهد تا Data link، Frame را ارسال کند.

اگر آدرسی مورد نیاز باشد، در این روش اگر در جدول آدرس مورد نباز وجود نداشت این روش انجام می شود یعنی از طریق ARP Request آدرس را بدست می آورد. در غیر این صورت از جدول که همیشه به صورت update نگه داشته می شود استفاده می کند.

*ARP تنها پروتکلی است که مستقیما روی خط قرار می گیرد و بسته بندی نمی شود.

پروتکل های دیگر لایه network:

از پروتکل های دیگر، می توان به پروتکل RARP اشاره نمود.

پروتکل RARP عکس پروتکل ARP است. در پروتکل ARP، آدرس IP را داشتیم و می خواستیم آدرس کارت شبکه را بدست آوریم ولی در اینجا آدرس کارت شبکه را داریم و می خواهیم آدرس IP را بدست آوریم.

برای دانستن IP این آدرس باید در جایی از حافظه Save شود. بنابراین ماشینهایی که Hard disk ندارند، نمی توانند آدرس IP را در جایی ذخیره کنند و این ماشین ها IP خود را نمی دانند.

در پروتکل RARP نیز یک Frame به شبکه ارسال می شود و می پرسد که این آدرس کارت شبکه برای چه کسی است و IP آدرس خود را اعلام کند. از طریق این روش به سادگی به آدرس IP مورد نظر خود دست می یابد.

ICMP:

پروتکل دیگر که بسیار معروف می باشد، ICMP است که مخفف Internet control message protocol می باشد. گفتیم که پروتکل IP که در لایه network مطرح شد، پروتکل غیر قابل اطمینان است که امکان رخداد خطا در آن وجود دارد.

استاندارد ICMP، استانداردی است که مدیریت کنترل Messageها را بر عهده دارد. ئاحد های اطلاعاتی ICMP در اپلت های IP جای گرفته اند و واحد های مستقلی را بطور مستقیم تولید نمی کنند. دیدیم که ARP، یک واحد اطلاعاتی مستقل تولید می کرد در صورتیکه ICMP چنین نیست. در واقع در ICMP ئاحد اطلاعاتی داخل یک packet قرار می گیرد و IP آنرا حمل می کند. اطلاعات ICMP در بخش information پروتکل IP حمل می شود.

ICMP برای اعلام خرابی بین host با host و بین host با router (انواع پیام) تعریف شده است.

پس ICMP، پروتکل IP را قدری مطمئن تر می کند چون حداقل اعلام خرابی می کند. باید توجه داشت که باید توجه داشت که ERROR CORRECTION بر عهده ICMP نمی باشد و فقط خطا را اعلام می کند.

حال باید دید واحد های اطلاعاتی که توسط ICMP برای اعلام خرابی تولید می شود، چه نوع خرابی هایی را اعلام می کند.

Message Type:

Query: زمانی که ما در شبکه هستیم و می خواهیم بدانیم شخص مورد نظر ما نیز online است یا نـه از دستوری بنام دستور ping استفاده می کنیم که اعلام می کند طرف مقابل در حال حاظر در شـبکه وجـود دارد یا خیر.

Source Quench: نوعی دیگر از Messageهایی که ICMP تولید می کنـد، Source Quench مـی باشد. در این پیغام که یـک Message خـاص اسـت، حـالتی از Flow Control اسـت کـه توسـط ایـن message (که از طرف فرستنده ارسال می شود) اعلام می شود که سرعت ارسال packet هـا بایـد کـم شود و وقتی یک host یک packet را ارسال می کند از طریق این پیغام سرعت packet پایین آورده می شود. اگر باز هم یک Source Quench دریافت کند باز هم سرعت پایین آورده می شود تا وقتی که دیگر Source quench ارسال نشود و در این لحظه دوباره سرعت آرام آرام بالا می رود تا بـه مقـدار اولیـه اش برسد.

IP کاری به این موضوع ندارد و این ICMP است که این کار را انجام می دهد.

Redirection: یکی دیگر از Messageهایی که ICMP تولید می کند Redirection نام دارد.

در اینجا زمانی که یک روتر خراب است یا مشکلی دارد و نمی تواند اطلاعات را ارسـال کنـد از طریـق ایـن پیغام اعلام می کند که از یک روتر دیگر استفاده کنید.

مثلاً روتر B از کار افتاده است و از طریق این پیغام اعلام می کند که از روتر A استفاده کنید.

این سرویس را هم پروتکل ICMP می دهد که برای اعلام خرابی به کار می رود بنابراین از طریق این روش ها پروتکل IP معتبر تر و مطمئن تر می شود.

IGMP:

این پروتکل نیز از دیگر پروتکـل هـای لایـه شـبکه مـی باشـد و مخفـف Internet Group Message Protocol می باشد و این امکان را به یک host می دهد که بتواند اطلاعات خود را بـرای مجموعـه ای از افراد بفرستد. در حالت عادی ارسال اطلاعات از host به host است و از یک مبدا به یک مقصد می رود.

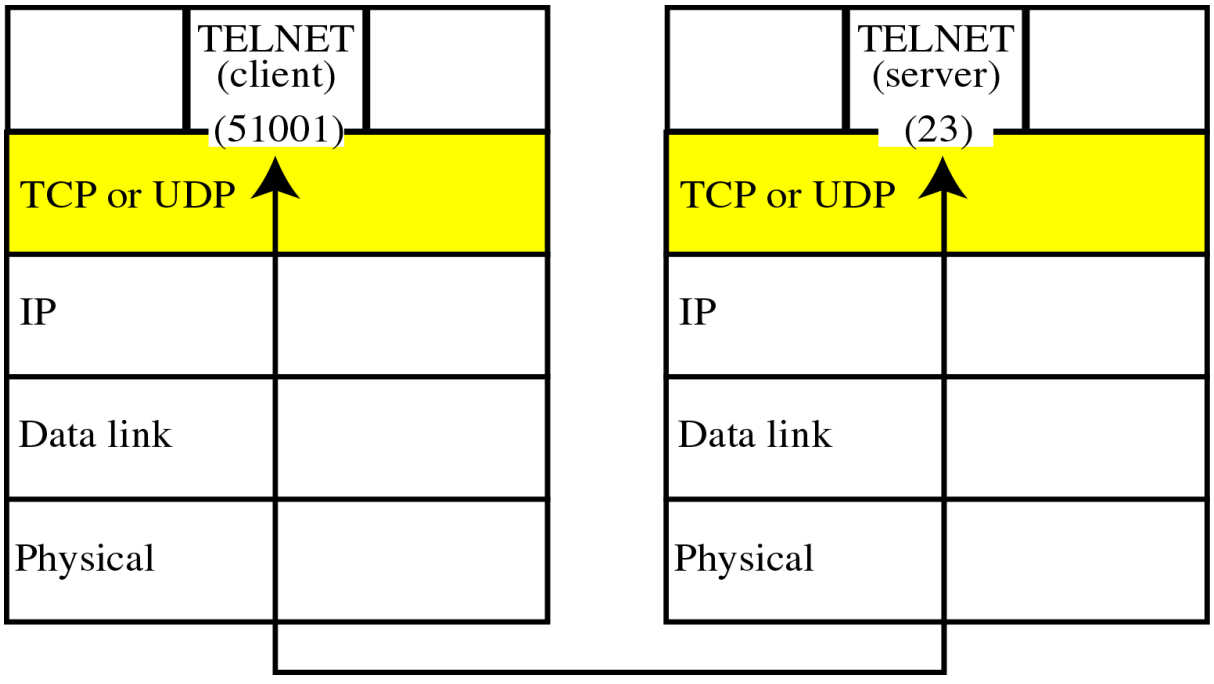
در ارسال اطلاعات پروتکل IGMP از آدرسهای Multi cast استفاده می شود که از طریق ایـن آدرس هـا یک host اطلاعات خود را به چندین host دیگر میفرستد.

اطلاعاتی که از آدرس multi cast استفاده می کنند از طریق روتری که این اطلاعات را دریافت مـی کنـد به تک تک اعضا گروه فرستاده می شود. کاربردهای این پروتکل در ویدئو کنفرانس و ارسال جـداول روترهـا می باشد. چون ارسال اطلاعات در IGMP به صورت Multicast است و در ویدئو کنفرانس و جداول روتـر

ها هم ما نیاز داریم که اطلاعات را به چندین نفر ارسال کنیم به همین دلیل از ایـن پروتکـل اسـتفاده مـی شود. پروتکل دیگری به نام Wins وجود دارد که این پروتکل هم از پروتکل IGMP برای ارسال داده هـای خود استفاده می کند.

لایه Transport:

چهارمین لایه از لایه های TCP/IP لایه Transport می باشد که دارای دو پروتکل بسیار معـروف TCP و UDP می باشد.



شکل ۲۰۴: لایه Transport

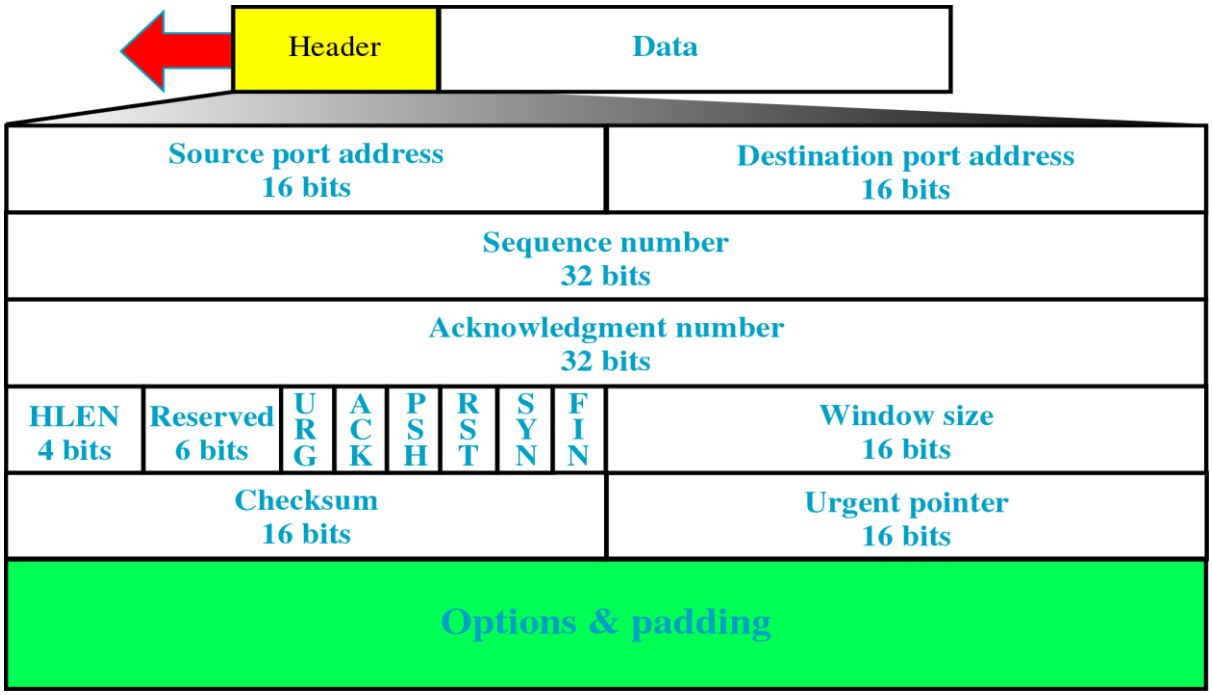
پروتکل TCP را می توان از پروتکل های مطمئن به حساب آورد. به دلیل اینکه همانطور که در شکل زیر دیده می شود، هر بسته ای که ارسال می شود به ازای آن ACK صادر می شود پس این پروتکل مطمئن است.

سرعت را در مقابل اطمینان از دست می دهیم، ولی در جایی که صحت ارسال اطلاعات برای ما مهم باشد، بسیار روش خوبی است. در کاربرد هایی که اطمینان مهمتر از سرعت است از TCP استفاده می کنیم.

TCP Segment Format:

هر سگمنت TCP از دو بخش Data و Header تشکیل شده است که قسمت Header آن شامل بیتهای کنترلی زیر می باشد:

Source port Address و Destination port Address که هر یک ۱۶ بیتی است آدرس Application فرستنده و گیرنده را مشخص می کنند. چون لایه Application در بالای لایه Transport قرار دارد و لایه Transport برای فرستادن اطلاعات خود به آدرس Application فرستنده و گیرنده نیازمند است که مشخص کند چه Applicationی در مبدا با چه Applicationی در مقصد می خواهد ارتباط بر قرار کند.



شکل ۲۰۵: TCP Segment Format

فیلد بعدی، قیلد Sequence number است که ۳۲ بیتی است. مشکلی که مطرح می شود این است که اگر حجم اطلاعات ارسالی ما زیاد باشد، این اطلاعات باید به واحد های کوچکتر شکسته شوند، که فیلد Sequence number برای آن در نظر گرفته شده است. فیلد Sequence number یک عدد random است (آدرس شروع، یک عدد random می باشد) که توسط فرستنده مشخص می گردد. (به عناون شماره ترتیب واحد اطلاعات)

فرض کنید واحد های اطلاعاتی ما شکسته شده است:

TCP_PDU	۱	۷۹	|	initial Sequence number
TCP_PDU	۲	۸۰	|	۳ octets in datafield
		۸۱	|	
		۸۲	|	
TCP_PDU	۳	۸۳	|	۲ octets in data field
		۸۴	|	

به اولین قسمت شماره ترتیب random، ۷۹ داده ایم. به قسمت دوم چون ۳ تا هشت تایی با ۳ بایت است ۳ شماره ترتیب ۸۰، ۸۱، ۸۲ داده شده است و ... سپس یک عدد Random اختصاص داده می شود و بر مبنای حجم اطلاعاتی که اضافه می شود، به عدد Random یک واحد اضافه می شود. که در اینجا عدد Random برابر ۷۹ بوده است و چون یک بایت است عدد بعدی ۸۰ می شود و برای قسمت سوم چون قسمت قبلی سه بایت بوده است از ۸۳ شروع می شود.

این کار به دلیل این است که گیرنده بتواند از روی این شماره های ترتیب، واحد اطلاعاتی را دوباره باز سازی کند.

در اینجا بحث Flow control مطرح می شود که فیلد Acknowledgement که ۳۲ بیتی است در نظر گرفته شده است. هر واحد اطلاعاتی که ساخته می شود و ارسال می گردد، به ازای هر کدام می بایست یک ACK صادر کند.

اگر مثال قبلی را در نظر بگیریم، فرستنده، واحد اطلاعاتی ۷۹ را ارسال کرده است. گیرنده پس از دریافت آن ۸۰ ACK را می فرستد که به معنای ارسال صحیح ۷۹ می باشد. پس از ارسال ۸۰ و ۸۱ و ۸۲ گیرنده ۸۳ ACK را می فرستد و ...

TCP_PDU	۱	۷۹	|	initial Sequence number

TCP_PDU	۲	۸۰	|	۳ octets in datafield
		۸۱	|	
		۸۲	|	
TCP_PDU	۳	۸۳	|	۲ octets in data field
		۸۴	|	

فیلد HLEN که طول header را مشخص می کند که ۴ بیتی است که ماکزیمم آن می تواند (۴*۱۶) ۶۰ بایت باشد.

فیلد بعدی فیلدی به نام Reserved است که ۶ بیتی می باشد و استفاده نمی شود.

فیلد URG اگر set شود، اعلام می شود که داده ها اضطراری هستند.

بیت ACK مشخص می کند که واحد اطلاعاتی که در حال ارسال شدن است ACK است نه Data

بیت Psh اگر set شود اعلام می کند که اطلاعات در گیرنده نباید در لایه Transport بافر شود و بلافاصله باید به لایه بالاتر ارسال شود. زمانی که ما حداقل تاخیر را در بعضی کاربردها نیاز داریم و سرعت مهم است این بیت را set می کنیم.

بیت Rst برای reset کردن ارتباط استفاده می شود.

بیت Syn که با بایت Fin & Ack ادغام می شود برای ساختن way handshake ۳ است در اینجا فیلد Ack و Syn را هر دو با هم set می کند و در Connection Confirm فیلد syn&fin را با هم set می کند تا بتواند way handshake ۳ را پیاده سازی کند (هم برای ارتباط هم برای قطع ارتباط).

فیلد بعدی window size است که برای پیاده سازی sliding window می باشد و ۱۶ بیتی است و طول پنجره ای که گیرنده می تواند دریافت کند را مشخص کند را مشخص می کند. (تعداد بایتهایی که امکان دریافت آن وجود دارد مشخص می کند.)

فیلد Check sum که ۱۶ بیتی است و برای کشف خطا در کل واحد اطلاعاتی یعنی هم header و هم Data

فیلد urgent pointer که ۱۶ بیتی است. اگر در بیت URG اعلام کرده باشیم که داده ها اضطراری است در این فیلد آدرس داده اظطراری را مشخص می کنیم. که در واقع آدرس انتهای داده اضطراری و ابتدای داده عادی را مشخص می کند.

خواص کلی پروتکل TCP:

- Full error correction
- Flow control

بحث کشف خطا وجود دارد

- Need open close process

احتیاج به پروسسها Open و close دادر (بحث ۳ way handshake)

- Most acknow ledgement each TCP-pdu

برای واحد های اطلاعاتی آن باید Acknowledge صادر شود

- Please heavy laad on sending and Receiving host

LOAD زیادی در سرورهای فرستنده و گیرنده به وجود می آورد به دلیل بحث Flow control

- Adds to network Traffic overhead

ترافیک شبکه را بالا می برد چون روی هر واحد اطلاعاتی آن باید ACK صادر شود.

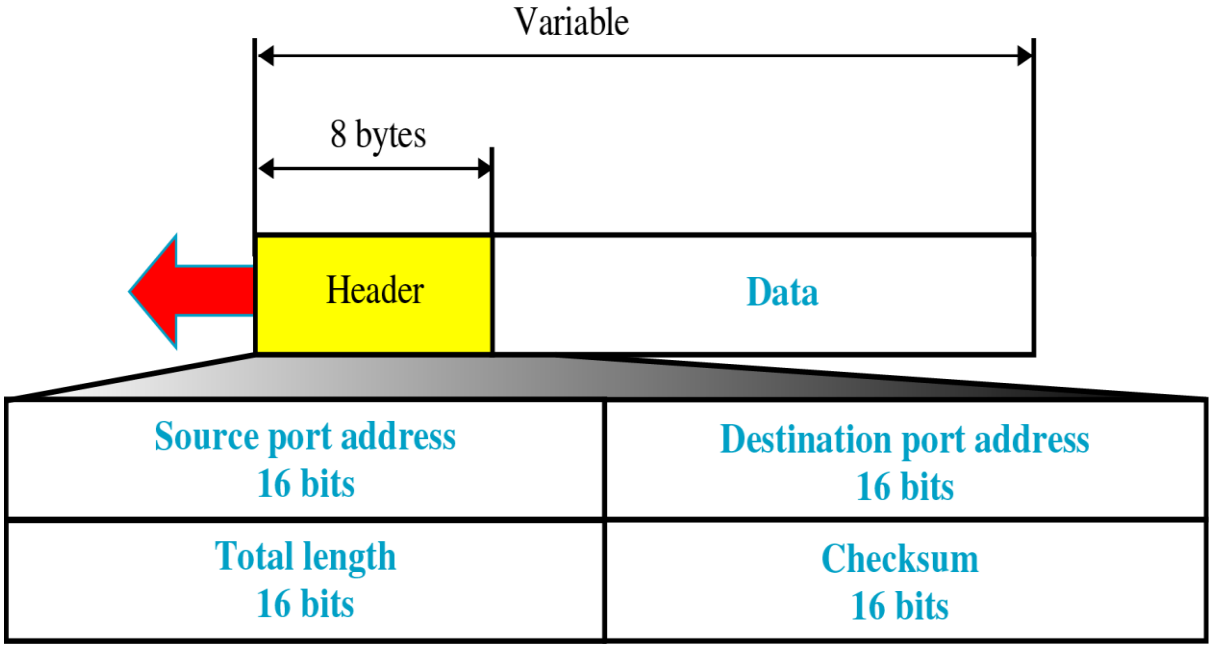
- CREATE latency in delivery Application PDUS to application program

به دلیل پیاده سازی ACK تاخیر هم در اینجا دیده می شود.

ولی در مقابل TCP را به عنوان یک پروتکل مطمئن در اختیار داریم. در کاربردهایی که احتیاج به اطمینان داریم از TCP استفادهه می کنیم اما زمانی که سرعت برای ما مهم است از این پروتکل استفاده نمی کنیم.

UDP Datagram Format:

در کنار TCP، پروتکل UDP قرار دارد و می توانیم به جاری TCP و UDP برای بسته بندی اطلاعات استفاده کنیم. بنابراین این موضوع در اختیار خود ما می باشد که در لایه Transport، پروتکل TCP اطلاعات ما را بسته بندی کند (اطلاعات لایه Application) یا UDP. قطعاً اگر TCP این کار را انجام دهد سرعت کند خواهد شد ولی مطمئن تر است.



شکل ۲۰۶: UDP Datagram Format

UDP پروتکلی است غیر قابل اطمینان و شبیه به IP بنابراین UDP پروتکلی است CONNECTION LESS در مقابل connectionoriented که در connectionoriented درخواست برقراری ارتباط و اعلام آمادگی از طرف گیرنده وجود دارد ولی در UDP این مباحث وجود ندارد و مستقیم اطلاعات را روی خط می فرستد بنابراین نه Errorcorrection و نه Flow control هیچکدام را ندارد.

در مقابل چون حجم واحد اطلاعاتی آن کم است، Processای روی آن (به آن صورت) انجام نمی شود و ترافیک شبکه را پایین می آورد و تاخیر زمانی آن هم کم می باشد چون بحث Flow control در اینجا وجود ندارد.

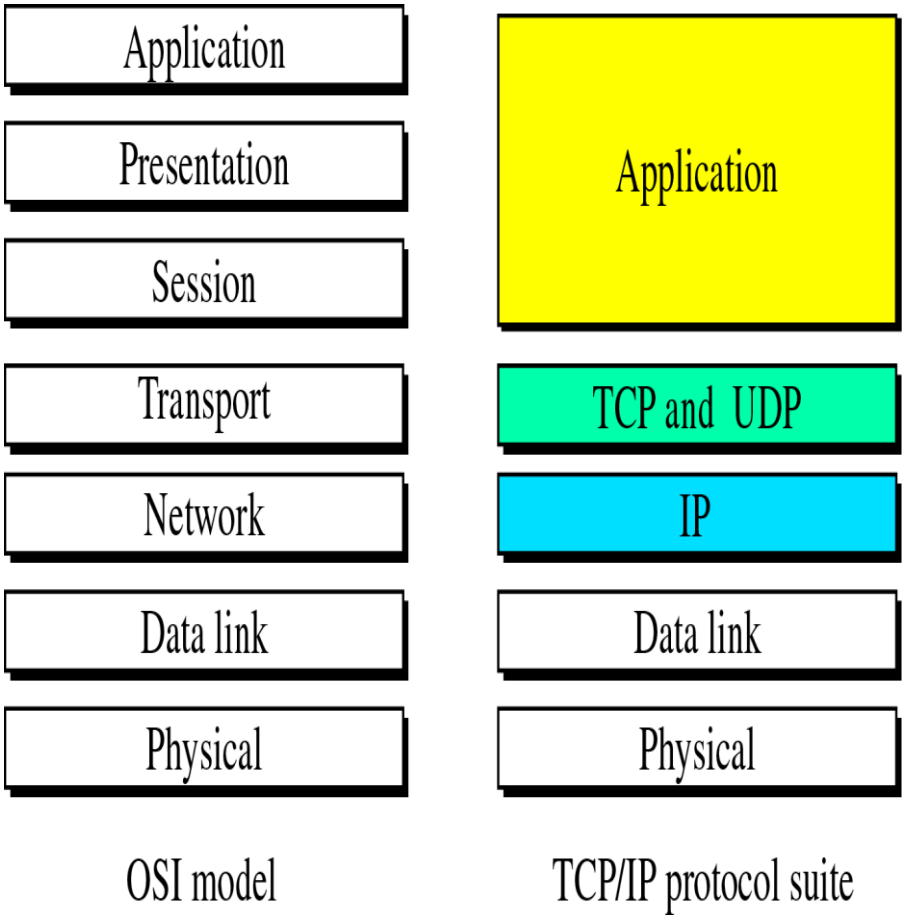
در فرمت اطلاعات UDP نیز همانطور که دیده می شود یک بخش Header و یک بخش Data دیده می شود که حجم اطلاعات Header آن خیلی کم می باشد و فقط دارای آدرس Application فرستنده و گیرنده، طول واحد اطلاعاتی و check sum می باشد. چیزی از Flow control و Error control در آن دیده نمی شود. به دلیل اینکه پروتکل UDP پروتکلی است ساده که بار ترافیکی را خیلی پایین می آورد.

فیلد CHECK SUM، چکی روی header انجام می دهد تا تشخیص دهد که Header سالم است یا نه. شکستن اطلاعات در این پروتکل نیز دیده نمی شود.

کاربرد پروتکل UDP در Network Management protocolها، جاهایی که از بین رفتن اطلاعات مهم نباشد، جاهایی که ترافیک بالا مشکل ساز است و ترافیک کم نیاز است و زمانی که voice داریم و بحث voice خیلی حیاتی می باشد، استفاده می شود.

لایه Application:

لایه Application، معادل با ۳ لایه Application و presentation و session می باشد.



شکل ۲۰۷: مقایسه مدل OSI و TCP/IP

در این لایه تنوعی از سرویس های متفاوت مطرح است. از قبیل: DNS,Electronic mail, FTTP, WWW و ...

مقایسه برخی پارامتر ها در سرویس های مختلف:

می خواهیم ببینیم در این سرویس هایی که استفاده می کنیم، چه چیزهایی مهم است و چه چیزهایی مهم نیست و در کدام لایه می توان از این سرویس ها استفاده کرد. در File Transfer یا انتقال پرونده Data نباید از دست برود. پهنای باند elastic است و می تواند تغییر کند. انتقال پرونده Time sensitive نیست چون در انتقال پرونده صحت از سرعت مهم تر است.

در E-mail ، Data نباید از دست برود چون مهم است. پهنای باند می تواند متغییر باشد و Time sensitive هم نمی باشد.

در web document یا صفحات وبی، Data loss آن بستگی به نوع Data دارد. پهنای باند متغیر است و Time sensitive هم نمی باشد.

در Real Time Audio/video ، data ممکن است loss شود، پهنای باند آن برای Audio و ویدئو هر یک مشخص شده است و Time sensitive می باشد و تا ۱۰۰ میلی ثانیه جا دارد. بقیه کاربرد ها هم به همین صورت طبق شکل بالا می باشد.

حال باید دید چه پروتکل هایی برای حمل این کاربردها استفاده می شوند:

در e-mail پروتکل حمل داده ها STMP می باشد. در e-mail گفتیم که داده ها برای ما مهم هستند. پس در لایه Transport توسط پروتکل TCP بسته بندی می شوند.

در مورد Remote Terminal Web و File Transter اطلاعات برای ما مهم هستند و باید از TCP استفاده شود. در streaming multi media و remote file server بستگی به کاربرد آن از TCP یا UDP استفاده می شود. در INTERNET TELEPHONY عموما از UDP استفاده می شود. چون در اینجا سرعت برای ما مهم است نه Data.

Internet history:

در سال ۱۹۶۱، آقای klien rock تئوری شبکه های packet switching را مطرح کرد. که کارایی شبکه های packet switching خیلی بیشتر از شبکه های circuit switching است.

در سال ۱۹۶۴ آقای Baran یک شبکه Packet switching در بحث نظامی را مطرح کرد.

در سال ۱۹۶۷ پروژه ARPANET بر مبنای این ایده شکل گرفت.

در سال ۱۹۶۹ اولیند ARPANET عملیاتی به وجود آمد (یعنی اولین NODE ای که بر مبنای این پروژه تحقیقاتی تعریف شده بود به وجود آمد)

در سال ۱۹۷۰ شبکه ماهواره ای به نام ALOHANET در هاوایی به وجود آمد (برمبنای پروژه ARPANET)

در سال ۱۹۷۲ یک Demo از پروژه ARPANET داده شد و پروتکلی که برای آن طراحی شد پروتکل NCP بود و اولین برنامه E-mail نیز در این سال طراحی شد. در سال ۱۹۷۲ شبکه ARPANET دارای ۱۵ node بود.

در سال ۱۹۷۳ آقای Metcalfe تزی برای پروژه های شبکه اترنت داد که پروژه وی پروژه PHD بود.

در سال ۱۹۷۴ آقای cref و kahn معماری برای ارتباطات internet working به وجود آوردند. چون nodeها به مراتب تعدادشان زیاد می شد و هر کدام از یک شبکه و پروتکل خاص پیروی می کردند.

در دهه ۱۹۷۰ معماری های دیگری هم در مقابل ARPANET به وجود آمد. مثل شبکه های SNA و ... و در اینجا بود که بحث شبکه های switching مطرح شد.

در سال ۱۹۷۹ تعداد nodeهای ARPANET به ۲۰۰ node رسید.

در سال ۱۹۸۳ با شبکه های متفاوت مواجه شدند و به دلیل اینکه هم ارتباطات بین شبکه ای را تعریف کنند و هم یک پروتکل استاندارد به وجود آورند، پروتکل TCP/IP به عنوان پروتکل استاندارد شبکه ARPANET تعریف شد.

در سال ۱۹۸۲ پروتکل SMTP تعریف شد که پروتکلی برای تبادل نامه های الکترونیکی بود.

در سال ۱۹۸۳ DNS یا domain name system مطرح شد که ترجمه بین نامه های اینترنتی و ip Addressها را انجام می داد.

در سال ۱۹۸۵ پروتکل FTP تعریف شد که پروتکلی برای انتقال پرونده ها بود.

از سال ۱۹۷۲ تا ۱۹۸۸ شبکه ها روز به روز شد پیدا کردند و انواع گوناگون شبکه به وجود آمد.

در سال ۱۹۹۰ شبکه ARPANET عملا منحل شد (به دو بخش تحقیقاتی و نظامی تقسیم شد)

در سال ۱۹۹۱ شبکه ای به نام NSF جای شبکه ARPANET را گرفت.

در سال ۱۹۹۰ بحث پروتکل های Hyper Text مطرح شد و همچنین بحث HTML

در سال ۱۹۹۴ بحث Browserهایی مثل Mosiac مطرح شد.

در دهه ۱۹۹۰ چیزی حدود ۵۰ میلیون host در شبکه اینترنت وجود داشت و Backbone اصلی اینترنت به Gbps ۱ ارتقا یافت.

پس طبق تاریخچه اینترنت، اینترنت از چیزی حدود ۴ یا ۵ node شروع شد تا اینجا گسترش پیدا کرد و به شبکه NSF رسید که شبکه NSF ، Backbone اصلی شبکه اینترنت است. وقتی که تعداد nodeها اینقدر افزایش پیدا کرد، دیدن که خودشان به عنوان provider می توانند به تمام nodeها شرویس دهند و این کار را به یکسری ISPها یا local ISPها محول کردند.

و به چند قسمت آنها را تقسیم بندی کردند NBP یا National Backbone Provider به عنوان Backbone با سرعت بسیار بالا به هم وصلند.

یک سری Regional ISP مطرح شد که ارتباط برقرار کردند. Local ISP از طریق Routerها به Regional ISP وصل می شود که مجموعه ای از Routerها می باشند و Regional به Backbone اصلی اینترنت وصل است.

همانطور که حفظ کردن آدرس های کارت شبکه کار مشکلی بود، حفظ کردن IP address ها هم مشکل است. برای حل مشکل از نامهای اینترنتی یا fully qualified domain name یا به اختصار fqdn استفاده می شود.

الگوی آن به صورت روبرو می باشد: someware domain

که someware هر چیزی می تواند باشد ولی domainها خاص هستند و هر domain حوزه فعالیت یک اسم خاص را مشخص می کند (عموماً)

آدرس های e-mail نیز دارای یک الگوی ثابت می باشند:

Local host@domain name

که domain name در اینجا نام سرویس دهنده پست الکترونیکی می باشد.

Domains:

چندین مدل domain وجود دارد:

Com: برای شرکت های تجاری بکار می رود.

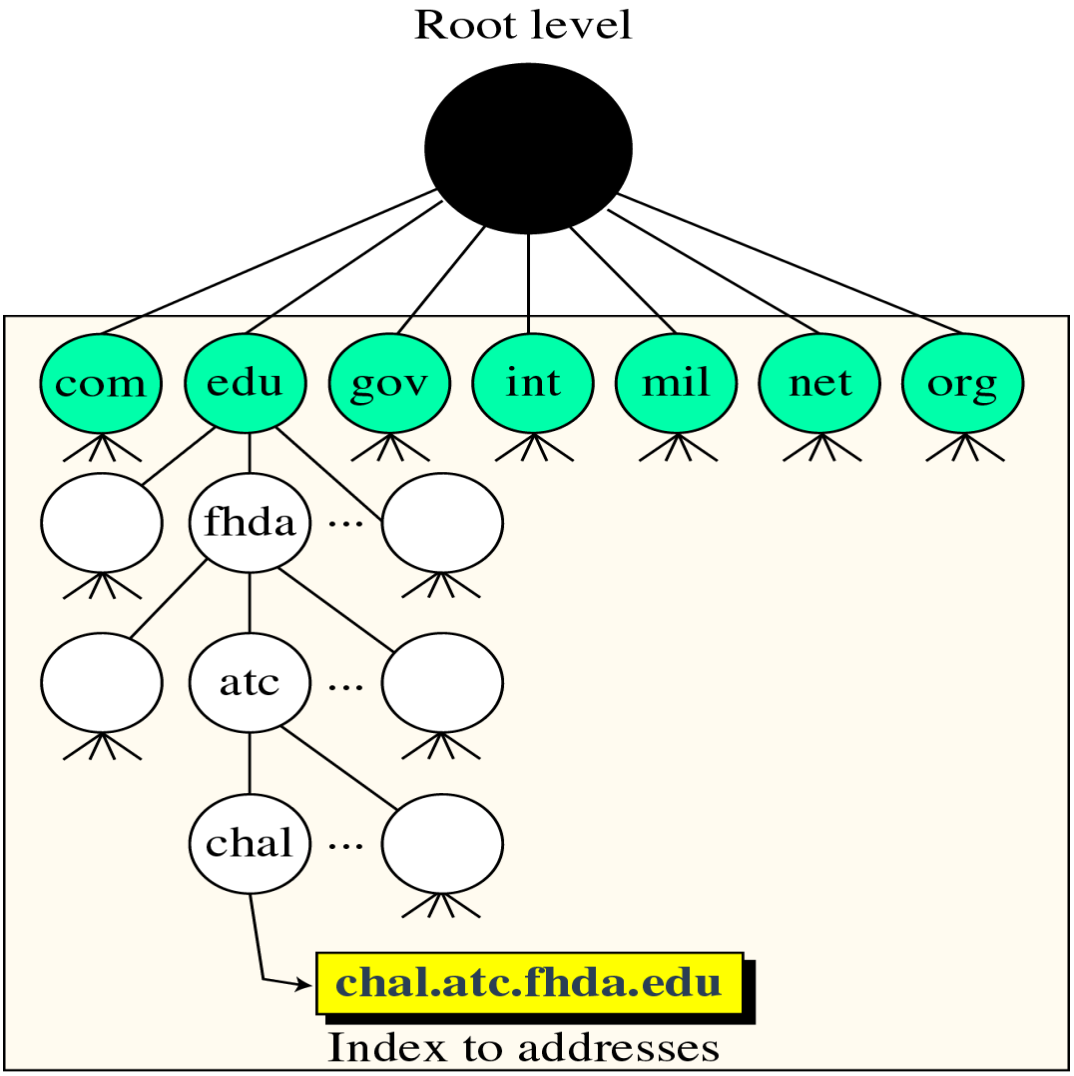
Edu: برای شرکت های آموزشی به کار می رود.

Gov: برای کاربرد های دولتی به کار می رود.

Org: برای سازمان ها به کار می رود.

Int: برای کاربرد های بین المللی است.

Net: برای کاربرد های شبکه و ...



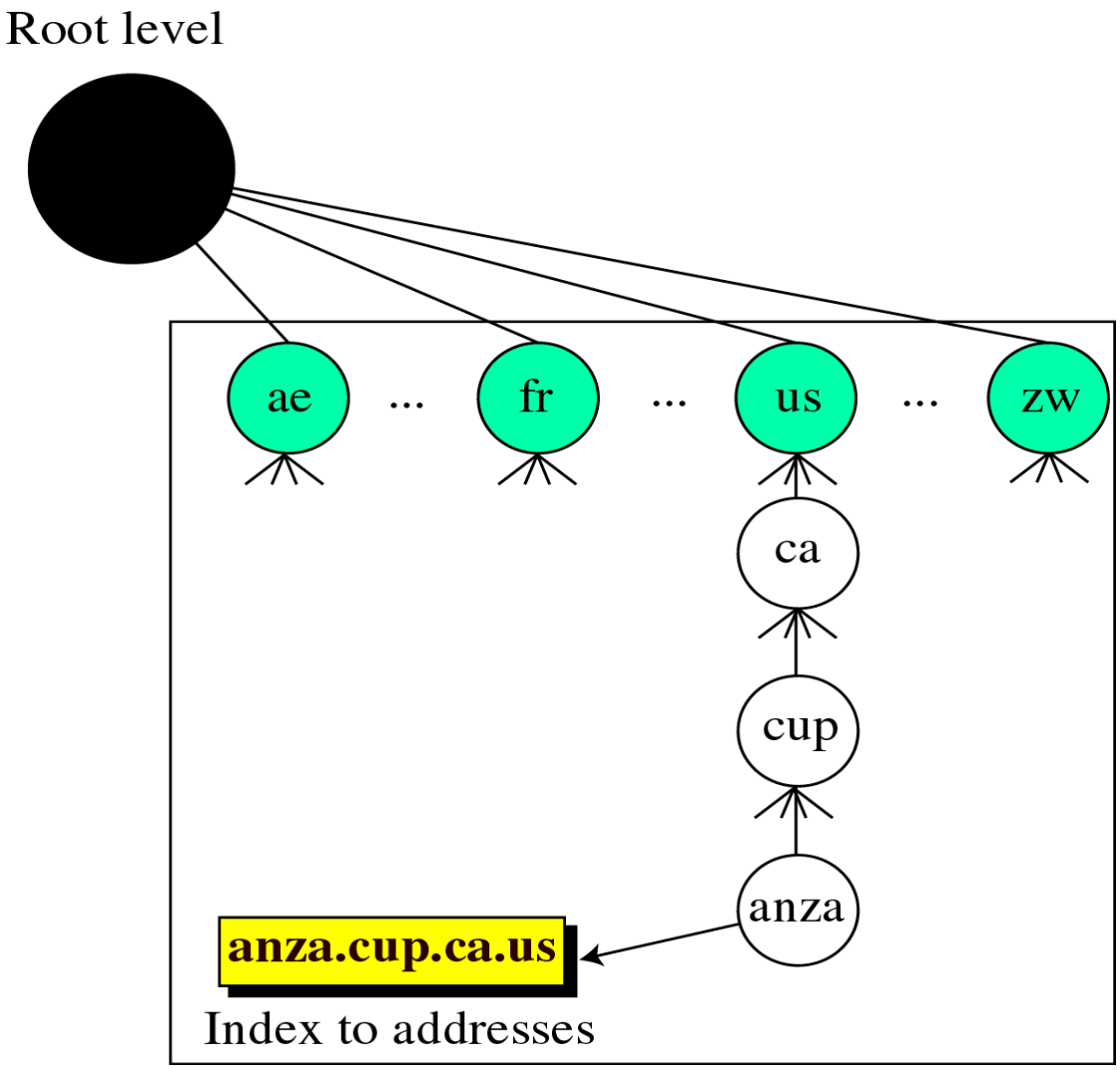
شکل ۲۰۸: Generic Domain

هر domain دارای یک root می باشد که ممکن است sub domainهایی هم داشته باشند. مانند مثال بالا: domainها از راست به چپ معنی می شوند.

Domainهای جغرافیایی نیز وجود دارند. هر کشور دارای یک Domain جغرافیایی خاص خود می باشد. به عنوان مثال: ایران ir و فرانسه FR و آمریکا با US و ... در اینجا هم domain ها می توانند subdomain داشته باشند.

بنابراین نامهای اینترنتی به این صورت می باشند که یک somewhere داریم که هر نامی می تواند باشد و یک domain که domainها مشخص شده هستند.

از دید لایه internet، آدرس های اینترنتی بی معنی هستند پس با مشکل مواجه می شویم و مشکل این است که آدرس های اینترنتی فقط برای راحتی کاربر است و در شبکه دارای هیچ مفهومی نمی باشند.



شکل ۲۰۹: Country Domain

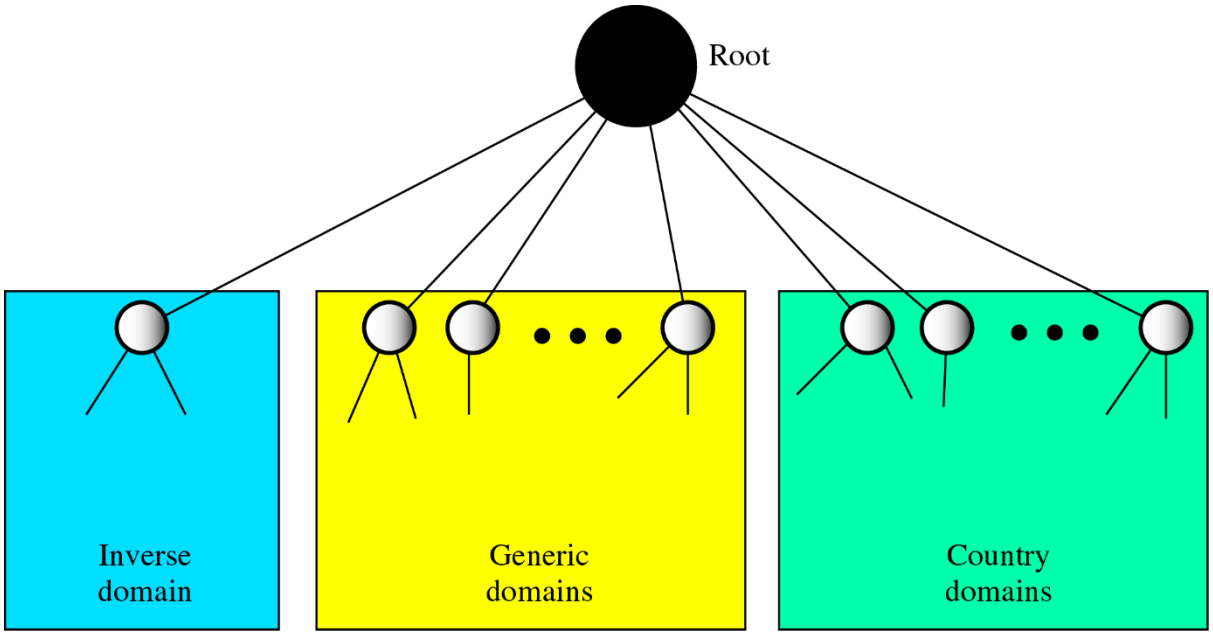
نامهای اینترنتی unic می باشند. هیچ دو نفری در اینترنت نمی توانند دارای یک نام یکسان باشند. بنابراین برای اینکه به هر شخص یک آدرس خاص داده شود، سازمانهایی وجود دارند که این وظیفه را بر عهده دارند که نام پیشنهادی کاربران را برای آنها ثبت می کنند.

بنابراین برای حل مشکل که مطرح شد، باید جدولی داشته باشیم که نامهای اینترنتی را در یک ستون و IP addressها را نیز در ستون دیگر داشته باشد. این جدول با افزایش تعداد nodeها روز به روز گسترش پیدا کرد و برای ارسال این جدول ترافیک خیلی زیادی در شبکه به وجود می آمد.

برای حل این مشکل مبحث domain name system ها یا سیستم های نام حوزه (dns) مطرح شد.

DNS:

دیدید که اگر یک سرویس دهنده وجود داشته باشد که نام اینترنتی را بگیرد و ip address بدهد، ترافیک خیلی بالا می رود. بنابراین ditributed data base ها را مطرح کردند که این بانک اطلاعات را به بخشهای مختلف می شکنند و هر مجموعه ای کافیست که dns محلی خاص خود را داشته باشد. وقتی که یک node بخواهد با node دیگر ارتباط برقرار کند، درخواست خود را به dns محلی می دهد که dns محلی ip را بر می گرداند و ip را به لایه transport ما می دهد و ارتباط از شریق لایه transport برقرار می شود. حال اگر چنین آدرسی در dns محلی وجود نداشت، این درخواست را به لایه بالاتر می فرستد و dns لایه بالاتر بررسی می کند و اگر آدرس باز هم پیدا نشد به لایه بالاتر می دهد تا بلاخره به root برسد.



شکل ۲۱۰: DNS در اینترنت

فرض کنید، surl.Eurecom.Fr می خواهد با شخصی بنام gaia.Cs.Umass.Edu ارتباط برقرار کند. یک ارتباط به local DNS برقرار می کند که نام این DNS محلی DNS.EURECOM.FR می باشد و آدرس IP طرف مقابل را از آن می پرسد. DNS محلی در داخل جدول خود نگاه می کند و می بیند که چنین چیزی ندارد در خواست را به DNS لایه بالاتر خود می فرستد که root name server نام دارد. چون آدرس DNS ریشه (ROOT) را می داند. ROOT NAME SERVER در جدول خود می گردد و آدرسی به نام GAIA.CS.UMASS.EDU را پیدا نمی کند که برای پیدا کردن آدرس از سمت راست به دنبال آن آدرس می گردد. اما می بیند که یک Name server با نام DNS.UMASS.EDU را دارد. درخواست خود را به dns.umass.edu می فرستد.

این DNS محلی در جدول خود می گردد و آدرس مورد نظر را پیدا می کند و IP address آن را دوباره به شخصی که درخواست آدرس IP کرده بود می فرستد.

با این روش dnsها می توانستند اولا ترافیک شبکه را کاهش دهند چون درخواست نام ابتدا از dns server محلی صورت می گیرد و اگر نبود به dns لایه بالاتر می رود. اگر بود ترافیک از آنجا بیرون نمی رود. ثانیا سرعت دسترسی با استفاده از dns بالا می رود.

محل قرار گیری dnsهای اصلی:

شکل ۲۱۱: محل قرارگیری dnsهای اصلی

در شبکه اینترنت ۱۳ dns root وجود دارد که در سراسر جهان پخش شده اند.

زمانیکه از اینترنت و از host مربوطه فضا می گیریم، نام ما در dns server آن ثبت می شود. که این dnsهای root مرتبا update می شوند.

Electronic mail:

از اولین پروتکل هایی که در شبکه اینترنت مطرح شد (و از اولین سرویسها) سرویس پست الکترونیکی می باشد.

شکل ۲۱۲: E-Mail

در پست الکترونیکی، ۳ عامل مهم وجود دارد:

User agents

Mail server

SMTP: Simple mail transfer protocol

User agent ها برنامه هایی هستند که روی client ها نشسته اند. در واقع user agent ها امکان ارسال و دریافت mail را برای ما فراهم می کنند (مانند outlook)

Mail serverها وظیفه نگهداری نامه های الکترونیکی را در صندوق پستی هر شخص بر عهده دارند.

پروتکل smtp، پروتکلی است که برای تبادل اطلاعات در سرویس پیت الکترونیکی استفاده می شود که پروتکلی ساده می باشد. قطعاً سرویس دهنده های پست الکترونیک گوش به زنگ پورت ۲۵ هستند و ارتباطی که برقرار می کنند از طریق پروتکل TCP می باشد.

پرتکل SMTP سه فاز برای تبادل اطلاعات دارد:

Hand shake

Transfer

Close

چون از پروتکل TCP استفاده می کند دارای این سه فاز می باشد. TCP هم دارای سه فاز ایجاد ارتباط، تبادل داده و قطع ارتباط می باشد. (۳ way hand shake)

SMTP یک پروتکل ascll می باشد. یعنی اطلاعات در این پروتکل به صورت ascll ارسال می شود.

ارسال نامه ها:

برای ارسال نامه، از پروتکل SMTP استفاده می شود. وقتی که یک Mail فرستاده می شود، ممکن است برای رسیدن به مقصد بین سرویس دهنده های گوناگون گردش کند. از سرویس دهنده فرستنده به سرویس دهنده طرف مقابل رفته و داخل صندوق پستی طرف مقابل (BOB) ذخیره می شود. BOB برای اینکه بتواند نامه را بخواند از یک پروتکل استفاده می کند. پروتکل که برای خواندن نامه ها استفاده می

شود، پروتکل POP ۳ OR IMAP می باشد یا پروتکل های HTTP که برای Web Mail ها اسـتفاده مـی شود (به عنوان مثال در yahoo)

سرویس دهنده های ارسال و دریافت نامه می توانند متفاوت باشند. یعنی برای ارسال نامه از یـک سـرویس دهنده و برای دریافت نامه از سرویس دیگری استفاده کنیم.

اگر بخواهیم از پروتکل POP ۳ استفاده کنیم باید از پورت ۱۱۰ استفاده کنیم چون فرستنده POP ۳ روی پورت ۱۱۰ قرار دارد.

File transfer protocol (ftp):

یکی دیگر از سرویس های رایج و اولیه در شبکه اینترنت ftp می باشد. جست و جو کـردن در ftp مشـکل است چون برای دیدن یک فایل باید آنرا download کرده و بعد آن را باز کنیم و بخوانیم.

پروتکلی که برای تبادل اطلاعات استفاده می شود پروتکل ftp است. ftp دارای دو پورت می باشـد: پـورت ۲۰ و ۲۱ که data از طریق پروت ۲۰ و اطلاعات کنترلی از طریق پورت ۲۱ ارسال می گردد و دارای یـک سری دستورات می باشد.

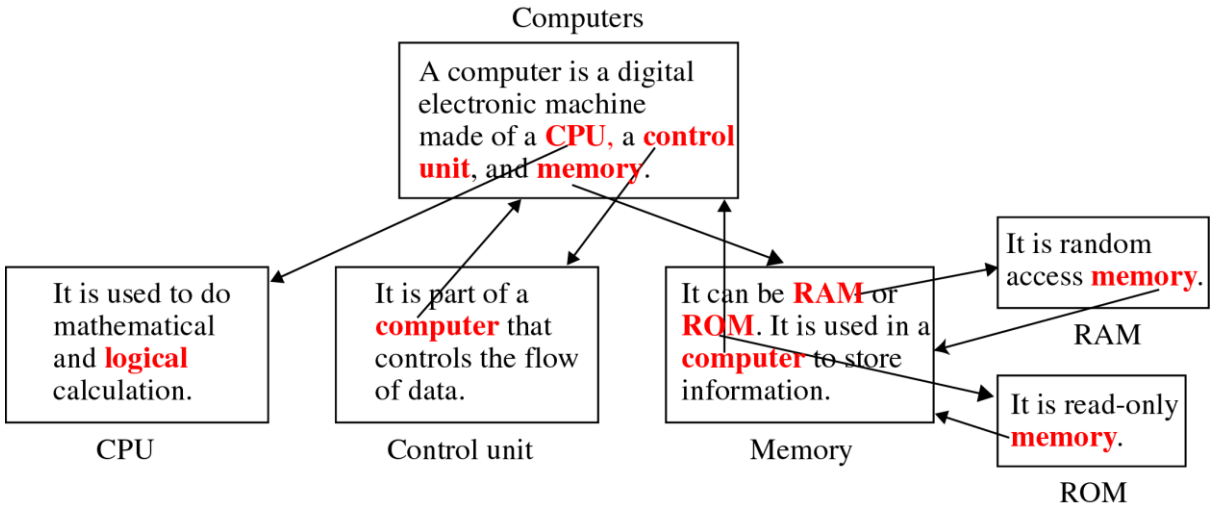
HTTP Protocol:

Web یکی از پرکارترین و رایج ترین سرویس های اینترنت می باشد. پروتکلی کـه در تبـادل اطلاعـات در web استفاده می شود، پروتکل http می باشد. (پروتکلی را برای تبادل اطلاعـات در لایـه application بین یک client و user استفاده می شود، پروتکل http می باشد.)

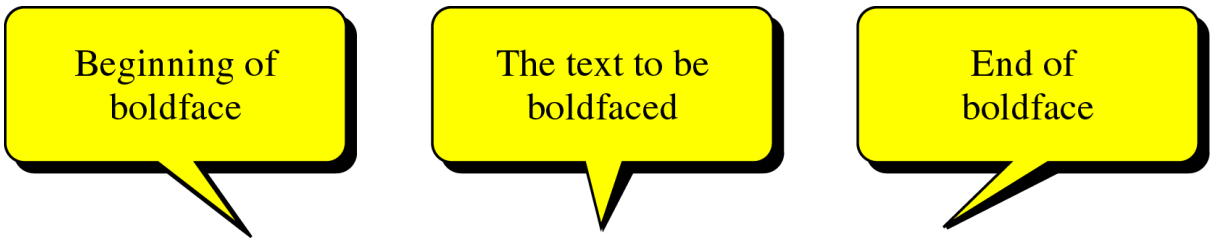
سرویس دهنده وب کار خاصی انجام نمی دهد و در شبکه جستجو می کند تا ببیند نامی را که ما درخواست کرده ایم در (url) وجود دارد یا نه.

HTML:

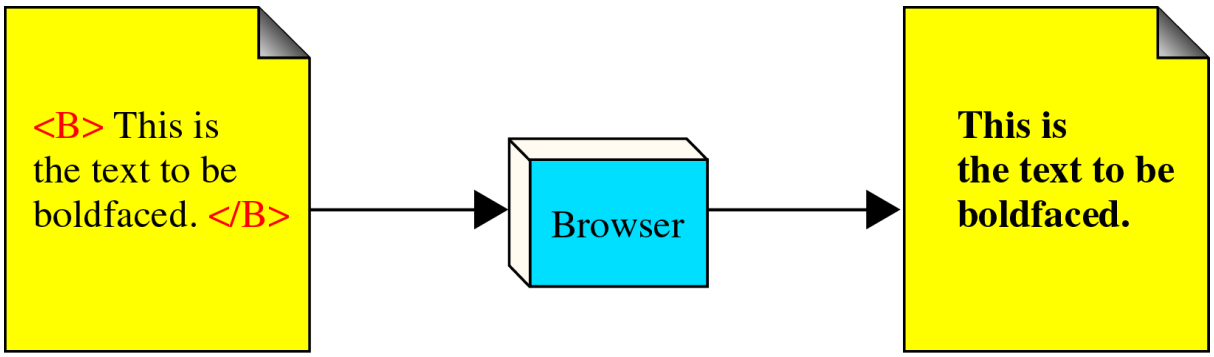
برای ساختن صفحات وب، از زبانی به نام HTML استفاده می شود. این زبان نشانه گذاری است هر علامتی در آن برای browser یک معنای خاص دارد. Html را می توان در یک editor معمـولی نوشـت و یـک browser این text file را می خواند و بر مبنای اطلاعات آن، صفحه آرایی و ... را انجام می دهد.



شکل ۲۱۳: HTML



شکل ۲۱۴: نمونه ای از کد HTML



شکل ۲۱۵: چگونگی نمایش نمونه ای از کد HTML

Security:

بحث امنیت در قالب ۳ لایه و تحت پروتکل های زیر بررسی می شود:

۱. PGP
۲. SSL و TSL
۳. IPSec

برای درک بحث امنیت، لازم است نیاز های امنیت تعریف شود امنیت شبکه به چهار نیاز اشاره می کند:

۱. جامعیت (integrity):

لازم می دارد که داده فقط توسط افراد مجاز قابل اصلاح باشد. اصلاحات شامل نوشتن، تغییر، تغییر وضعیت، پاک کردن و ایجاد موضوع است.

۲. قابلیت اعتبار:

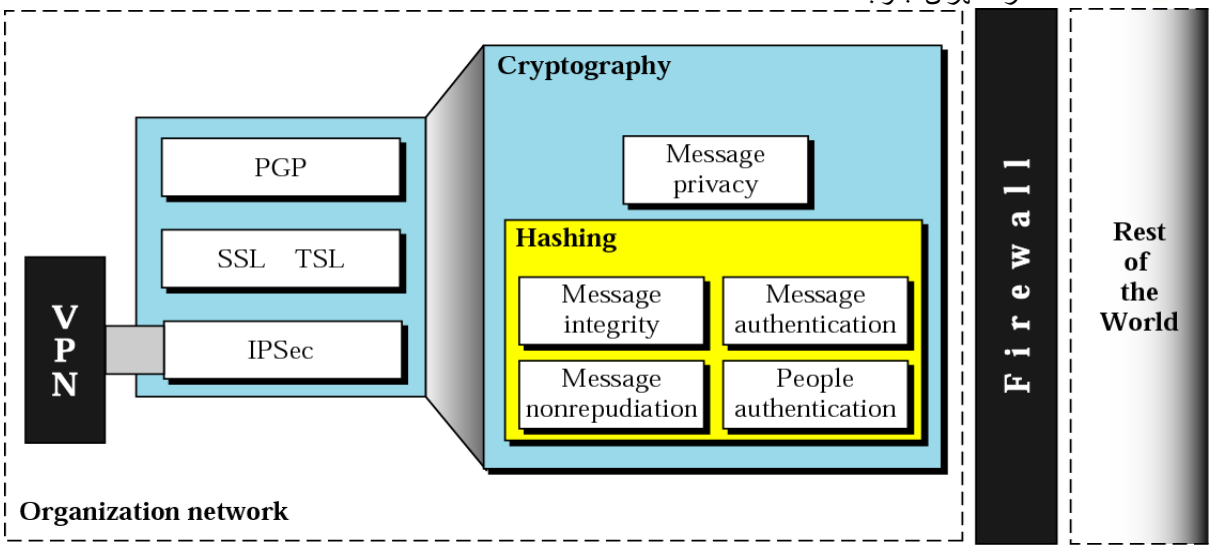
لازم می دارد که کامپیوتر میزبان یا سرویس دهنده قادر به بازبینی هویت کاربر باشد.

۳. محرمانه بودن (confidentiality):

نیاز دارد که داده ها فقط به وسیله افراد مجاز قابل دسترسی باشند. این نوع دستیابی شامل چاپ، نمایش و دیگر فرم های نمایان ساختن است.

۴. قابلیت دسترسی (availability):

داده فقط در اختیار افراد مجاز یاشد.

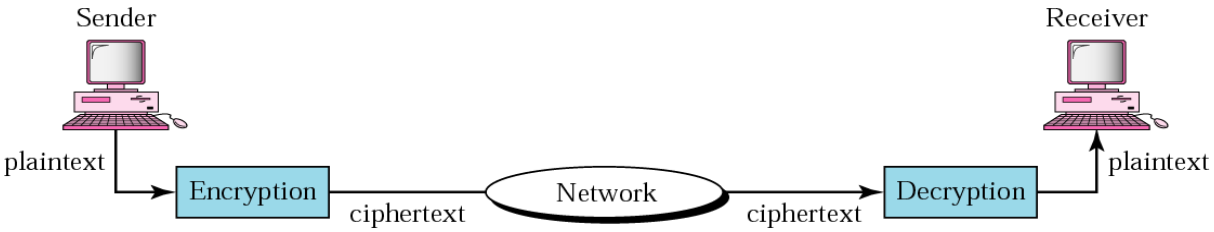


شکل ۲۱۶: امنیت

Cryptography (رمز شناسی):

واژه ای است که برای رمزنگاری استفاده می شود، به منظور محرمانه ماندن اطلاعات در حال انتقال بین مبدا و مقصد یا همان privacy رمزنگاری در قالب ۴ رده زیر می باشد:

۱. Message integrity: نظارت بر صحت و درستی اطلاعات
۲. Message authentication: احراز هویت برای پیغام های ارسالی.
۳. People authentication: تایید هویت کاربر مقابل در ارتباط قبل از آنکه اطلاعات در اختیار او قرار گیرد و یا اینکه اطلاعات ارسالی از طرف او دریافت شود.
۴. Message nonrepudiation: به هویت فرستنده باز می گردد، فرستنده اطلاعات نتواند بعد از ارسال پیام آنها را انکار کند.



شکل ۲۱۷: مولفه های رمز شناسی

اطلاعات توسط یک الگوریتم رمزنگاری با استفاده از یک کلید خاص که محرمانه است به صورت رمز درآمده و بدین ترتیب plaintext به ciphertext تبدیل می شود. در مقصد این اطلاعات رمز شده توسط یک الگوریتم رمز گشایی و توسط یک کلید خاص محرمانه بصورت اطلاعات خام در می آید.

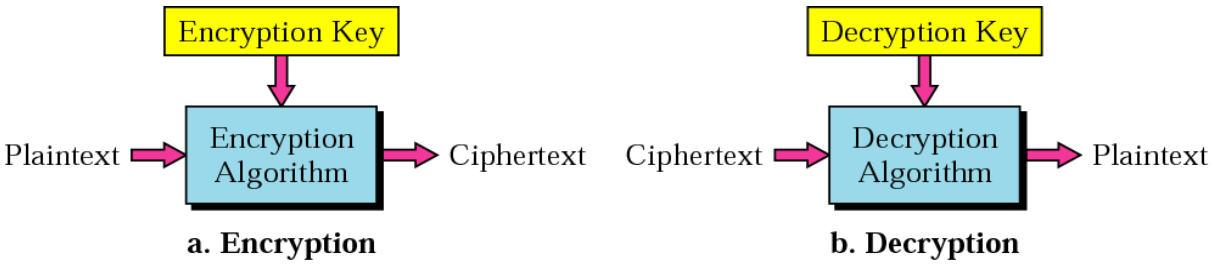
در اینجا کلید های رمز گشایی پنهان و محرمانه هستند و الگوریتم ها عمومی و غیر قابل پنهان کردن می باشند.

روش های محرمانه سازی:

۱. روش های متقارن یا symmetric
۲. روش هاس نا متقارن یا Asymmetric

محرمانه سازی با رمزگذاری متقارن:

رمزگذاری متقارن که به آن رمزگذاری متداول یا رمزگذاری تک کلیدی هم می گویند تنها رمزگذاری مورد استفاده قبل از معرفی رمزگذاری کلید عمومی در اواخر دهه ۱۹۷۰ بود.



شکل ۲۱۸: مولفه های رمز گذاری متقارن

طرح رمزگذاری متقارن دارای ۴ جز می باشد:

- متن اصلی: این همان پیام یا داده اصلی است که به الگوریتم به عنوان ورودی داده می شود.
- الگوریتم رمزگذاری: الگوریتم رمزگذاری انواع جایگزینی ها و تبدیل ها را روی متن اصلی انجام می دهد.
- کلید محرمانه: یک ورودی به الگوریتم رمزگذاری است. تبدیل و جایگزینی دقیق انجام شده توسط الگوریتم به کلید بستگی دارد.
- متن رمزگذاری شده: پیامی به هم ریخته است که بعنوان خروجی تولید می شود. این کار به متن اصلی و کلید محرمانه بستگی دارد. برای یک پیام، دو کلید متفاوت دو متن رمز گذاری شده متفاوت ایجاد می کند.

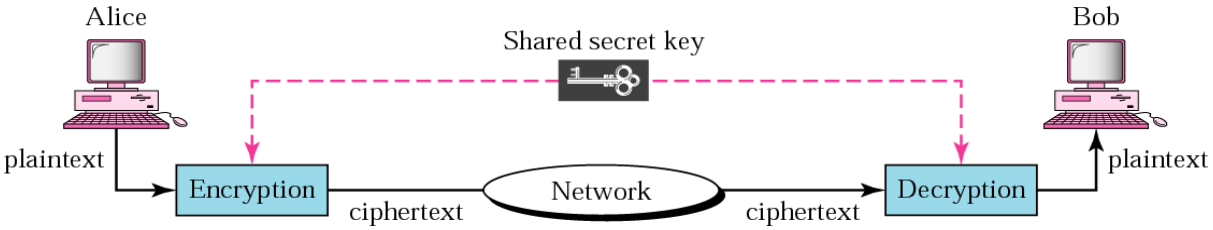
- الگوریتم رمزگشایی: در واقع این همان الگوریتم رمزگذاری است که به صورت معکوس انجام می شود ایم الگوریتم از متن رمزگذاری شده و کلید محرمانه استفاده کرده و متن اصلی را تولید می نماید.

در رمزگذاری متقارن یک کلید مشترک بین فرستنده و گیرنده اطلاعات وجود دارد که هر دو از آن استفاده می کنند و این کلید برای هم دو عمل رمز نگاری و رمز گشایی استفاده می شود.

محرمانه سازی با رمز گذاری متقارن چون سربار زیادی ایجاد نمی کند برای حجم های بالا مناسب می باشد

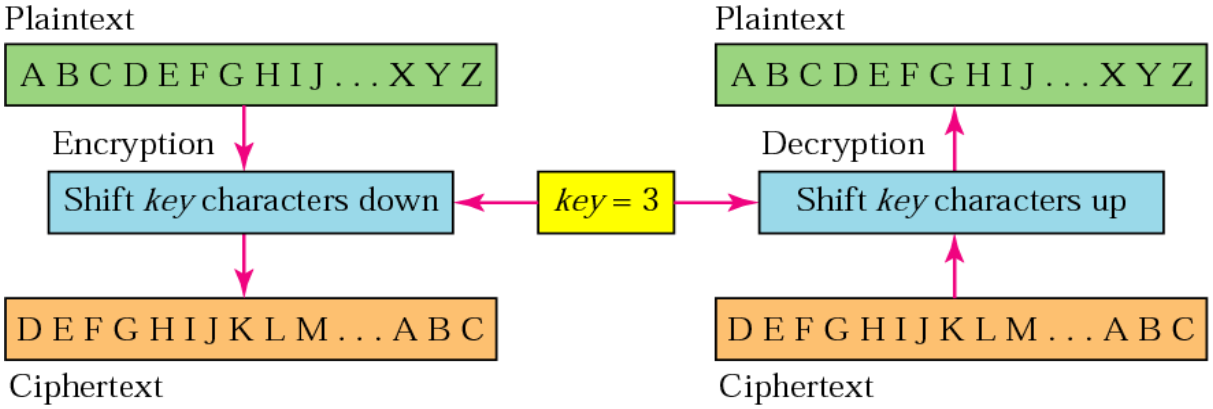
برای استفاده امن از رمزگذاری متقارن دو نیاز وجود دارد:

۱. یک الگوریتم رمزگذاری قوی حداقل می خواهیم الگوریتم چنان باشد که اگر مهاجمی قادر به دستیابی به یک یا چند متن رمزگذاری شده شد نتواند متن رمزگذاری شده را از رمز خارج کند و یا خود رمزگذاری نموده و یا کلید را شناسایی کند.
۲. فرستنده یا گیرنده باید کپی هایی از کلید محرمانه را به شکل امن در اختیار داشته باشند و آن را به صورتی مطمئن حفظ کنند.



شکل ۲۱۹: محرمانه سازی با رمزگذاری متقارن

شکل یک روش رمزگذاری متقارن با استفاده از الگوریتم caesar را نشان می دهد. این الگوریتم یک الگوریتم عمومی است ولی کلید های مختلفی می تواند داشته باشد. این الگوریتم با توجه به مقدار کلید حروف را جابجا می کن. مثلاً در این شکل بواسطه کلید که مقدار آن ۳ است حروف به اندازه ۳ تا جابجا می شوند. این کلید یک کلید محرمانه است و از مقدار آن فقط فرستنده و گیرنده آگاه هستند چون الگوریتم caesar یک الگوریتم عمومی می باشد اگر همه افراد از مقدار کلید با خبر باشند به راحتی می توانند اطلاعات را رمز گشایی کنند و از محتوای آن مطلع شوند و یا آن را تغییر دهند.

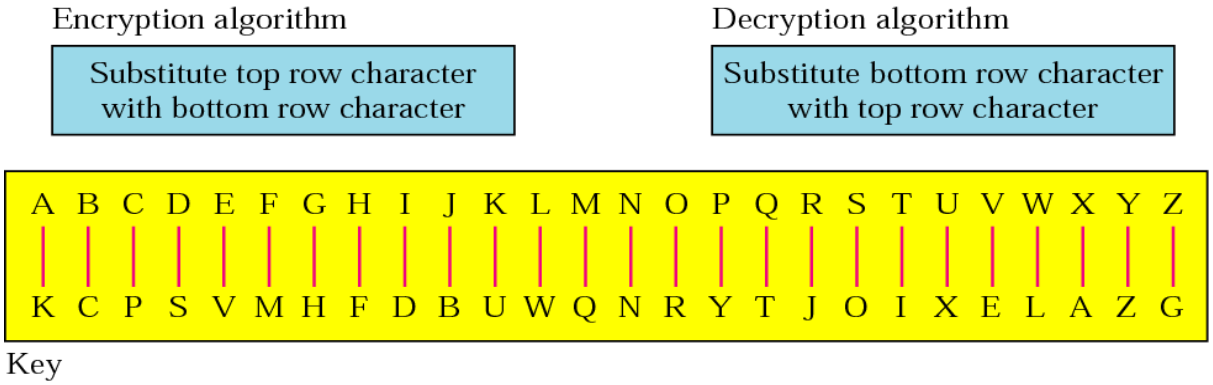


شکل ۲۲۰: الگوریتم caesar

شکل روشی از رمزگذاری که به آن monoalphabetic substitution می گویند را نشان می دهد که در آن یک کد به کد دیگری تبدیل شده است. دی اینجا یک تناظر یک به یک بین اطلاعات رمز شده و اطلاعات خام وجود دارد.

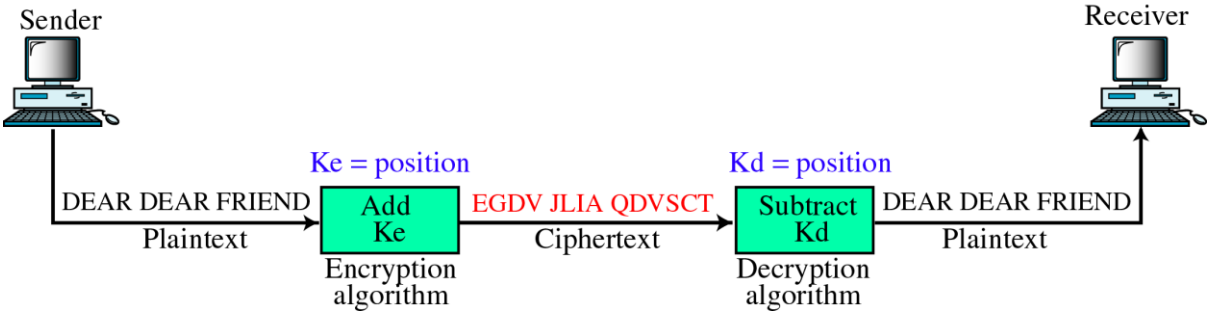
در این روش کلید به صورت یک جدول است که در فرستنده با توجه به آن حروف نامه ارسالی را به حروف دیگر تبدیل می کنند و در گیرنده با توجه به آن عمل عکس را صورت می گیرد.

در اینجا نیز فقط فرستنده و گیرنده از کلید که یک جدول می باشد اطلاعا دارند.



شکل ۲۲۱: الگوریتم monoalphabetic substitution

شکل روش polyalphabetic substitution را نشان می دهد با توجه به این الگوریتم در فرستنده مقدار مکان هر حرف به کد اسکی همان حرف اضافه می شود و متن رمز شده ارسال می گردد و در گیرنده عمل عکس انجام می گیرد تا متن اصلی بدست آید.



شکل ۲۲۲: polyalphabetic substitution

محرمانه سازی با رمزگذاری نامتقارن:

در این روش دیگر کلید بین مبدا و مقصد یکی نیست. بعبارتی مبدا با یک کلید رمزنگاری می کند و مقصد با استفاده از کلید دیگری رمزگشایی می کند.

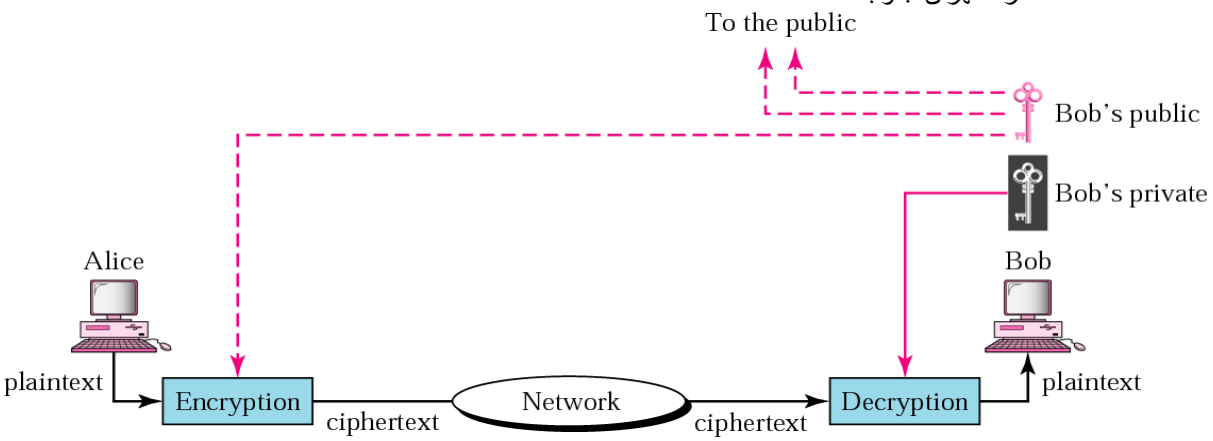
در این روش دو کلید وجود دارد کلید عمومی یا public key و کلید خصوصی یا private key کلید خصوصی هر کس محرمانه است و فقط خودش از آن خبر دارد و برای هیچ کس نمایش داده نمی شود ولی از کلید عمومی همه افراد شبکه خبر دارند، بعبارتی هر شخص کلید عمومی خود را در اختیار همه قرار می دهد.

کلید های عمومی و خصوصی هر شخص با یکدیگر دارای رابطه ریاضی می باشند.

هر شخص یک زوج کلید دارد (کلید عمومی و کلید خصوصی هر شخص کلید عمومی خود را برای فرستادن اطلاعات می تواند در اختیار دیگران قرار دهد. اگر کسی بخواهد برای نفر دوم اطلاعاتی را بفرستد باید کلید عمومی آن را داشته باشد. بعبارت دیگر کلید عمومی و خصوصی هر کس با یکدیگر ارتباط دارد پس کسی می تواند این اطلاعات را رمزگشایی کند که کلید خصوصی مربوط به آن کلید عمومی را داشته باشد.

روش کلید عمومی:

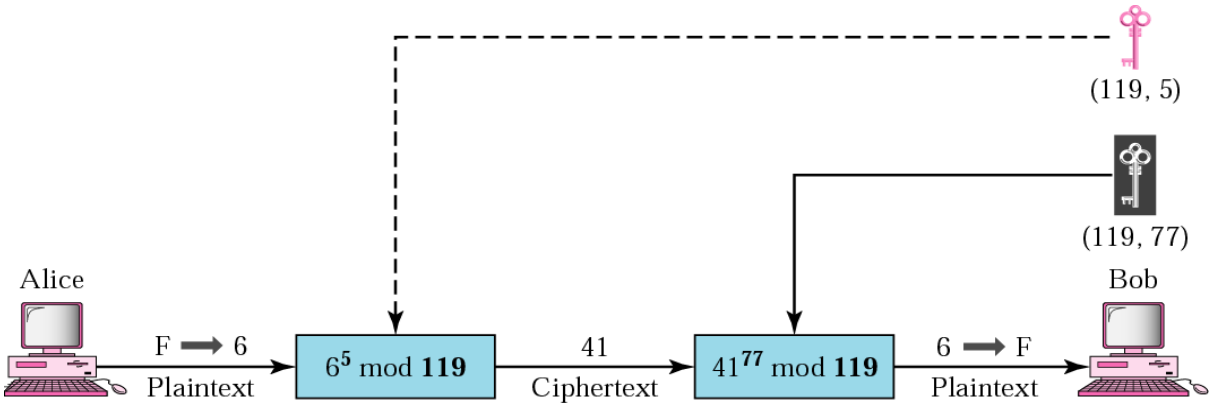
در شکل آلیس و باب می خواهند ارتباط برقرار نمایند. هر کس برای باب اطلاعات بفرستد باید کلید عمومی باب را داشته باشد. آلیس با استفاده از کلید عمومی باب اطلاعات خود را رمزنگاری می کند و برای باب می فرستد. باید به این نکته توجه کرد که فقط کسی می تواند اطلاعات رمزگذاری شده با کلید عمومی را decode کند که کلید خصوصی خود اطلاعات را رمزگشایی می کند.



شکل ۲۲۳: رمزگذاری کلید عمومی

الگوریتم رمزگذاری کلید عمومی RSA:

آلیس می خواهد کاراکتر F را برای باب بفرستد، یک کد مانند ۶ به آن اختصاص می دهد آلیس تیم کد را در رابطه رمزگذاری (با توجه به کلید عمومی باب ۱۱۹,۵) قرار می دهد و عدد ۴۱ را بدست می آورد و باب با توجه به کلید خصوصی خود ۱۱۹,۷۷ عدد ۴۱ را رمز گشایی می کند.



شکل ۲۲۴: الگوریتم رمزگذاری کلید عمومی RSA

از آنجا که از کلید خصوصی باب کسی خبر دار نیست کسی نمی تواند آن را رمز گشایی کن و هر چقدر تعداد ارقام این عدد بالاتر رود اطمینان بالاتر می رود.

کلید عمومی و کلید خصوصی در یک عدد مشترک هستند ولی یک عدد متفاوت دارند که در ایم مثال ۷۵ و ۵ است. دیگران ۱۱۹,۵ را می دانند چون کلید عمومی است و براحتی می توانند ۱۱۹,۷۷ که کلید خصوصی باب است را حدس بزنند ولی در واقع این عدد ۷۷ صد رقمی است که حدس آن آسان نیست.

منطق این روش ریاضی است و پردازش آن در مبدا و گیرنده بسیار طولانی می شود پس برای حجم های کوتاه استفاده می شود.

چگونگی انتخاب کلید عمومی و خصوصی در الگوریتم RSA:

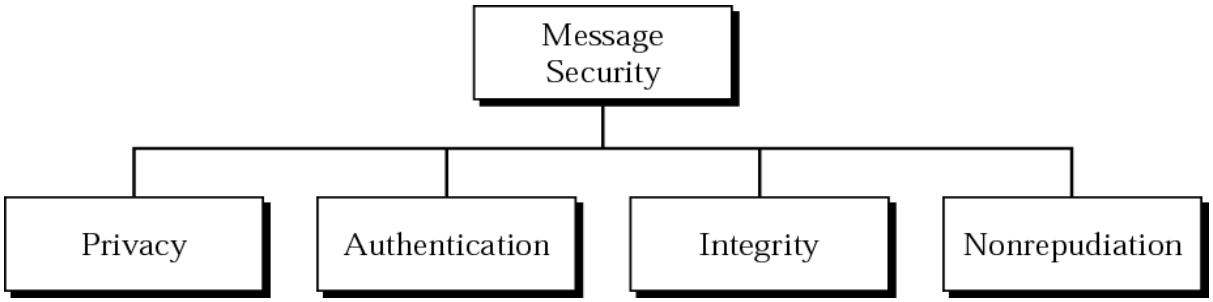
۱. دو عدد اول p و q را انتخاب می کنیم.
۲. مقدار n=p*q را بدست می آوریم و همچنین مقدار z=(p-۱)(q-۱)
۳. مقدار e را برمبنای e<n طوری بدست می آوریم که نسبت به z عددی اول باشد.
۴. D را بر مبنای فرمول ed mode z=۱ بدست می آوریم
۵. کلید عمومی برابر (n,e) و کلید خصوصی برابر (n,d) است.

می خواهیم یک pattern به نام m را کد گذاری کنیم. بعبارتی m در حکم اطلاعات می باشد.

۱. از الگوریتم فوق (n,e) و (n,d) را بعنوان کلید عمومی و کلید خصوصی محاسبه می کنیم.
۲. M را به توان e می رسانیم.
۳. باقی مانده تقسیم عدد بدست آمده از مرحله ۲ بر n اطلاعات رمزنگاری شده می باشد.

در گیرنده اطلاعات رمز شده را به توان d می رسانند و باقی مانده تقسیم آن بر n اطلاعات خام (m) می باشد.

Message Security: (امنیت پیام)



امنیت message در ۴ قالب زیر (که قبلاً در مورد آنها صحبت شده است) بررسی می شود:

۱. محرمانه بودن (privacy)
۲. اعتبار سنجی (authentication)
۳. سلامت اطلاعات (integrity)
۴. عدم انکار (nonrepudiation)

امنیت پیام در رمزگذاری به روش متقارن:

محرمانگی با استفاده از روش متقارن تامین می شود. در اینجا بحث چگونگی رد و بدل کردن کلید بین فرستنده و گیرنده وجود دارد، همانطور که قبلاً صحبت شد در محرمانه سازی به روش متقارن الگوریتم

عمومی و کلید محرمانه است، مساله اصلی در اینجا این است که چگونگی کلید محرمانه بین فرستنده و گیرنده رد و بدل شود و هیچ کس دیگری آنرا کشف نکند.

امنیت پیام در رمزگذاری به کلید عمومی:

با وجود اینکه این الگوریتم زمان و پیچیده است ولی مشکل محرمانه بودن اطلاعات را حل می کند.

Digital Signature (امضای دیجیتال):

بطور کلی می توان گفت امضا تعیین کننده صحت شخص است و موجب اعتبار و احراز هویت می گردد.

براساس امضا می بایست:

۱. گیرنده بتواند هویت شخص فرستنده پیام را بررسی کند.
۲. فرستنده بعداً نتواند پیام ارسالی خود را انکار کند.
۳. گیرنده نیز نتواند برای خود پیام های جعلی بسازد و ارسال آنها را به دیگران نسبت دهد.

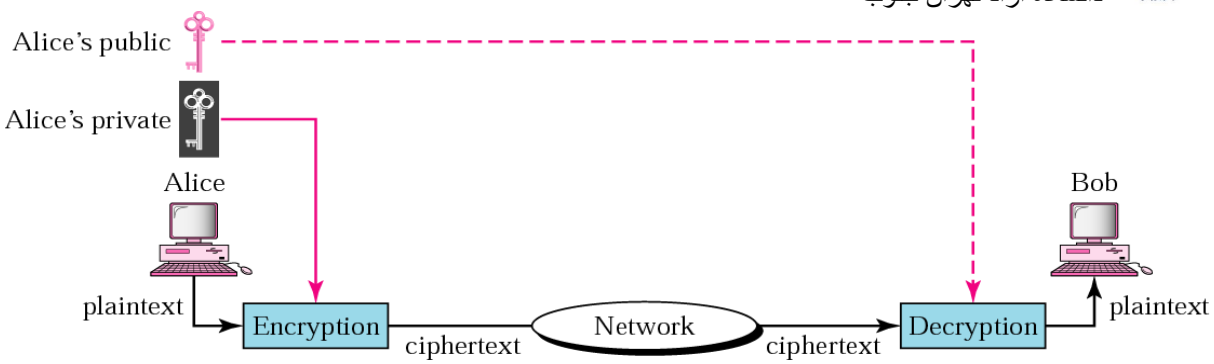
امضا دیجیتال در دو قالب بررسی می شود:

۱. کل سند را امضا کنیم
۲. بخشی از سند را امظا کنیم

Signing the whole Document (امضای کل سند)

فرض کنید که آلیس می خواهد پیامی به باب بفرستد و علی رغم اینکه پیام مهم نیست، می خواهد که پیام محرمانه بماند و می خواهد باب مطمئن شود که پیام از طرف اوست.

در این حالت آلیس از کلید اختصاصی خود برای رمز کردن پیام استفاده می کند. وقتی باب اطلاعات را دریافت می کند و در می یابد که می تواند با کلید عمومی آلیس آنرا رمزگشایی کند، پس پیام باید بوسیله آلیس رمزگذاری شده باشد. هیچ کس کلید اختصاصی آلیس ا ندارد و هیچ کس نمی تواند متنی رمزدار را ایجاد کند که بتواند با کلید عمومی آلیس آن را رمز گشایی کرد. بنابراین کل پیام رمزدار نقش یک امضای دیجیتال را دارد. علاوه بر آن، امکان ندارد پیام را بدون دستیابی به کلید اختصاصی آلیس عوض کند، بنابراین از نظر منبع و صحت داده معتبر است.

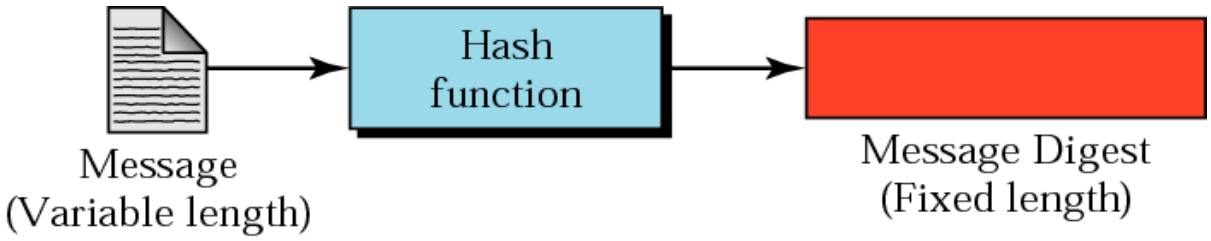


شکل ۲۲۵: امضای دیجیتال برا کل سند

Signing the Digest (امضای بخشی از سند)

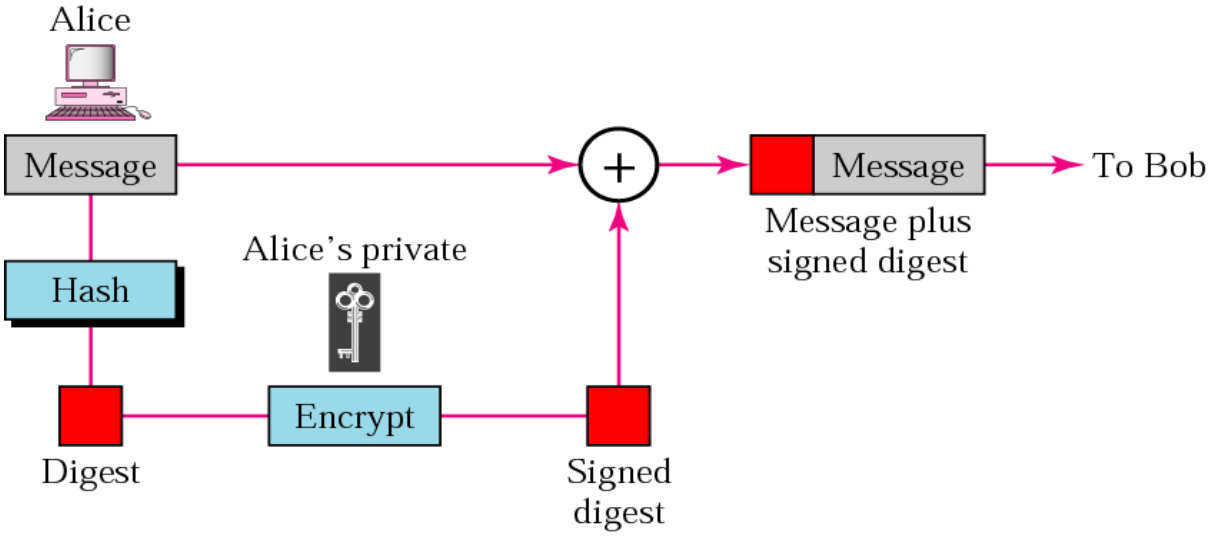
در طرح قبل، کل پیام رمزی می شد و هر چند مولف و پیام را معتبر می نمود، ولی حافظه زیادی را لازم دارد. هر سند باید به صورت متن اصلی نگهداری شود تا برای اهداف عملی مورد استفاده قرار گیرد. یک کپی هم باید به صورت رمزی باشد تا اصل و محتویات نتوانند در هنگام لزوم مقایسه و بازبینی شوند. راه مفیدتری در رسیدن به همان نتایج، رمزی کردن بلوک کوچکی از بیت هاست که تابعی از اسناد است. چنین بلوکی را بلوک اعتبار سنج می نامند و باید این خاصیت را داشته باشد که تغییر سند بدون تغییر اعتبارسنج، امکان نداشته باشد.

الگوریتم Hash روی قسمتی از اطلاعات انجام می شود و آن را کد می کند که طول آن ثابت است و تغییر نمی کند. چون کد کردن روی قسمتی از متن صورت می گیرد بنابراین زمان کمتری می برد.



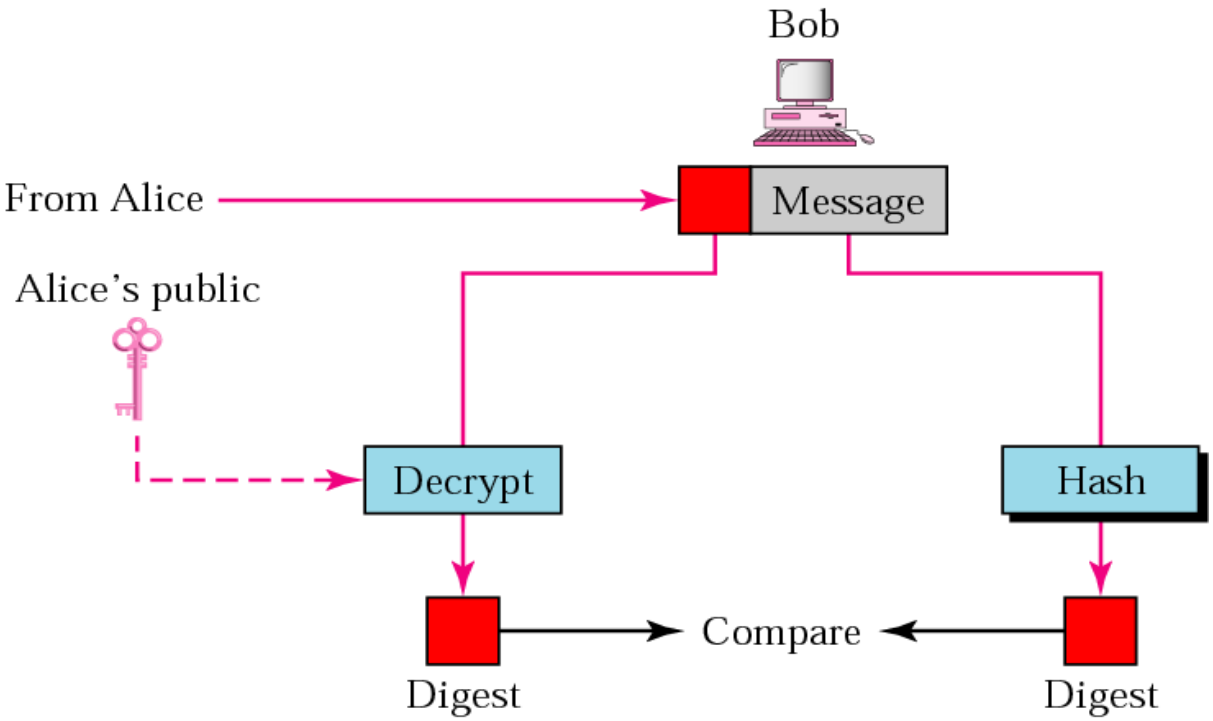
شکل ۲۲۶: امضای بخشی از سند.

در مثال زیر الگوریتم Hash روی قسمتی از پیام آلیس اعمال می شود بعد با کلید خصوصی کد می شود و سپس به همراه متن اصلی پیام برای گیرنده ارسال می شود.



شکل ۲۲۷: امضای بخشی از سند فرستنده

باب که در این مثال گیرنده است پیام را دریافت می کند. قسمت Hash شده را با کلید عمومی آلیس رمزگشایی می کند. متن عادی پیغام را با الگوریتم hash کد می کند. سپس حاصل این دو عمل را با هم مقایسه می کند.



شکل ۲۲۸: امضای بخشی از سند گیرنده

لازم است تاکید شود که امضای دیجیتالی محرمانه بودن را فراهم نمی کند یعنی پیامی که ارسال می شود به لحاظ تغییر امن است ولی در برابر استراق سمع ایمن نیست. این برای حالتی که یک امضا بر اساس

بخشی از پیام است نیز واضح است، زیرا بقیه پیام بصورت صریح ارسال می گردد حتی به هنگامی که رمزگذاری کامل است، حفاظتی برای محرمانه ماندن نیست زیرا مشاهده گری می تواند پیام را با کلید عمومی ارسال کننده رمزگشایی نماید.

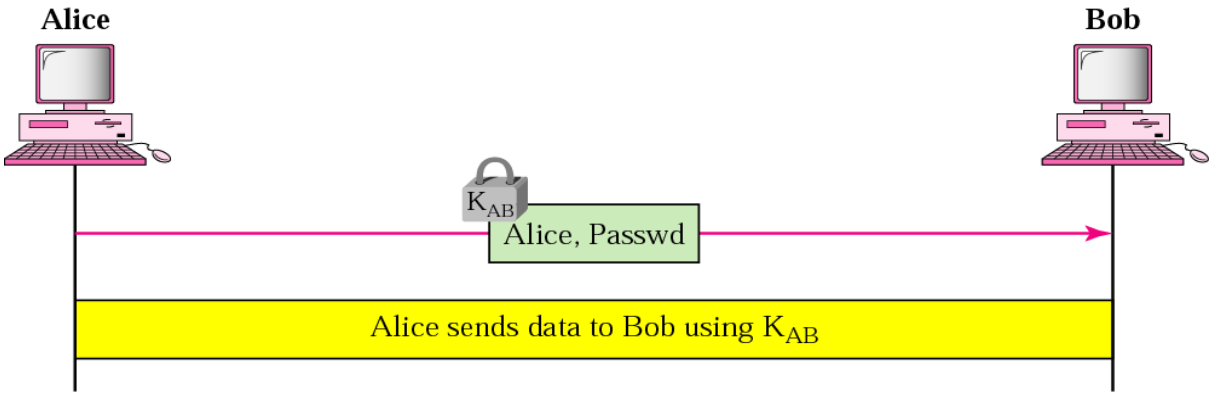
User Authentication (اعتبار سنجی):

برای بررسی هویت کاربر می توانیم از دو روش زیر استفاده کنیم:

۱. With symmetric key (با استفاده از کلید متقار)
۲. With public key (با استفاده از کلید عمومی)

اعتبار سنجی با استفاده از کلید متقارن

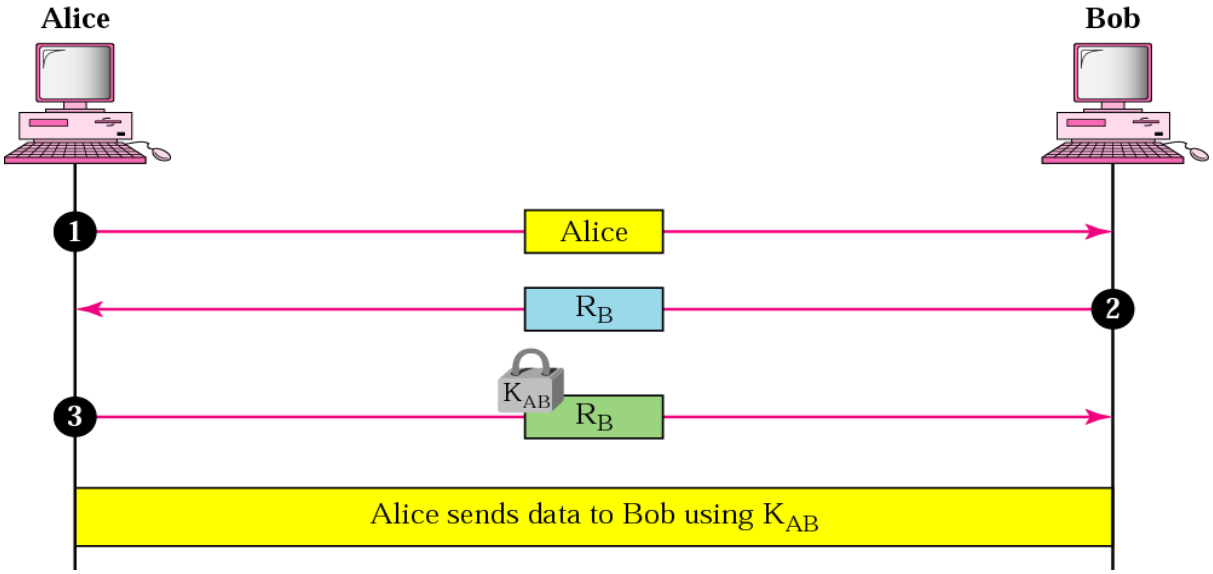
آلیس و باب قبل از صحبت باید تعیین هویت شوند و شناخته شوند. در روش متقارن کلیدی وجود داشت که بین آلیس و باب مشترک بود. آلیس اسم و password خود را با این کلید مشترک کد می کند و می فرستد. چون فقط باب این کلید مشترک را دارد آن را دیکد می کند و دیگران نمی توانند به آن دسترسی داشته باشند.



شکل ۲۲۹: اعتبار سنجی فقط با استفاده از کلید متقارن

مشکل این روش این است که هر کسی روی خط بیاید می تواند این اطلاعات کد شده را بفرستد و به جای آلیس تشخیص داده می شود.

برای حل این مشکل آلیس برای برقراری ارتباط نام خود را برای باب می فرستد. باب یک عدد تصادفی برای آلیس می فرستد. آلیس این عدد را (RB) با کلید مشترک کد می کند و می فرستد. باب آن را با اطلاعات خود که فرستاده مقایسه می کند اگر درست بود هویت آلیس اثبات شده است.



شکل ۲۳۰: اعتبار سنجی با استفاده از کلید متقارن

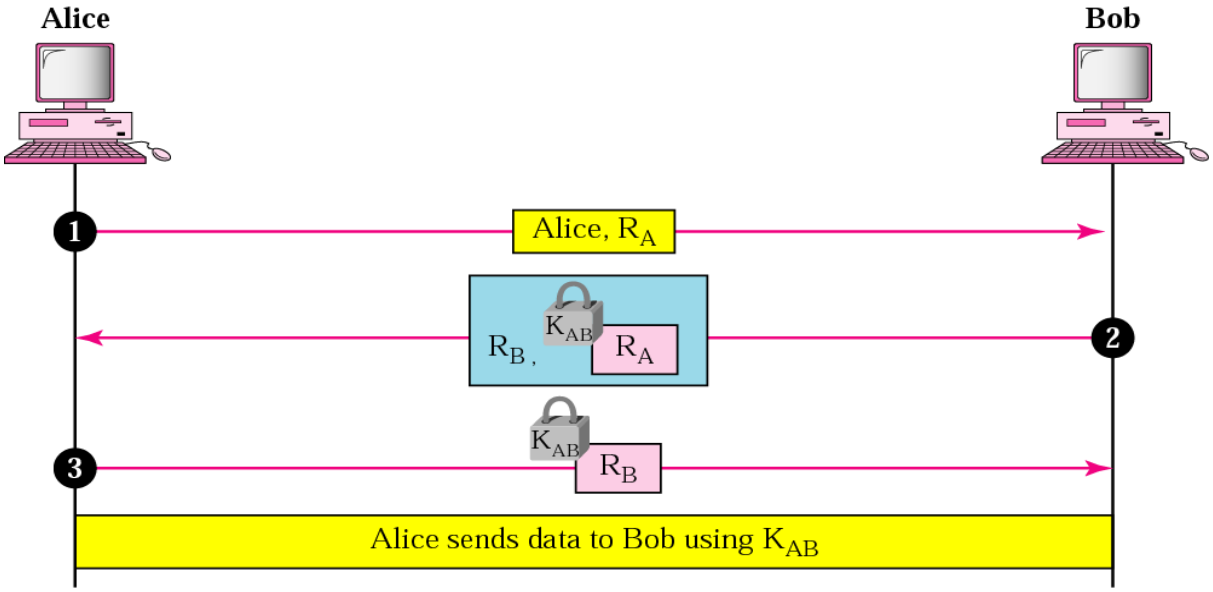
مشکل روش اعتبار سنجی با استفاده از کلید متقارن:

۱. تعداد کلید ها زیاد می باشد (N^2)
۲. مدیریت کلید ها
۳. قبل از رد و بدل کردن اطلاعات می بایستی کلید رد و بدل شود.

در اعتبار سنجی با استفاده از کلید عمومی پیچیدگی به n می رسد، در این روش محرمانگی و سلامت کاملاً تایید شده است.

اعتبار سنجی دوطرفه:

برای اینکه هویت باب هم برای آلیس روشن شود، آلیس نام خود را با یک عدد تصادفی برای باب می فرستد سپس باب (RA) را با کلید مشترک کد می کند، اگر عدد فرستاده شده خودش بود هویت باب برایش ثابت می شود سپس RB برای آلیس می فرستد. آلیس RA کد شده را با کلید مشترکدیکد میکند، اگر عدد فرستاده شده خودش بود هویت باب برایش ثابت می شود سپس RB را با کلید مشترک کد می کند و برای باب می فرستد. باب نیز آن را با کلید مشترک دیکد کرده و با اطلاعاتی که فرستاده بود چک می کند اگر درست بود هویت آلیس نیز درست است.



شکل ۲۳۱: اعتبار سنجی دوطرفه

RA یا RB به صورت اطلاعات خام هستند که هر کس می تواند آنها را دریافت کن مهم کد شده آنها است که کسی غیر از شخصی که کلید مشترک را دارد نمی تواند آن را باز کند.

Key Management (مدیریت کلید):

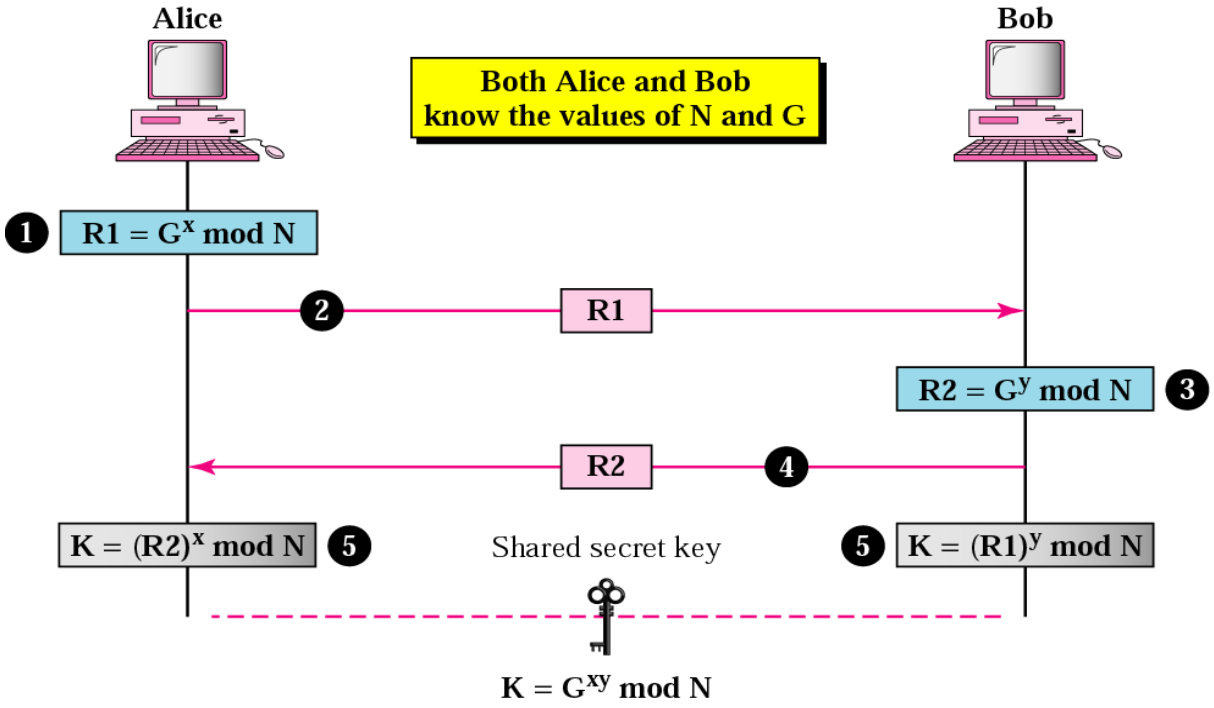
برای مدیریت کلید می توانیم از دو روش زیر استفاده کنیم:

۱. Symmetric key Distribution (کلید متقارن تو زیع شده)
۲. Public key Certification (کلید عمومی تصدیق شده)

روش متقارن با استفاده از الگوریتم Diffie hellmen

یک کلید متقارن بین دو قسمت فقط برای یکبار استفاده، مفید است. آن کلید برای یک session ایجاد می شود و وقتی آن session تمام شد از بین می رود.

می خواهیم کلید را از روی خط نفرستیم و آن را بین دو کاربر ایجاد کنیم.



شکل ۲۳۲: الگوریتم Diffie Hellmen

N و G اعداد اول هستند. آلیس و باب از این دو عدد خبر دارند N خیلی بزرگ است. آلیس (فرستنده) یک عدد تصادفی مثل X را انتخاب می کند که کسی از آن خبر ندارد و $R_1 = G^x \bmod N$ محاسبه کرده و آن را به سمت باب می فرستد باب y را که یک عدد تصادفی است و عدد بزرگی است انتخاب می کند و $R_2 =$ $G^y \bmod N$ را برای آلیس می فرستد.

آلیس $K = R_2^x \bmod N$ را حیاب می کند. باب هم $K = R_1^y \bmod N$ را حساب می کند این k یک عـدد مشترک می شود که به عنوان کلید توافقی استفاده می شود و کسی از آن خبر ندارد. در ارتباط بعدی x و y دیگری در نظر گرفته می شود.

این الگوریتم مشکل روش استفاده از کلید متقارن را حل کرده و به عبارتی کلید بصورت بلادرنگ رد و بـدل شد و به جز گیرنده و فرستنده کسی از آن خبر ندارد.

مثال از الگوریتم فوق که در آن کلید ۱۸ بدست می آید

Assume G = ۷ and N = ۲۳. The steps are as follows:

۱. Alice chooses x = ۳ and calculates $R_1 = ۷^۳ \bmod ۲۳ = ۲۱$
۲. Alice sends the number ۲۱ to bob
۳. Bob chooses y = ۶ and calculates $R_2 = ۷^۶ \bmod ۲۳ = ۴$
۴. Bob sends the number ۴ to alice

۵. Alice calculates the symmetric key K = ۴^{۳} mod ۲۳ = ۱۸
۶. Bob calculates the symmetric key K = ۲۱^{۶} mode ۲۳ = ۱۸

The value of K is the same for both Alice and bob

G^{xy} mode N = ۷^{۱۸} mod ۲۳ = ۱۸

پروتکل های امنیتی در اینترنت:

این بخش مربوط به امنیت لایه های اینترنت است:

۱. لایه Transport (در واقع بین لایه Transport و Application)
۲. لایه internet (معادل لایه network در OSI)
۳. لایه Aplication

IP level security (IPSec):

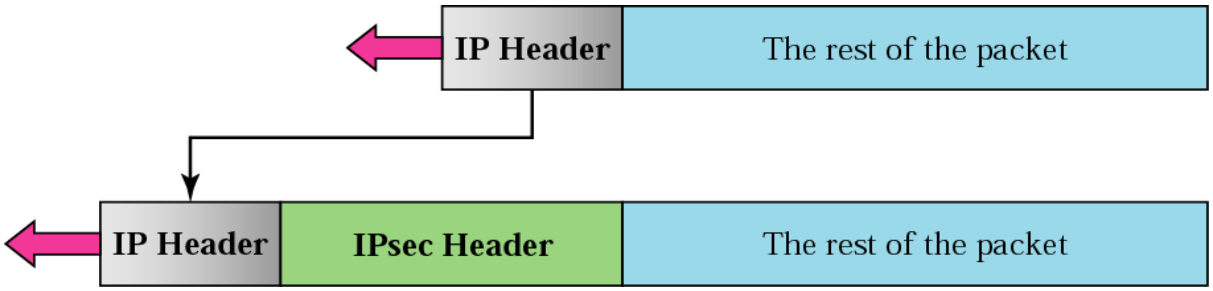
پیاده سازی آن در لایه اینترنتو به صورت سخت افزاری است. گذاشتن یک پروتکل به صورت سخت افـزاری درون Routerها بسیار مشکل است.

روی دو مد Transport و Tunnel کار می کند.

در مد Transport وقتی پروتکل IPSec را اضافه می کنیم کسی متوجه حمل آن نمی شود. برای استفاده از این پروتکل باید طرفهای مقابل نیز این پروتکل را support کنند.

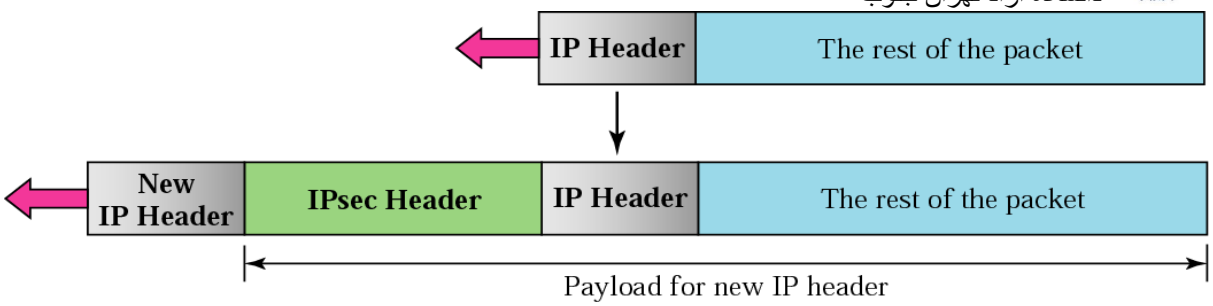
در پکت حاوی پروتکل IP بین بخش header و بخش data یک IPSec Header اضافه می شود.

پروتکـل IP یـک پروتکـل Connection less اسـت در صـورتی کـه پروتکـل IPSec یـک پروتکـل Connection oriented می باشد.



شکل ۲۳۳: پروتکل IPSec در مد Transport

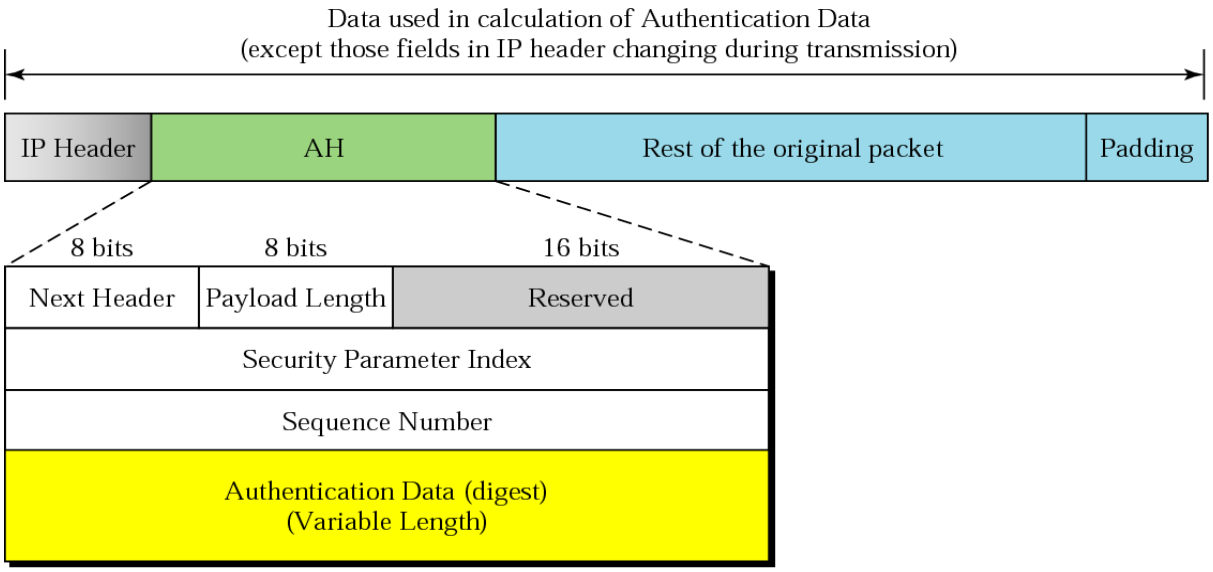
در مد Tunnel بسته بندی جدیدی صورت می گیرد و به صورت بسته بندی جدیدی ارسال می شود. یـک header جدید به آن اضافه می شود.



شکل ۲۳۴: پروتکل IPSec در مد Tunnel

پروتکل Authentication Header (AH):

در مد Transport کار می کند.



شکل ۲۳۵: پروتکل AH

Header تغییر نکرده و با همان فیلد های قبلی خودش است.

Next Header: مقدار فیلد پروتکل در IP Header، وارد این فیلد می شود و مقدار پروتکل ۵۱ می شود که مربوط به پروتکل AH است.

Pay load length: طول AH را مشخص می کند که مضربی از ۴ است.

Reserved: استفاده نمی شود

SPI: نقش عدد Logic در روش virtual circuit را دارد برای همه packetها یکسان است.

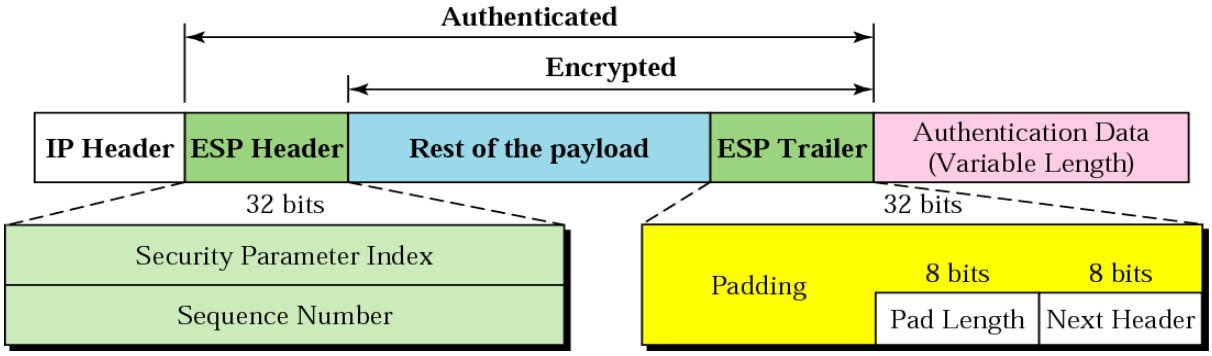
Sequence number: اگر حجم اطلاعات زیاد باشد، شماره packet ها را مشخص می کند.

Authentication data: نتیجه اجرای hash روی تمام datagram است یعنی روی تمام فیلدها الگوریتم hash اجرا می شود. روی فیلد های TTL و CHECKSUM انجام نمی شود. اگر تغییر کند، دیگر digest همان قبلی نیست. پس تشخیص خطا می دهد. پس سلامت اطلاعات تضمین می شود.

پروتکل AH اعتبار سنجی و صحت اطلاعات را فراهم می کند ولی privacy را فراهم نمی کند.

ESP:

در مد tunnel کار می کند.



شکل ۲۳۶: پروتکل ESP

یکسری ESP tailer به دیتا اضافه می کند و آن را کد می کند و یک ESP header هم قبل از آن قرار می دهد و Header اصلی دیتا اعتبار سنجی، سلامت و محرمانه بودن را تضمین می کند چون اطلاعات رمز می شود.

Next header: کد فیلد پروتکل ۵۰ ESP است که محتوای آن در این فیلد قرار می گیرد.

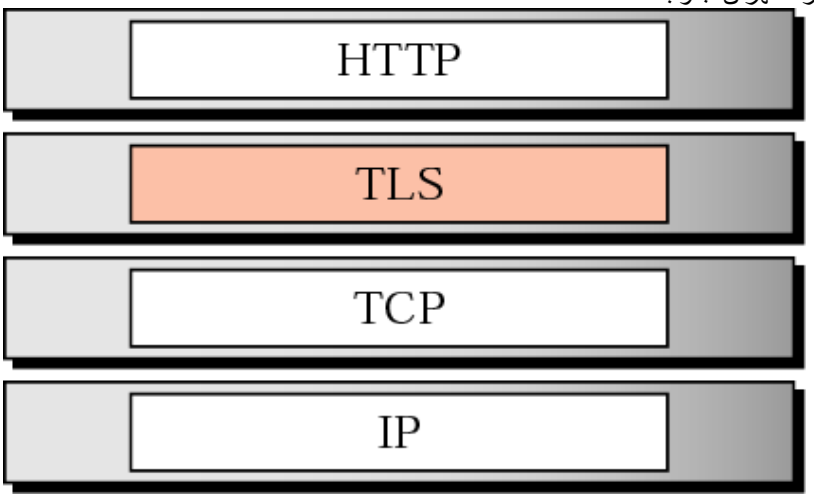
Pad length: ۸ بیت است که طول padding را مشخص می کند.

Authenticated: شامل فیلدهایی برا تضمین اعتبار است بر مبنای الگوی message digest

Padding: اطلاعات داخل Payload را حمل می کند.

Transport Layer Security (TLS):

TLS پروتکل این لایه است و جایگاه اصلی آن بین Trnsport و Application است.



شکل ۲۳۷: جایگاه TSI

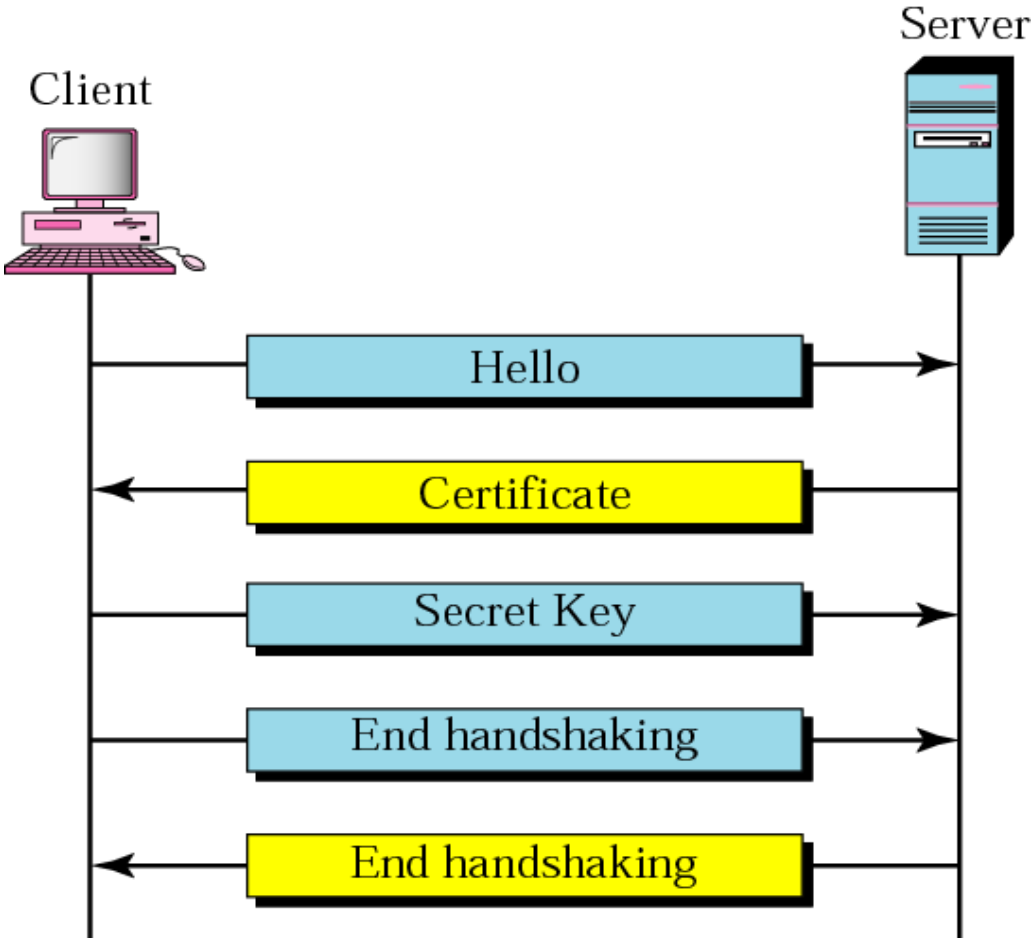
Position of TLS تضمین کننده امنیت اطلاعات است.

دارای دو پروتکل است:

۱. Hand shake protocol
۲. Data exchange protocol

پروتکل handshake:

برای اعتبار سنجی بین server و client است.



شکل ۲۳۸: پروتکل hand shake

درخواست hello از طرف کاربر به سرور فرستاده می شود. تاییدیه certification به کاربر فرستاده می شود (بحث مربوط به CA). از روی آن کاربر می تواند به کلید عمومی سرور دسترسی پیدا کند. کاربر که کلید عمومی سرور را دارد، یک عدد تصادفی را با کلید عمومی سرور کد کرده و به سرور می فرستد. تنها سرور می تواند با کلید خصوصی خود دیکد کند که در آن کلیدی است که کاربر تعیین کرده، سرور کلیدی را بدست می آورد که از این به بعد انتقال اطلاعات با آن کلید مشترک از طریق روش متقارن صورت می گیرد.

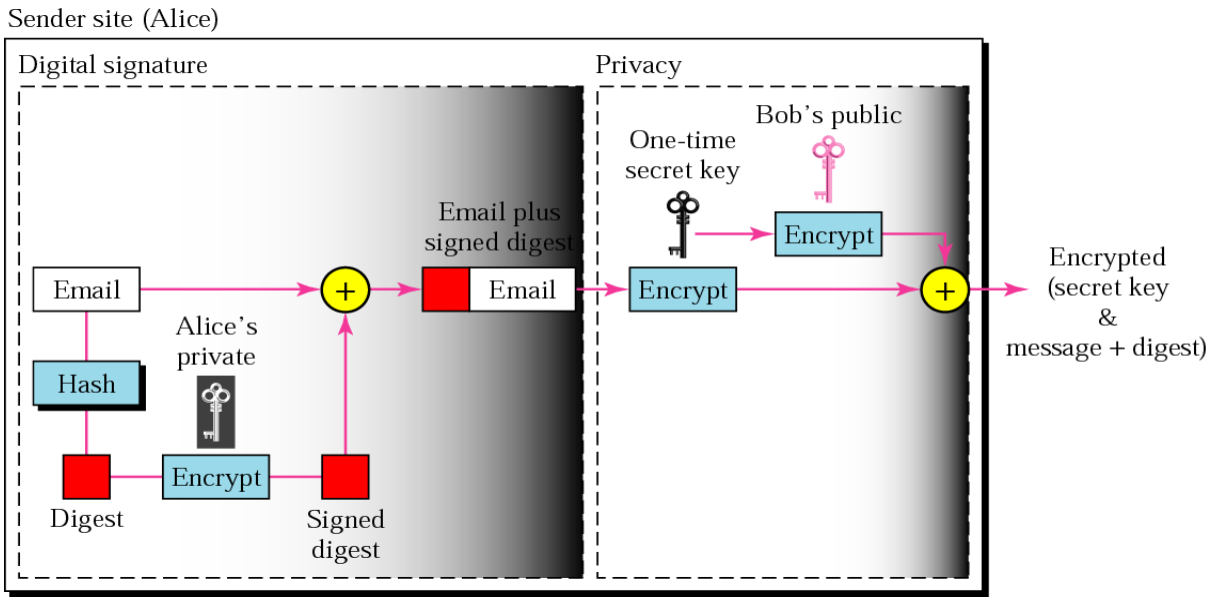
در انتها برای قطع ارتباط یک سیگنال end handshake از طرف سرور به کاربر فرستاده می شود.

توجه داشته باشید که با پایان یافتن session جاری تماماً از بین می رود.

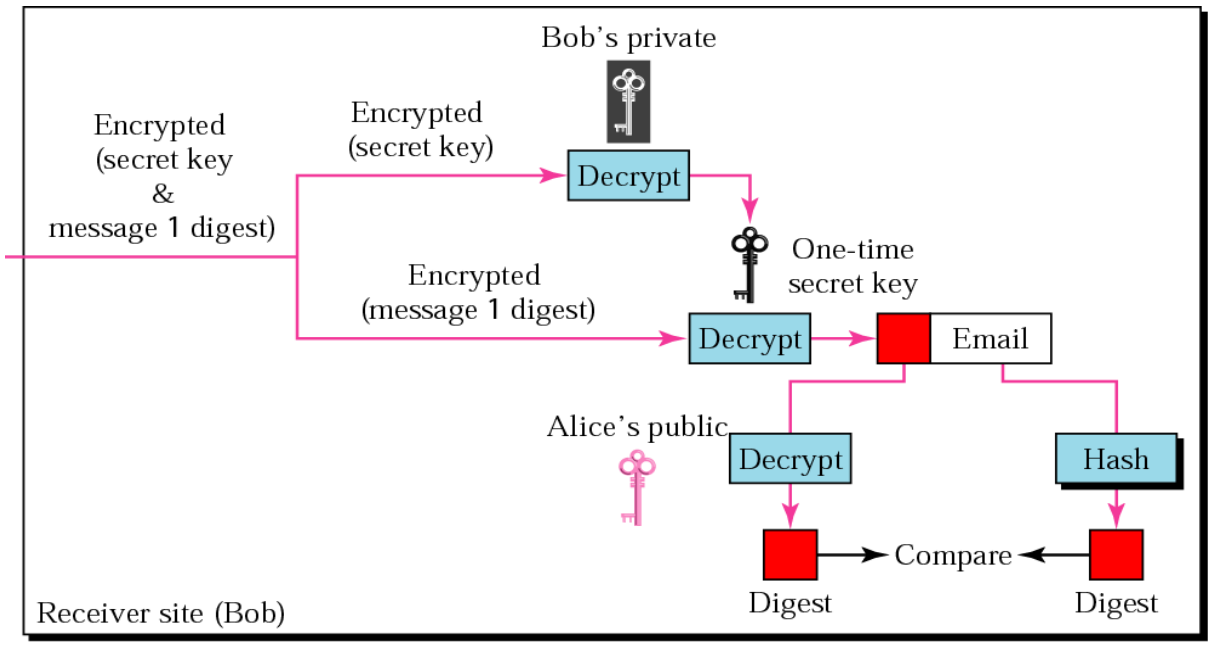
Application Layer Security:

پروتکل PGP (Pretty Good Privacy) بر روی mail پیاده سازی می شود و هر ۴ مورد صحت، سلامت، هویت سنجی و غیر قابل انکار بودن را روی mail پیاده سازی می کند.

در مثال زیر آلیس یک کلید لحظه ای می گیرد، اطلاعات را کد کرده و اطلاعات را با استفاده از کلید عمومی باب نیز کد می کند و هر دو را می فرستد. می توانستیم تنها با کلید عمومی باب کد کنیم اما اگر شخصی دوبار اطلاعات را از روی خط می گرفت و برای باب می فرستاد به جای آلیس قبولش می کرد به همین دلیل از onetime استفاده می کنیم چون در session تغییر می کند.



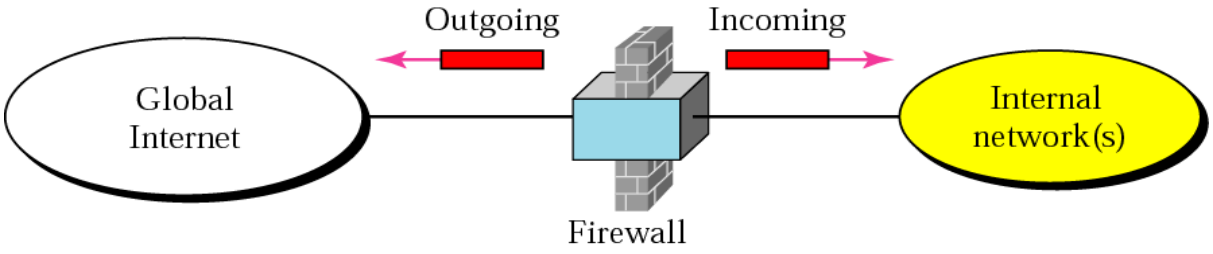
شکل ۲۳۹: پروتکل PGP در فرستنده



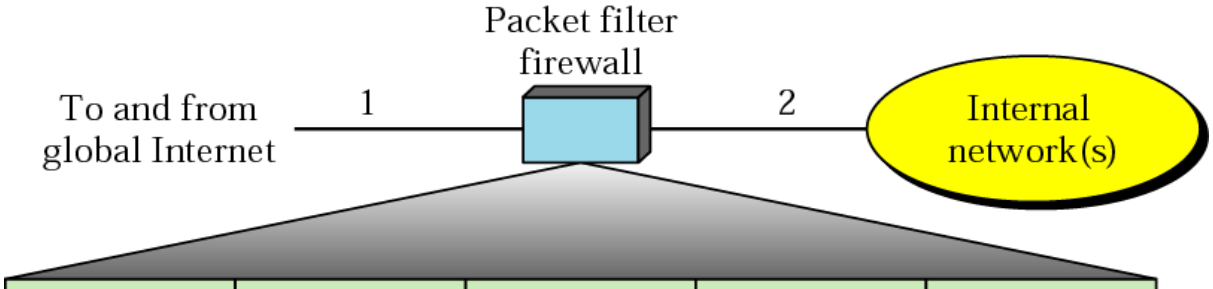
شکل ۲۴۰: پروتکل PGP در گیرنده

باب با کلید خصوصی خود بسته ارسالی از طرف آلیس را دیکد می کند و به دو بخش می رسد یکی بخـض اطلاعات خام و دیگری اطلاعات کد شده. اطلاعات کد شده را با استفاده از کلید عمومی آلـیس رمزگشـایی می کند و آنها را با هم مقایسه می کند اگر با هم برابر بودند خطا رخ نداده است.

Firewall:



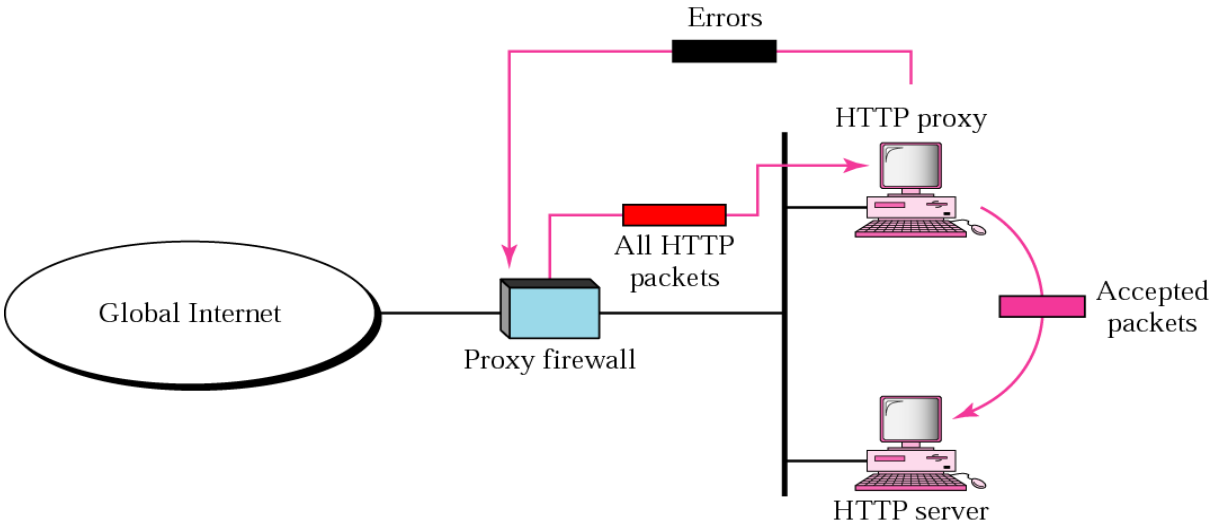
شکل ۲۴۱: firewall



Interface	Source IP	Source Port	Destination IP	Destination Port
1	131.34.0.0	*	*	*
1	*	*	*	23
1	*	*	194.78.20.8	*
2	*	80	*	*

شکل ۲۴۲: packet filter firewall

Proxy firewall:



شکل ۲۴۳: proxy firewall