



وب سایت گروه کامپیوتر دانشگاه آزاد اسلامی واحد تهران شرق
(قیامدشت)

www.Compg.ir



دانشگاه آزاد اسلامی
واحد تهران شرق

مهندسی اینترنت

مهندس سعید سلطانعلی

سید رسول موسوی فیه

۸۹۱۰۹۱۲۴۱

کارشناسی ناپیوسته

مهندسی کامپیوتر-نرم افزار

زمستان، بهار ۹۱-۱۳۹۰. روزهای پنجشنبه



۱.....	مروری بر شبکه
۱.....	اجزای اصلی شبکه
۱.....	دسته‌بندی بر اساس وسعت جغرافیایی
۱.....	انواع ارتباطات
۲.....	توپولوژی‌ها
۲.....	الف) BUS
۲.....	ب) Ring
۳.....	ج) Star
۳.....	د) Mesh
۳.....	ه) Tree
۴.....	و) Hybrid
۴.....	لایه‌ها
۴.....	مدل مرجع OSI
۵.....	۱- لایه فیزیکی (Physical Layer)
۵.....	۲- لایه پیوند داده‌ها (Data Link Layer)
۶.....	۳- لایه شبکه (Network Layer)
۶.....	۴- لایه انتقال (Transport Layer)
۶.....	۵- لایه جلسه (Session Layer)
۶.....	۶- لایه ارائه (Presentation Layer)
۷.....	۷- لایه کاربرد (Application Layer)
۷.....	مدل چهار لایه‌ای TCP/IP
۸.....	۱- لایه واسط شبکه (Network Interface)
۸.....	۲- لایه شبکه (Internet)
۸.....	۳- لایه انتقال (Transport)
۸.....	۴- لایه کاربرد (Application)
۸.....	انواع معماری‌های سرویس دهی در شبکه‌ها
۹.....	تفاوت سویچ و هاب
۹.....	پروتکل‌های لایه MAC (Media Access Control)
۹.....	۱- رقابتی (Contention Base)
۹.....	۲- غیر رقابتی (Contention Less)
۹.....	Aloha



۹.....	Slotted Aloha
۱۰.....	CSMA
۱۰.....	CSMA/CD
۱۱.....	ARQ (Automatic Repeat Request)
۱۱.....	Stop & Wait ARQ
۱۱.....	مکانیزم‌های Sliding Windows
۱۳.....	IP (Internet Protocol)
۱۳.....	سرآیند اینترنت پروتکل نسخه ۴ (Header IP v4)
۱۳.....	Version
۱۳.....	HL (IP Header Length)
۱۴.....	Type of Service
۱۴.....	Total Length
۱۴.....	Identification
۱۴.....	fragment offset
۱۴.....	TTL (Time To Live)
۱۵.....	Protocol
۱۵.....	Header Check Sum
۱۵.....	Destination و Source
۱۵.....	Option
۱۵.....	Pay Load
۱۶.....	کلاس IP نسخه چهار
۱۶.....	کلاس A
۱۶.....	کلاس B
۱۶.....	کلاس C
۱۶.....	کلاس D
۱۶.....	کلاس E
۱۷.....	آدرس‌های شخصی (Private IP)
۲۰.....	Sub netting (زیر شبکه سازی)
۲۰.....	زیر شبکه سازی با استفاده از Subnet Mask
۲۳.....	دستورات:
۲۴.....	Super netting (ادغام زیر شبکه‌ها)



۲۶.....	Default Gateway
۲۷.....	دستورات
۲۸.....	پروتکل (Internet Control Message Protocol) ICMP
۲۸.....	ICMP Header
۲۸.....	برخی از پیغام‌های ICMP
۲۹.....	پروتکل (Address Resolution Protocol) ARP
۳۰.....	پروتکل (Dynamic Host Configuration Protocol) DHCP
۳۲.....	مسیریابی (Routing)
۳۲.....	مسیریابی در دو بخش صورت می‌گیرد:
۳۲.....	تفاوت Routing و Switching
۳۲.....	انواع Switching
۳۳.....	مسیریابی به روش Virtual Circuit Switching (VC)
۳۳.....	مسیریابی به روش Datagram
۳۳.....	الگوریتم‌های مسیریابی
۳۳.....	بر اساس جمع‌آوری اطلاعات
۳۳.....	۱- متمرکز (سراسری)
۳۳.....	۲- غیر متمرکز
۳۳.....	بر اساس هوشمندی
۳۳.....	۱- ایستا (Static)
۳۴.....	۲- پویا (Dynamic)
۳۴.....	بر اساس زمان اجرا
۳۴.....	۱- پیش‌دستانه (Pro Active)
۳۴.....	۲- واکنشی (On Demand یا Re Active)
۳۴.....	Wi-Fi
۳۴.....	DCF
۳۵.....	مسیریابی سیل‌آسا (Flooding)
۳۵.....	روش Link State
۳۶.....	روش Distance Vector
۳۶.....	مسیریابی در اینترنت (مسیریابی سلسله‌مراتبی)
۳۹.....	لایه‌ی انتقال
۳۹.....	وظایف تعریف شده برای لایه‌ی Transport



۳۹	کاستی‌های IP
۳۹	راهکارهای TCP
۴۰	سرآیند پروتکل TCP
۴۰	فیلد Source Port
۴۰	فیلد Destination Port
۴۱	فیلد Sequence Number
۴۱	فیلد Acknowledgment number
۴۱	فیلد TCP Header Length
۴۱	بیت‌های Flag
۴۲	فیلد Window size
۴۲	فیلد Checksum
۴۲	فیلد TCP Segment length
۴۲	فیلد Urgent Pointer
۴۲	پروتکل دست‌تکانی سه‌مرحله‌ای (3-Way Handshaking)
۴۴	مکانیزم کنترل ازدحام در TCP (Congestion Control)
۴۶	زمان‌سنج‌ها در TCP
۴۶	Retransmission Timer (RT)
۴۶	Keep Alive
۴۶	Persistent Timer
۴۶	Quite Timer
۴۶	Idle Timer
۴۷	سوکت (Socket)
۴۷	ایجاد TCP Socket از نوع سنکرون و تک‌نخی (Single Thread)
۴۷	نسخه‌ی سرور برای چت دو نفره‌ی ساده
۴۸	نسخه‌ی کلانیت برای چت دونفره‌ی ساده
۴۹	سیستم نام‌گذاری دامنه (Domain Name System) یا DNS
۵۰	روش تکراری (Iterative)
۵۰	روش بازگشتی (Recursive)
۵۱	روش معکوس (Reverse)
۵۲	DNS Cache
۵۲	URL



۵۲.....	ساختار بانک اطلاعاتی DNS Server
۵۴.....	پروتکل انتقال فایل (File Transfer Portocol [FTP])
۵۴.....	حالت Normal
۵۵.....	حالت Passive
۵۵.....	پروتکل TFTP (Trivial File Transfer Protocol)





مروری بر شبکه

اجزای اصلی شبکه

- فرستنده و گیرنده (کاربر)

- رسانه انتقال (Media)

- داده

- پروتکل (قراردادها و استانداردها برای هماهنگی در شبکه)

- سخت‌افزار و نرم‌افزار شبکه

دسته‌بندی شبکه‌ها: از دیدگاه‌های مختلف شبکه‌ها بررسی می‌شوند که معروفترین آنها وسعت جغرافیایی است؛

دسته‌بندی بر اساس وسعت جغرافیایی

* Personal Area Network (PAN)

* (Local Area Network) LAN: همانند PAN است اما تعداد کامپیوترها بیشتر است.

* CAN: در محدوده‌ی خاص مانند دانشگاه (پردیس)

* (Metropolitan Area Network) MAN: شبکه شهری

* WAN: بیشتر از یک شهر Wide Area Network

* Internet: ترکیبی از کلیه شبکه‌ها

← اگر اینترنت با I بزرگ نوشته شود تعریف آن همان اینترنت جهانی است؛ اما اگر با i کوچک نوشته شود به معنی مفهوم آن است و ممکن در کل جهان نباشد.

انواع ارتباطات

اتصال‌گرا: Connection Oriented - قبل از ارسال داده‌ها، گیرنده باید Accept نماید، (توافق ارتباط بین طرفین. مانند ارتباط تلفنی)

بدون اتصال: فرستنده بدون آگاهی گیرنده می‌تواند داده را ارسال کند. (مانند ارسال SMS)

مطمئن: فرستنده از دریافت داده توسط گیرنده مطلع می‌گردد. (مانند تأییدیه دریافت SMS)

نامطمئن: فرستنده از سرنوشت داده دریافتی توسط گیرنده مطلع نگردد.

* ارتباطات ترکیبی از دو گروه می‌تواند باشد، مانند اتصال‌گرای مطمئن یا بدون اتصال نامطمئن و ...

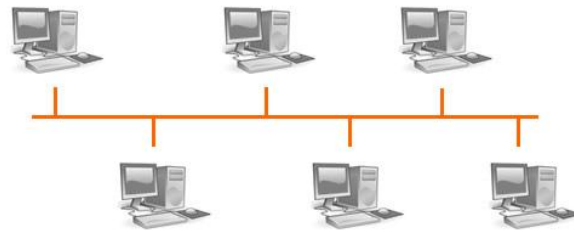


توپولوژی‌ها

نحوه‌ی اتصال فیزیکی شبکه را هم‌بندی یا توپولوژی گویند.

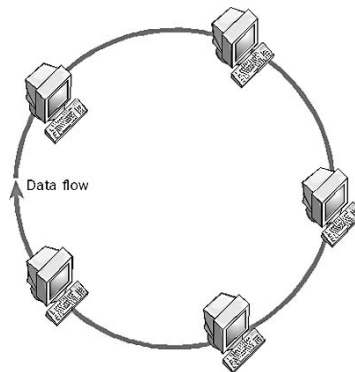
BUS (الف)

در این نوع توپولوژی تمام ایستگاه‌ها از طریق یک کانال فیزیکی مشترک به همدیگر متصل شده‌اند و هرگونه تبادل اطلاعات از طریق این کانال انجام خواهد شد. این توپولوژی به دلیل سادگی در نصب و راه‌اندازی و ارزان بودن یکی از شبکه‌های پر رونق دنیا محسوب می‌شود. ولی امروزه جای خود را به انواع دیگر داده است. از معایب این توپولوژی این است که در صورت از کار افتادن یک ایستگاه کل شبکه از کار خواهد افتاد.



Ring (ب)

در توپولوژی حلقه، ایستگاه‌ها در یک ساختار بسته‌ی حلقوی به یکدیگر متصل هستند. این توپولوژی همانند خطی (Bus) است.





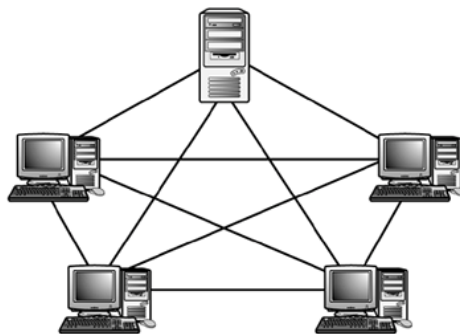
ج) Star

در این توپولوژی ارتباط تمامی ماشین‌های شبکه از طریق یک گرهی مرکزی برقرار می‌شود. این گره می‌تواند یک سویچ سریع و هوشمند و یا یک هاب و حتی یک کامپیوتر باشد. این توپولوژی امروزه رونق بسیاری گرفته و جایگزین مدل خطی شده است.



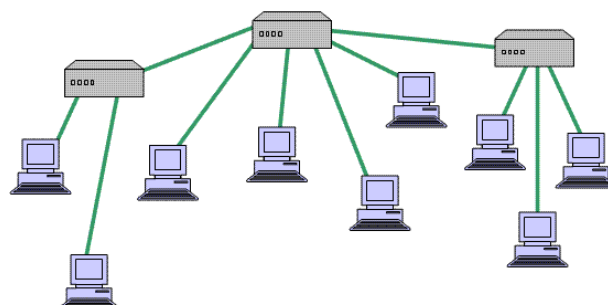
د) Mesh

این شبکه شبیه به یک گراف کامل است یعنی از هر کامپیوتر به کامپیوتر دیگر چندین مسیر وجود دارد. به علت اینکه چندین مسیر برای ارسال و دریافت اطلاعات وجود دارد از ضریب اطمینان بیشتری نسبت به خرابی برخوردار است. این توپولوژی امکان توزیع بسته‌های ارسالی (Load Balancing) در صورت بار زیادی را دارا می‌باشد. از معایب این روش هزینه‌ی زیاد آن است.



ه) Tree

به صورت یک شبکه‌ی درختی است. از مزایای این شبکه این است که ترافیک هر بخش در خود آن بخش محدود است.





Hybrid (و)

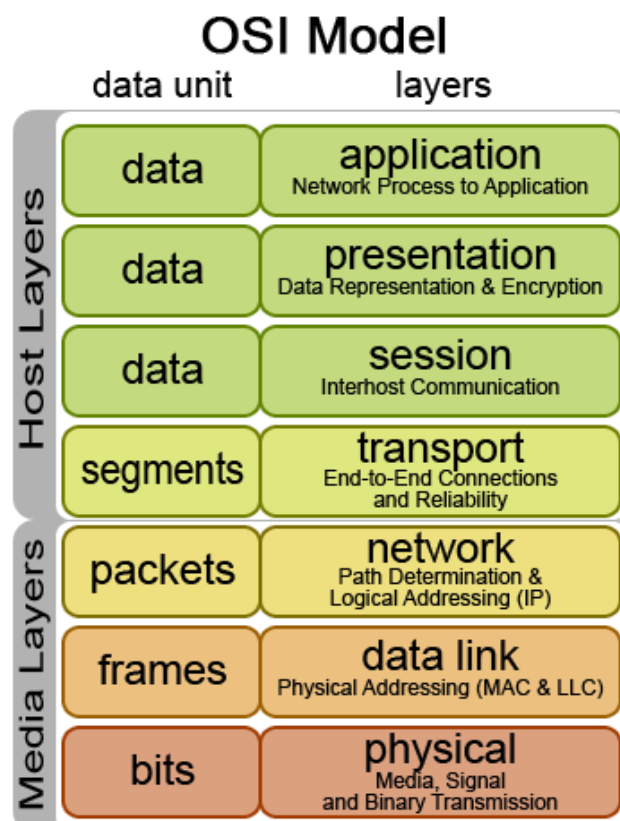
تلفیقی از تمامی شبکه ها است.

لایه‌ها

چون ماشین‌های فرستنده و گیرنده‌ی متعددی در یک شبکه وجود دارد مسائلی مثل ازدحام، تداخل و تصادم در شبکه بوجود می‌آید که این مشکلات به‌همراه مسائل دیگر باید در سخت‌افزار و نرم‌افزار شبکه حل شود. به دلیل اینکه مسائل و مشکلات دارای ماهیت‌های متفاوتی از یکدیگر هستند، طراحی شبکه به صورت لایه به لایه صورت می‌گیرد؛ و هر لایه وظیفه‌ی مخصوص به خود را خواهد داشت.

مدل مرجع OSI

برای آنکه طراحی شبکه‌های سلیقه‌ای و پیچیده نشود سازمان جهانی استاندارد ISO، مدلی هفت لایه‌ای برای شبکه ارائه کرد، به گونه‌ای که وظایف و خدمات شبکه در این هفت لایه به صورت مجزا تعریف شده‌اند. این مدل بنام مدل مرجع OSI (Open System Interconnection) معروف است. لایه بندی برای ۱- تقسیم و وظایف و ۲- استقلال آنها از هم صورت می‌گیرد. به عبارت دیگر لایه بندی برای تقسیم وظایف و انجام درست آنها صورت می‌گیرد. هفت لایه‌ی این مدل عبارتند از: ۱- لایه‌ی فیزیکی ۲- لایه‌ی پیوند داده‌ها ۳- لایه‌ی شبکه ۴- لایه‌ی انتقال ۵- لایه‌ی جلسه ۶- لایه‌ی نمایش (ارائه) ۷- لایه‌ی کاربرد.





۱- لایه‌ی فیزیکی (Physical Layer)

وظیفه‌ی اصلی در لایه‌ی فیزیکی انتقال بیت‌ها بصورت سیگنال الکتریکی و ارسال آن بر روی کانال می‌باشد. واحد اطلاعات در این لایه بیت است. بنابراین این لایه هیچ اطلاعی از محتوای پیام ندارد. پارامترهایی که باید در این لایه در نظر گرفته شوند عبارتند از:

- ❖ ظرفیت کانال فیزیکی و نرخ ارسال
 - ❖ نوع مدولاسیون
 - ❖ چگونگی کوپلاژ با خط انتقال
 - ❖ مسائل مکانیکی و الکتریکی مانند نوع کابل، باند فرکانسی، و نوع کانکتور کابل.
- این لایه هیچ وظیفه‌ی در رابطه با تشخیص و ترمیم خطا ندارد.

۲- لایه‌ی پیوند داده‌ها (Data Link Layer)

وظیفه‌ی این لایه آن است که با استفاده از مکانیزم‌های کشف و کنترل خطا، داده‌ها را روی یک کانال انتقال که ذاتاً دارای خطا است، بدون خطا و مطمئن به مقصد برساند. به عبارت دیگر وظیفه‌ی این لایه بیمه‌ی اطلاعات در مقابل خطاهای احتمالی است؛ زیرا ماهیت خطا طوری است که قابل رفع نیست اما با تدابیری می‌توان فرستنده را از رسیدن یا نرسیدن اطلاعات به طور صحیح به گیرنده مطلع کرد که وی در صورت بروز خطا مجدداً اقدام به ارسال اطلاعات کند. (خطایابی فیزیکی).

از دیگر وظایف لایه‌ی پیوند داده‌ها این است که اطلاعات ارسالی از لایه‌ی بالاتر را به واحدهای استاندارد و کوچکتری شکسته و ابتدا و انتهای آن را از طریق نشانه‌های خاصی که Delimiter نام دارند، مشخص کند. این قالب استاندارد فریم (Frame) نام دارد.

کشف خطا می‌تواند از طریق اضافه کردن بیت‌های کنترل خطا مانند بیت‌های Parity Check و Checksum و CRC انجام شود.

یکی دیگر از وظایف لایه‌ی دوم کنترل جریان یا به عبارتی تنظیم جریان ارسال فریم‌ها به گونه‌ای است که یک دستگاه گُند هیچ‌گونه فریمی را به خاطر آهسته بودن از دست ندهد.

یکی دیگر از وظایف این لایه اطلاع رسانی به فرستنده از وصل یا عدم وصول داده‌ها است.

از وظایف دیگر لایه‌ی پیوند داده‌ها، وضع قراردادهایی برای جلوگیری از تصادم سیگنال ایستگاه‌هایی که از کانال مشترک استفاده می‌کنند، است.

وقتی یک واحد اطلاعاتی تحویل یک ماشین متصل به کانال فیزیکی در شبکه شد، وظیفه‌ی این لایه پایان می‌یابد. از دیدگاه این لایه ماشین‌هایی که به کانال فیزیکی متصل نمی‌باشند، در دسترس نیستند. کنترل سخت‌افزار لایه‌ی فیزیکی به عهده‌ی این لایه است. وظایف این لایه با استفاده از سخت‌افزارهای دیجیتال انجام می‌شود.



۳- لایه‌ی شبکه (Network Layer)

در این لایه اطلاعات به صورت بسته‌هایی سازماندهی می‌شود (Packet). و برای انتقال مطمئن تحویل لایه‌ی دوم می‌شود. با توجه به اینکه بین دو ماشین در شبکه مسیرهای گوناگونی وجود داشته باشد، لذا وظیفه‌ی این لایه این است که هر بسته‌ی اطلاعاتی را پس از دریافت به مسیری هدایت کند تا بسته بتواند به مقصد برسد. در این لایه تدابیری اندیشیده می‌شود که از ازدحام (ترافیک بیش از اندازه‌ی بسته‌ها در یک مسیر یا سویچ) جلوگیری شده و از ایجاد بن بست ممانعت بعمل آید. در این لایه تمام ماشین‌های شبکه دارای یک آدرس جهانی منحصر بفرد هستند، که هر ماشین بر اساس این آدرسها اقدام به هدایت بسته‌ها به سمت مقصد خواهد کرد.

وظایف این لایه به سیستم نام‌رسانی تشبیه شده است؛ یعنی آنکه پاکت حاوی نامه پس از درج مشخصات به صندوق پست انداخته خواهد شد، بدون آنکه زمان دقیق رسیدن نامه و وجود گیرنده‌ی نامه در مقصد از قبل حدس زده شود. در ثانی ممکن است نامه گم شود و یا به اشتباه در راهی بیفتد که مدت‌ها در مسیر باشد و زمانی به گیرنده برسد که هیچ ارزشی نداشته باشد. در این لایه تضمینی وجود ندارد که وقتی بسته ای برای مقصد ارسال می‌شود، مقصد آمدگی دریافت آن را داشته باشد و بتواند آنرا دریافت کند؛ و یا تضمینی وجود ندارد که بسته‌ها به همان ترتیب ارسال به مقصد برسند. و یا حتی ممکن است وقتی بسته ای به مقصدی ارسال گردید به دلیل تاخیر در رسیدن از اعتبار ساقط شود و مجدداً ارسال شود و هر دو بسته (جدید و قدیم) همزمان به مقصد برسند.

۴- لایه‌ی انتقال (Transport Layer)

در این لایه بر اساس خدمات لایه‌ی زیرین، یک سرویس انتقال بسیار مطمئن و اتصالگرا ارائه می‌شود. تمام مشکلات لایه‌ی شبکه در این لایه حل و فصل می‌شود. قبل از ارسال بسته‌ها، نرم‌افزار این لایه اقدام به ارسال یک بسته‌ی ویژه می‌کند تا مطمئن شود که ماشین مقصد آماده‌ی دریافت اطلاعات است.

جریان ارسال اطلاعات شماره‌گذاری شده تا هیچ بسته‌ای گم نشود یا دو بار دریافت نگردد. ترتیب جریان بسته‌ها حفظ می‌شود. واحد اطلاعات در این لایه قطعه (Segment) است.

۵- لایه‌ی جلسه (Session Layer)

وظیفه‌ی این لایه فراهم آوردن شرایط یک جلسه (نشست) همانند ورود به سیستم از راه دور، احراز هویت طرفین، نگهداری این نشست و از سرگیری یک نشست در هنگام قطع ارتباط می‌باشد. واحد اطلاعاتی این لایه پیام (message) است.

۶- لایه‌ی ارائه (Presentation Layer)

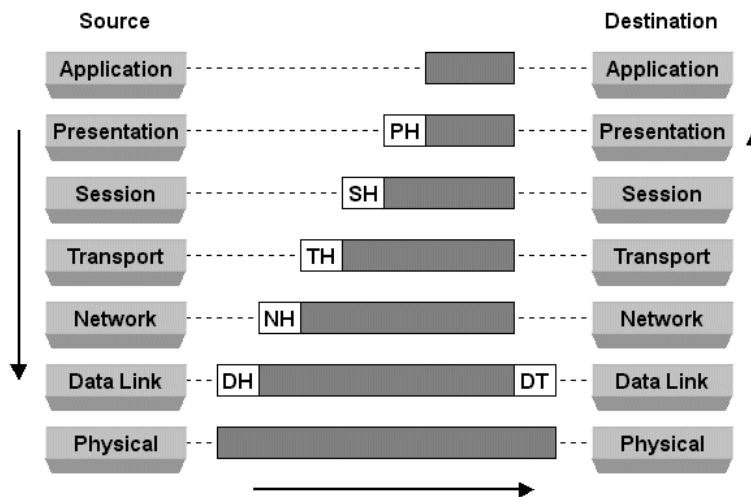
در این لایه کارهایی از قبیل: فشرده سازی فایل - رمزنگاری برای ارسال داده‌های محرمانه - رمزگشایی - تبدیل کدها به یکدیگر وقتی دو ماشین از استانداردهای مختلفی برای متن استفاده می‌کنند. (مانند تبدیل EBCDIC به ASCII و بلعکس)



۷- لایه‌ی کاربرد (Application Layer)

در این لایه استاندارد مبادله‌ی پیام بین نرم‌افزارهایی که در اختیار کاربر بوده و به نحوی با شبکه در ارتباطند تعریف می‌شود. لایه‌ی کاربرد شامل تعریف استانداردهایی نظیر: انتقال نام‌های الکترونیکی - انتقال مطمئن فایل - دسترسی به بانک‌های اطلاعاتی راه دور - مدیریت شبکه و انتقال صفحات وب است.

در مدل لایه‌ای شبکه، وقتی یک برنامه‌ی کاربردی در لایه‌ی آخر اقدام به ارسال یک واحد اطلاعات می‌نماید، سرآیند لازم به آن اضافه شده و به لایه‌ی زیرین تحویل می‌دهد؛ آن لایه نیز سرآیند خود را اضافه کرده و تحویل لایه‌ی زیرین خود می‌نماید و این امر ادامه پیدا کرده آن واحد روی کانال فیزیکی ارسال شود. در مقصد عکس این عمل انجام می‌شود.

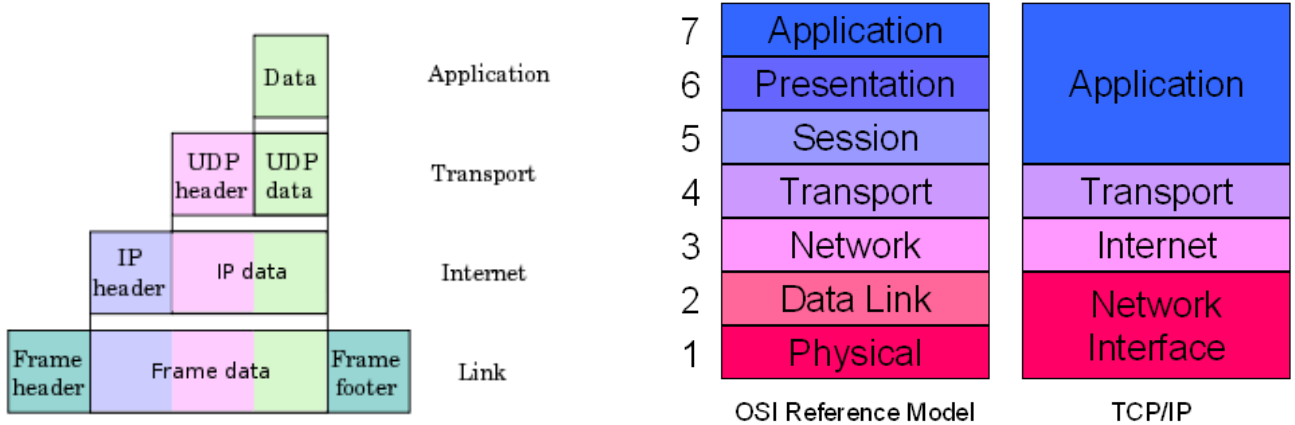


* کنترل جریان: سرعت بین گیرنده و فرستنده را تنظیم می‌کند، اما کنترل ازدحام شبکه را در نظر می‌گیرد.

مدل چهار لایه‌ای TCP/IP

این مدل زاده‌ی جنگ سرد است. برای اولین بار روش سویچ بسته در این شبکه معرفی شد. امروزه TCP/IP به عنوان محبوبترین پروتکل شبکه در تمام سیستم‌های عامل حمایت می‌شود. این مدل یک ساختار چهار لایه‌ای برای ارتباط گسترده تعریف می‌نماید. پشته‌ی TCP/IP مجموعه‌ای شامل بیش از صد پروتکل متفاوت است که برای سازماندهی کلیه‌ی اجزاء شبکه‌ی اینترنت به کار می‌رود.

در مقایسه این مدل با مدل مرجع OSI، لایه‌ی اول، یعنی لایه‌ی دسترسی به شبکه فیزیکی به صورت تلفیقی از وظایف لایه‌ی فیزیکی و پیوند داده‌ها در OSI است. لایه‌ی دوم TCP/IP معادل لایه‌ی سوم از مدل OSI یعنی لایه‌ی شبکه است؛ لایه‌ی سوم همنام و معادل لایه‌ی چهارم در مدل OSI یعنی لایه‌ی انتقال بوده و و لایه‌ی پنجم و ششم در مرجع OSI در این مدل (TCP/IP) نبوده و وظایف آنها در صورت لزوم در لایه‌ی چهارم از مدل TCP/IP ادغام شده است. لایه‌ی هفتم نیز معادل بخشی از لایه‌ی چهارم مدل TCP/IP است.



۱- لایه‌ی واسط شبکه (Network Interface)

در این لایه استانداردهای سخت‌افزار، نرم‌افزارهای راه‌انداز و پروتکل‌های شبکه تعریف می‌شود. این لایه درگیر با مسائل فیزیکی، الکتریکی و مخابراتی کانال انتقال، نوع کارت شبکه و راه‌اندازهای لازم برای نصب کارت شبکه می‌باشد.

۲- لایه‌ی شبکه (Internet)

این لایه در ساده‌ترین عبارت وظیفه دارد بسته‌های اطلاعاتی (بسته‌های IP) را روی شبکه هدایت کند و از مبدأ به مقصد به پیش برسد. در این لایه چندین پروتکل وظیفه‌ی این نقل و انتقال را دارند. (مسیریابی و تحویل بسته‌ها). کلیدی‌ترین پروتکل در این لایه IP است. و برخی از پروتکل‌های مهم در این لایه عبارتند از: BOOTP – IGMP – ICMP – RIP – RARP – ARP و... . یک واحد اطلاعاتی که باید تحویل مقصد شود در این لایه دیتاگرام نامیده می‌شود؛ پروتکل IP می‌تواند یک دیتاگرام را در قالب بسته‌های کوچکتری قطعه قطعه کرده و پس از اضافه کردن اطلاعات لازم برای بازسازی، آنها را روی شبکه ارسال کند.

۳- لایه‌ی انتقال (Transport)

این لایه ارتباط ماشین‌های انتهایی (ماشین‌های میزبان) را در شبکه برقرار می‌کند؛ یعنی می‌تواند بر اساس سرویسی که لایه‌ی دوم ارائه می‌کند یک ارتباط اتصال گرا و مطمئن برقرار کند.

۴- لایه‌ی کاربرد (Application)

در این لایه بر اساس خدمات لایه‌های زیرین، سرویس سطح بالایی برای خلق برنامه‌های کاربردی ویژه و پیچیده ارائه می‌شود. این خدمات در قالب پروتکل‌های استاندارد از قبیل: Telnet - انتقال فایل یا FTP - مدیریت پست الکترونیکی - خدمات انتقال صفحات ابرمتنی و ... است.

انواع معماری‌های سرویس دهی در شبکه‌ها

(الف) نظیر به نظیر (Peer To Peer)

(ب) سرویس دهنده / سرویس گیرنده (Client – Server)



* سرویس‌ها در لایه‌ی Application هستند.

تفاوت سویچ و هاب

سویچ در لایه‌ی پیوند داده‌ها کار می‌کند و هوشمند است، و پس از پردازش سرآیند پیوند داده و گرفتن آدرس مقصد، بسته را فقط به مقصد ارسال می‌کند، اما هاب در لایه‌ی فیزیکی کار کرده و یک بسته را به تمام اتصالات به جز اتصالی که بسته از آن ارسال شده ارسال می‌کند.

سویچ به صورت Reverse Learning عمل می‌کند و پس از مدتی یاد می‌گیرد که چه پورتی به چه کامپیوتری وصل است و این کار از روی Switch Table صورت می‌گیرد.

پورت	آدرس MAC
1	E00:A1:11:88
2	.
.	.
.	.

پروتکل‌های لایه‌ی MAC (Media Access Control)

پروتکل‌های کنترل دسترسی به رسانه هستند که به دو گروه زیر تقسیم می‌شوند؛

۱- رقابتی (Contention Base)

از این پروتکل در Lan استفاده می‌شود. معروفترین این پروتکل‌ها Aloha, Slotted Aloha, CSMA, CSMA/CD, CSMA/CA هستند. این مدل پروتکل بر پایه‌ی Baseband است و هر سیستمی که برنده‌ی رقابت شد از آن استفاده می‌کند.

۲- غیر رقابتی (Contention Less)

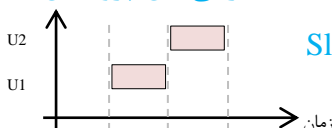
معمولاً یک Node وجود دارد که مشخص می‌کند که ارسال توسط چه ایستگاهی انجام شود، یا استفاده از الگوریتم نوبت دهی زمانی، یا نوبت دهی فرکانسی؛ این مدل پروتکل در شبکه‌های WAN و MAN استفاده می‌شود.

Aloha

در جزایرهاوایی برای اولین بار استفاده شد، عملکرد آن بدین صورت است که بسته را ارسال کرده و اگر تصادم رخ داد به ارسال کننده اطلاع می‌دهد که بسته کامل به مقصد نرسیده است. کارایی این پروتکل ۱۸٪ است و عیب آن این است که در صورت وقوع تصادم بین دو بسته، هر دوی آنها از بین خواهند رفت.

Slotted Aloha

اصلاح شده‌ی مدل قبل است و زمان را برش می‌دهند، هر کاربری که خواهان ارسال بسته‌ای باشد باید منتظر رسیدن به ابتدای Slot باشد. از معایب این پروتکل این است که فریم‌ها باید به اندازه‌ی برش‌های زمان باشند. کارایی این پروتکل ۳۶٪ است. از دیگر معایب آن این است که اگر دو کاربر همزمان خواهان ارسال باشند در ابتدای Slot تصادم رخ خواهد داد.





CSMA

مشکل حالت قبل در این پروتکل حل شده است و قبل از ارسال خط را بررسی کرده اگر خط خالی باشد داده ارسال می‌گردد در غیر اینصورت منتظر می‌ماند.

در این پروتکل تصادم هنگامی که خط خالی باشد و دو بسته همزمان ارسال گردند، رخ می‌دهد.

* برای برطرف کردن مشکل فوق ارسال بر اساس احتمال دادن صورت می‌گیرد؛ یعنی به طور مثال اگر عدد مورد نظر رؤیت شد ارسال صورت می‌گیرد در غیر اینصورت در حالت انتظار باقی می‌ماند تا احتمال بعدی؛ به این شکل درصد تصادم کمتر می‌گردد.

CSMA/CD

این پروتکل پس از ارسال بسته باز به شنود خط ادامه داده و در صورت تصادم ارسال را قطع می‌کند، در اینجا هدر رفت خط در هنگام تصادم کمتر می‌شود.

← **Hidden Terminal**: این مشکل در شبکه‌های بیسیم صورت می‌گیرد، فرض کنید چهار دستگاه بیسیم A, B, C, D به ترتیب وجود داشته باشند و به طور مثال ترمینال C توانایی ارتباط با A را نداشته باشد اما هر دوی آنها با ترمینال B در ارتباط هستند، اکنون دستگاه A در حال ارسال داده برای B است و هم زمان دستگاه C که اطلاعی از این موضوع ندارد شروع به ارسال داده برای B می‌کند، فلذا ترمینال B در هنگام دریافت دچار تصادم می‌شود چرا که از دو ترمینال همزمان در حال دریافت داده است.

← **Exposit Terminal**: بر اساس مثال بالا (چهار ترمینال وجود دارد). اکنون فرض را بر این گذاشته که B در حال ارسال داده برای A است و از آنجا که C در محدوده‌ی B است اما ترمینال D در آن محدوده پشتیبانی نمی‌شود. با توجه به این موضوع اگر C خواهان ارسال داده به D باشد به علت مشغول بودن خط توسط B (وجود در یک محدوده) ارسال صورت نمی‌گیرد، در صورتی که همانگونه که گفته شد فرکانسهای D و B با یکدیگر تداخل ندارند. این مشکل را Exposit Terminal گویند.

برای حل مشکل Hidden Terminal از حالت CSMA/CA استفاده می‌کنیم، به این صورت که هر فرستنده قبل از ارسال به گیرنده ابتدا درخواستی به گیرنده ارسال می‌کند (RTS) و در صورت آزاد بودن گیرنده، فرستنده شروع به ارسال می‌کند (CTS).

برای حل مشکل Exposit Terminal نیز از (RTS) و (CTS) استفاده می‌شود.

* برای RTS و CTS یک بازه‌ی اصلاح زمانی وجود دارد و همیشه در بازه‌ی زمانی خاصی ارسال می‌شود.

* به CSMA/CA که برای شبکه‌های بیسیم کاربرد دارد به صورت مخفف MACA گفته می‌شود.

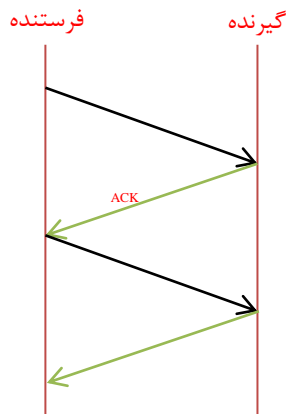


ARQ (Automatic Repeat Request)

- این روش در ارتباط مطمئن کار برد دارد، به این صورت که اگر بسته ارسال نگردید، مجدداً ارسال خواهد شد، این عمل بر اساس مکانیزمهایی که در ادامه گفته شده انجام می شود. (قبل از پرداختن به مکانیزمها یادآوری‌های زیر را در نظر بگیرید؛)
- * Simplex: پروتکل یک طرفه (ارسال فقط در یک جهت صورت می گیرد)
 - * Half Duplex: کانال دو طرفه‌ی ناهمزمان (به ازای هر ارسال از فرستنده گیرنده پاسخ می‌دهد).
 - * Full Duplex: کانال دوطرفه (به ازای ارسال گروهی از داده‌ها گیرنده پاسخ خواهد داد).

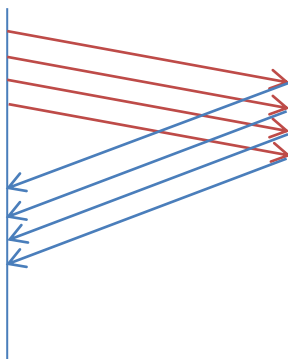
Stop & Wait ARQ

ابتدا یک فریم ارسال کرده و منتظر ACK می‌ماند و به همین ترتیب به کار خود ادامه داده تا بسته به صورت کامل به مقصد برسد. این مکانیزم در Half Duplex کاربرد دارد. مشکل روش فوق تأخیر زمانی آن است.



مکانیزم‌های Sliding Windows

در این حالت فرستنده به اندازه‌ی W فریم ارسال کرده و سپس منتظر ACK می‌ماند. در مثال زیر $W=4$ است؛



- از زیر گروه‌های این مکانیزم Go Back N ARQ و Selective Repeat ARQ است.
- Go Back N ARQ: در این حالت اگر یک فریم درست دریافت نشود مابقی را نیز ارسال نمی‌کند و مجبور می‌شود که همه را دوباره به ترتیب ارسال کند.
- مثلاً اگر ۱، ۲، ۳، ۴ ارسال و ACK دریافت شد؛ سپس ۵، ۶، ۷، ۸ نیز ارسال گردد اما در هنگام دریافت ۵ دریافت نشود، مجبور است گروه دوم را دوباره ارسال کند!
- Selective Repeat ARQ: در این مکانیزم فقط فریم گم شده دوباره ارسال می‌شود.
- * برای جلوگیری از گم شدن داده‌ها برای هر بسته یک شماره ترتیب قرار می‌دهیم.
 - * شماره ترتیب نیز باید بهینه انتخاب گردد؛ و این بهینه بهتر است دو برابر طول W باشد.
 - * در Selective Repeat ARQ به اندازه‌ی W فرستنده، گیرنده خارج از ترتیب بسته قبول می‌کند.
 - * مکانیزم‌های فوق در حالت Full Duplex کاربرد دارند.



* برای کم کردن زمان انتظار در Go Back N اگر عدد ترتیبی بعدی رسید اما قبلی نرسید؛ NACK، یعنی عدم دریافت بسته‌ی قبلی را به فرستنده می‌دهد که زمان انتظار کمتر گردد و فرستنده گروه داده‌ی مورد نظر را دوباره ارسال نماید.



پس از مقدمه‌ی فوق‌الذکر به بررسی مباحث مهم در پروتکل TCP/IP می‌پردازیم. در این مدل بیشترین اتفاقات در لایه‌های میانی، یعنی Inter Network و Transport صورت می‌گیرد؛ که در لایه‌ی اولی IP مطرح است و در لایه‌ی دیگر UDP و TCP. در ادامه به بررسی کامل موارد فوق خواهیم پرداخت.

IP (Internet Protocol)

در حال حاضر آی‌پی دارای دو نسخه‌ی ۴ و ۶ است، که نسخه‌ی ۶ را به دلیل عدم عملیاتی نشدن مورد بررسی قرار نخواهیم داد.

* استانداردهایی که پروتکل TCP/IP استفاده می‌کند RFC نام دارند.

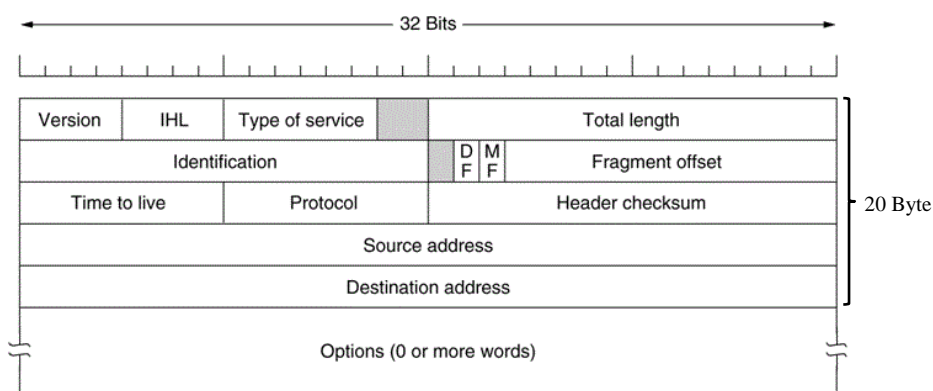
* تمام استانداردهای اینترنت در کنترل سازمان IAB است؛ که دارای زیر گروه‌های زیادی است که معروفترین آنها IETF و IRTF نام دارند.

* مؤسسه‌ی Internic نیز متولی آدرس‌ها و Domainها است.

* همانگونه که در قبل گفته شده وظیفه‌ی لایه‌ی Inter Network، بسته بندی - آدرس دهی - مسیریابی - کنترل ازدحام است.

پس سرآیند IP باید دارای اطلاعاتی باشد که وظایف بالا را شامل گردد.

سرآیند اینترنت پروتکل نسخه ۴ (Header IP v4)



Version

این فیلد چهار بیت است و نسخه پروتکل IP را مشخص می‌کند. پروتکلی که هم اکنون در اینترنت از آن استفاده می‌شود پروتکل نسخه چهارم می‌باشد.

HL (IP Header Length)

این فیلد نیز ۴ بیتی است و طول Header بسته را مشخص می‌کند، اگر عدد موجود در این فیلد در ۴ ضرب شود طول Header به بایت بدست می‌آید. به عنوان مثال اگر در این فیلد عدد ۱۰ قرار گرفته باشد بدین معنی است که طول Header،



40 بایت خواهد بود. حداقل طول Header (در هنگامی که option برابر صفر باشد) برابر ۲۰ بایت و بنابراین حداقل IHL عدد ۵ می‌باشد. اگر در بسته ای IHL کمتر از ۵ باشد از این بسته صرف نظر می‌شود. حداکثر این مقدار نیز برابر عدد ۱۵ است. بنابراین حداکثر طول Header می‌تواند ۶۰ بایت باشد و در نتیجه قسمت option می‌تواند بین صفر تا ۴۰ بایت تغییر کند.

Type of Service

در این قسمت اطلاعات مربوط به اولویت بندی و کیفیت سرویس ذخیره می‌شود. به عبارت دیگر نوع خدماتی که به بسته تعلق می‌گیرد است که ۸ بیت است.

Total Length

طول یک بسته شامل قسمت Header و data را مشخص می‌کند. باتوجه به تعداد بیت‌های Total Length می‌توان گفت که ماکزیمم طول بسته IP، 64 کیلوبایت می‌باشد و حداقل طول آن طول ثابت Header یعنی 20 بایت است. * در شبکه‌های امروزی بیشتر از ۶۴ کیلوبایت نیز ارسال می‌گردد که واحد آن را در Option تغییر می‌دهند.

Identification

در این قسمت مشخص می‌شود که اطلاعات موجود در این قسمت داده در این بسته IP مربوط به چه دیتاگرامی از لایه بالاتر می‌باشد.

fragment offset

این فیلد در سه بخش سازماندهی شده است:

(الف) ¹DF: با یک شدن این بیت در یک بسته IP هیچ مسیریابی حق ندارد در بین راه این بسته را به بسته های کوچک تر تقسیم نماید. اگر بسته بزرگ باشد توسط روتر از بین می‌رود و به فرستنده اطلاع داده می‌شود.

(ب) ²MF: این بیت مشخص می‌کند که آیا بسته IP آخرین قسمت مربوط به یک دیتاگرام می‌باشد و یا هنوز هم بسته های دیگری وجود دارد. اگر یک باشد یعنی اینکه بسته ادامه دارد و اگر صفر باشد یعنی بسته پایان پذیرفته است.

(ج) **Fragment Offset**: این قسمت ۱۳ بیتی است و در حقیقت شماره ترتیب داده های هر بسته در دیتاگرام شکسته شده می‌باشد. بنابراین یک دیتاگرام می‌تواند به حدود ۸۰۰۰ بسته تقسیم شود.

به عبارت دیگر شماره ترتیب قرار گیری بسته‌های شکسته شده است. اعداد قرار گرفته در این فیلد فاصله بسته‌ها از یکدیگر است و نه عدد ترتیبی دلیل این کار این است که ممکن است در میانه‌ی راه باز بسته‌ی ارسالی شکسته شود که در این حالت نمی‌توان عدد ترتیبی در نظر گرفت و بهترین حالت همان فاصله هر بسته از مبدا است.

TTL (Time To Live)

TTL در نقش یک شمارنده طول عمر بسته را تعیین می‌نماید. طول عمر بسته به زمانی اشاره می‌کند که یک بسته IP می‌تواند در شبکه سرگردان باشد. بیشترین عددی که می‌توان در این قسمت قرارداد عدد ۲۵۵ (یک بایت) است. این عدد توسط فرستنده بسته تنظیم شده و با عبور از هر مسیریاب «هر مرحله عبور از مسیریاب را یک hop یا پرش می‌نامند» یک

¹ Don't Fragment

² More Fragment



واحد از آن کم می‌شود. به ازای هر ثانیه انتظار در صف نیز یک واحد از آن کم می‌شود. وقتی این عدد به صفر برسد بسته IP از مسیر حذف شده و از رسیدن آن به مقصد جلوگیری می‌شود. عددی که به طور معمول توسط سیستم عامل در این قسمت قرار می‌گیرد عدد ۳۰ است و عددی که معمولاً بوسیله آن می‌توان از نقطه ای به نقطه دیگر حرکت کرد عدد ۱۵ است.

Protocol

این فیلد مشخص می‌کند که پروتکل تحویلی از لایه بالاتر TCP یا UDP می‌باشد؛ هر پروتکل دارای یک شماره خاص است.

Header Check Sum

برای کشف خطا بکار می‌رود. این فیلد به دلیل اینکه برخی از اطلاعات بسته در عبور از هر مسیریاب تغییر می‌نماید باید دوباره مقداردهی شود. برای سرعت بخشیدن به عمل مقایسه درست رسیدن بسته، عدد در فرستنده مکمل ۱ شده و در گیرنده با خودش جمع می‌شود؛ اگر نتیجه‌ی جمع صفر شد یعنی بسته درست رسیده است.

Destination و Source

آدرسهای ۴ بایتی منحصر به فرد بر روی اینترنت می‌باشند که مبدا و مقصد را مشخص می‌کنند.

Option

این قسمت اختیاری است و معمولاً اطلاعاتی در خصوص مسیریابی و مسیرهای بهینه در آن قرار می‌گیرد که مورد استفاده مسیریاب‌ها است.

یکی از کارهای آن Source Routing است؛ یعنی آنکه مشخص کنیم که از کدام روتر ارسال گردد و از آنجا که هر روتر دارای آدرسی ۴ بایتی (۳۲ بیتی) است و قسمت Option حداکثر ۴۰ بایت است با تقسیم عدد ۴۰ بر ۴ نتیجه می‌گیریم که تا ۱۰ روتر (گام) را می‌توانیم مشخص کنیم.

Pay Load

در این قسمت داده‌ها و یا در واقع قطعه ای از دیتاگرام لایه بالاتر قرار می‌گیرد.

* **MTU (Maximum Transmit Unit):** بزرگترین واحدی که یک لینک (روتر) می‌تواند داشته باشد.

* **Path MTU:** حداکثر طول مسیر که از مجموع همه لینک‌ها است. (به عبارت دیگر بزرگترین واحدی که در مسیر وجود دارد است). و حداقل آن کوچکترین MTU از کل لینک‌ها است.

* برای بدست آوردن Path MTU یک بسته با DF=1 ارسال می‌گردد، اگر بسته به مقصد رسید، یک بسته‌ی بزرگتر دیگر ارسال می‌شود آنقدر این عمل انجام می‌شود تا بسته‌ی ارسالی دیگر نتواند از لینک‌ها عبور کرده و به مقصد برسد.



* IP نسخه چهار ۳۲ بیتی است که توانایی تولید آدرس را 2^{32} دارد، که به عبارتی 2^{30} معادل ۲ گیگابایت و 2^2 را که ۴ است، که در نهایت به صورت تقریبی ۴ گیگابایت آدرس تولید می‌شود.

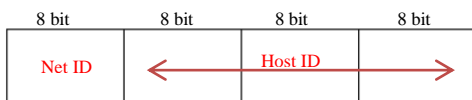
* IP نسخه شش، ۱۲۸ بیتی است که با تقسیم 2^{128} بر 2^{32} عددی معادل 2^{96} بدست می‌آید که چندین برابر نسخه چهار آن است.

کلاس IP نسخه چهار

آی‌پی نسخه چهار به صورت سلسه مراتبی (غیر Flat) است که باعث مدیریت ساده‌تر آن می‌شود. این IP دارای Net ID که سمت چپ‌ترین است و Host ID که در سمت راست قرار می‌گیرد است قسمت هاست آی‌دی اعداد مربوط به ایستگاه‌ها (کامپیوترها) است و قسمت اول مربوط به شبکه است.

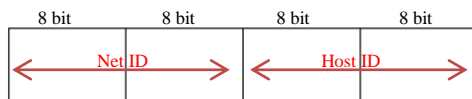
یادآور می‌شود که IP نسخه چهار دارای ۳۲ بیت (۴ بایت) است که توسط نقطه به چهار قسمت ۸ بیتی تقسیم می‌شود. کلاس آی‌پی در این نسخه به صورت زیر است.

کلاس A



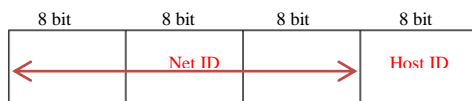
رنج این کلاس 0 - 127 است. (قسمت Net ID)

کلاس B



رنج این کلاس 128 - 191 است. (قسمت Net ID)

کلاس C



رنج این کلاس در قسمت Net ID اعداد؛ 192 - 223 است.

کلاس D

این کلاس دارای قسمت Net و Host نیست و برای گروهی از کامپیوترها (چند بخشی - Multi Cast) استفاده می‌شود. رنج این کلاس به صورت 224 - 239 است.

کلاس E

این کلاس همانند کلاس بالا دارای قسمت Net و Host نمی‌باشد و رزور شده است، (به صورت عمومی استفاده نمی‌شود) و دارای رنج 240 - 255 است.

* IP در حقیقت اعداد باینری هستند که برای راحتی، آنها را در مبنای ده می‌نویسیم؛

10001110 . 10001110 . 10001111 . 11100011



* آدرس شبکه برای یک آی پی به صورت 0 Net ID . است؛ برای مثال آدرس شبکه برای آی پی ۱۲۹.۱۶۸.۱۸.۹۲ برابر با ۱۹۲.۱۶۸.۱۸.۰ خواهد بود.

تعداد کامپیوتر	تعداد شبکه	کلاس ها
$2^{24} - 2$	$2^7 - 2$	A
$2^{16} - 2$	2^{14}	B
$2^8 - 2$	2^{21}	C

کم کردن ۲ بابت آدرس شبکه و Broadcast است. به عبارتی دیگر دو آی پی 0 و 255 رزرو شده هستند. (هم برای Net و هم برای Host).

* آدرسی که به صورت *.*.*.127 باشد به آن Loop Back می گوئیم که برای تست نرم افزارهای شبکه بکار می رود. در این IP اطلاعات تحویل لایه ی زیر Network نمی شود و از همان لایه ی Network به صورت فرضی ارسال می گردد.

* اگر روی بسته ای 192.168.18.255 باشد؛ به آن Broadcast می گوئیم، یعنی آنکه به تمام کامپیوترهای درون شبکه ارسال شود.

* اگر 255.255.255.255 را در نظر بگیریم باز عمل بالا انجام می شود چرا که روتر آی پی های Broadcast را دور می ریزند پس این مدل فقط در شبکه درونی ارسال می گردد. از این مدل IP وقتی استفاده می کنیم که کلاس IP شبکه را ندانیم و بخواهیم به کامپیوتر جاری IP اختصاص دهیم.

* اگر آدرس فرستنده 0.0.0.0 باشد، یعنی بازگشت به خود سیستم صورت بگیرد، چرا که گیرنده نمی تواند چنین آدرسی داشته باشد.

وقتی آدرس فرستنده را ۴ صفر قرار می گیرند که فرستنده IP خودش را نداند در نتیجه یک بسته سراسری ارسال (Broadcast) ارسال کرده و IP فرستنده را ۴ صفر قرار داده تا در مرحله ی بعد یک آی پی به آن اختصاص یابد.

آدرس های شخصی (Private IP)

در شبکه های داخلی استفاده می شوند و از استانداردهای زیر تبعیت می کنند.

10.*.*.* 172.16.*.* - 172.31.*.* 192.168.*.*

* اگر در دو کامپیوتر یا بیشتر از دو رنج IP متفاوت استفاده کنیم برای برقراری ارتباط حتماً نیاز به یک روتر (مسیریاب) است. مانند؛

192.168.1.1 -- 172.16.31.5

* اگر بسته ای با یکی از IP های شخصی وارد اینترنت شود در اولین روتر از بین می رود.

در فوق آدرس دهی با توجه به کلاس را بررسی کردیم؛ آنچه در ادامه می آید آدرس دهی بدون کلاس است.

اولین روش آدرسی دهی بدون کلاس استفاده از NAT است که یک آدرس معتبر را به یک شبکه اختصاص می دهند که از طریق Router به شبکه ی داخلی وصل است.



روش دیگر استفاده از Subnet Mask است؛ که در آن مرز بین Net ID و Host ID از پیش تعیین نشده است و یک عدد ۳۲ بیتی (۴ بیتی) دیگر که آنرا Subnet Mask می‌نامیم تعداد بیت‌های Net ID را مشخص می‌کند. تعداد بیت‌های یک در Subnet Mask، تعداد بیت‌های Net ID در آدرس IP را مشخص می‌کند.

سابتنت به دو صورت نوشته می‌شود؛

اول اینکه بعد از IP اصلی یک اسلش (/) قرار داده و تعداد بیت‌های یک را مشخص می‌کنیم؛ مانند: $255.255.255.0/24$.

دوم آنکه به صورت یک IP عادی نوشته می‌شود؛ مانند: $255.255.255.0$

برخی مواقع سابتنت به صورت بالا (۲۵۵) نیست و عدد دیگری ممکن است باشد، برای به دست آوردن آدرس شبکه در این حالت IP را با سابتنت AND می‌کنیم.

برای مثال آدرس شبکه برای آی‌پی $131.18.20.12$ با سابتنت $255.255.192.0$ را به صورت زیر به دست می‌آوریم؛

ابتدا برای راحتی کار از خصوصیت AND استفاده می‌کنیم:

$$x \text{ AND } 1 = x \quad x \text{ AND } 0 = 0$$

پس: از آنجا که عدد ۲۵۵ در مبنای ۲ معادل 11111111 است پس AND هر عددی با آن خود همان عدد می‌شود.

$$\begin{array}{r} 131 . 18 . \underline{20} . 12 \\ 255 . 255 . \underline{192} . 0 \\ \hline 121 . 18 . 0 . 0 \end{array}$$

$$20 \rightarrow 00010100$$

$$192 \rightarrow 11000000$$

$$\text{AND} \rightarrow 00000000$$

مثال) آی‌پی $191.18.160.20$ را در نظر بگیرید؛

الف) آدرس شبکه در صورت استفاده از روش آدرس‌دهی مبتنی بر کلاس:

از آنجا که IP فوق در کلاس B قرار دارد پس آدرس شبکه به صورت $191.18.0.0$ خواهد بود.

ب) آدرس شبکه در صورتی که Subnet Mask برابر با $255.255.192.0$ باشد:

با توجه به اینکه دو بایت اول ۲۵۵ هستند و با استفاده از خاصیت AND این دو بایت خود اعداد IP خواهند بود برای بایت سوم نیز عدد ۱۹۲ و ۱۶۰ را باهم AND کرده و پاسخ را می‌نویسیم و در نهایت برای بایت چهارم از خاصیت AND صفر با هر عدد صفر است استفاده می‌کنیم.

$$160 \rightarrow 10100000$$

$$192 \rightarrow 11000000$$

$$\text{AND} \rightarrow 10000000$$

در نهایت آدرس شبکه به صورت $191.18.128.0$ خواهد بود.

ج) آدرس Broadcast با استفاده از آدرس‌دهی مبتنی بر کلاس:

$$191.18.255.255$$



(د) آدر Broadcast اگر $255.255.192.0$ Subnet Mask:

با توجه به اینکه حالت باینری سابنت به صورت $00000000 . 11000000 . 11111111 . 11111111$ است، پس Subnet ۱۸ بیت است برای بدست آوردن Broadcast، ۱۸ بیت اول آی پی داده شده را نگه داشته و مابقی را یک می کنیم. پس دو بیت پر ارزش عدد ۱۶۰ را در IP دست نمی زنیم و بقیه بیت ها را یک می کنیم. (عدد ۱۶۰ در مبنای دو به صورت 10100000 است که دو بیت اولش را به دلیل آنکه دو بیت اول بایت سوم Subnet جزء Net ID هستند دست نزده و مابقی را یک می کنیم (یعنی قسمت Host ID) پس می شود $10111111 . 11111111$ ؛ که عدد به دست آمده را به مبنای ده تبدیل کرده و کل Broadcast IP را می نویسیم که به صورت $191.18.191.255$ خواهد بود.

* Subnet Mask معتبر همیشه تعدادی یک پشت سر هم دارد و مابقی صفر است مانند 1110000 در غیر این صورت سابنت مذکور نا معتبر است.

(مثال) کدام یک از Subnet های زیر معتبر و کدام نامعتبر است؟

$255 . 255 . 0 . 0$

معتبر است چرا که اگر آنرا در مبنای دو بنویسیم تمام یک ها به ترتیب پشت سر هم هستند.

$255 . 0 . 255 . 0$

نامعتبر است چرا که بین یک ها صفر خواهد بود.

$255 . 160 . 0 . 0$

نامعتبر است چرا که عدد ۱۶۰ در مبنای دو به صورت 10100000 خواهد بود که یک ها پیوسته نیستند.

$255 . 255 . 255 . 255$

معتبر است

$0 . 0 . 0 . 0$

معتبر است.

Subnet Mask پیش فرض برای کلاسها به صورت زیر است؛

کلاس A: $255.0.0.0$

کلاس B: $255.255.0.0$

کلاس C: $255.255.255.0$



Sub netting (زیر شبکه سازی)

تبدیل یک شبکه‌ی بزرگ به شبکه‌های کوچکتر را گویند.

زیر شبکه‌سازی برای فروش راحت‌تر و دسته‌بندی IP‌های Valid است که به ISPها اختصاص داده می‌شود. دلیل دیگر این است که شبکه‌ها را کوچکتر کرده تا برای Broadcast کردن فقط در زیر شبکه خودشان کار کنند و نه در کل شبکه.

زیر شبکه‌سازی با استفاده از Subnet Mask

تعداد بیت‌های لازم برای زیر شبکه‌سازی = $\lceil \log_2 \text{تعداد زیر شبکه} \rceil$

برای مثال برای IP زیر و شبکه‌ای با ۴ زیر شبکه می‌خواهیم عملیات تقسیم بندی را انجام دهیم؛
IP= 170.20.0.0

از آنجا که برای آی‌پی فوق سابنت بیان نشده سابنت آنرا بر اساس کلاس و به صورت پیش‌فرض در نظر می‌گیریم؛
255.255.0.0 که معادل ۱۶ بیت اول برابر با یک می‌شود.

الف) Subnet Mask جدید برای زیر شبکه‌ها را بنویسید:

$$\log_2 4 = 2 \text{ bit}$$

اکنون Subnet اصلی (پیش‌فرض) را با بیت‌های جدید به دست آمده جمع می‌کنیم؛ کار فوق یعنی برای ۴ زیر شبکه ۲ بیت متغییر لازم است و مابقی بیت‌های باقی‌مانده برای Host استفاده خواهند شد.

بیت اول یک خواهند شد $16 + 2 = 18$

$$11111111 . 11111111 . 11000000 . 00000000 \rightarrow 255.255.192.0$$

ب) آدرس هر ۴ زیر شبکه را بدست آورید:

برای اینکار آی‌پی داده شده را نوشته و بر اساس بیت‌های به دست آمده (۲) حالت‌های ممکن را می‌نویسیم مابقی بیت‌ها در ادامه صفر خواهند شد (Host ID) و ۱۶ بیت ابتدایی دست نخواهند خورد.

	170	.	20	.	0	.	0
170	.	20	.	00	000000	.	00000000
170	.	20	.	01	000000	.	00000000
170	.	20	.	10	000000	.	00000000
170	.	20	.	11	000000	.	00000000

که به ترتیب در مبنای ده به صورت زیر خواهند بود:

170.20.0.0
170.20.64.0
170.20.128.0
170.20.192.0

ج) IP= 170.20.88.22 در کدام زیر شبکه قرار دارد:

برای اینکار یا به آدرس‌های شبکه بالا نگاه کرده و مشخص می‌کنیم که در کدام رنج قرار دارد یا اینکه آنرا با Subnet Mask به دست آمده AND کرده و جواب را می‌نویسیم. که جواب نهایی به صورت 180.20.64.0 خواهد بود.



(د) Broadcast هر زیر شبکه را بنویسید:

برای اینکار می توان مانند قبل ۱۸ بیت اول را که قبلاً به عنوان Net ID به دست آورده ایم ثابت نگه داشته و مابقی بیت ها را یک کنیم؛ یا اینکه به آدرس شبکه ها نگاه کرده و از زیر شبکه ی بعدی یکی کم کرده و آنرا به عنوان Broadcast شبکه ی جاری در نظر بگیریم، برای آخرین زیر شبکه نیز به الگوی صعودی نگاه می کنیم و آنرا بدست می آوریم.

---- . ---- . 00 1111111 . 111111111 → 170.20.63.255
 ---- . ---- . 01 1111111 . 111111111 → 170.20.127.255
 ---- . ---- . 10 1111111 . 111111111 → 170.20.191.255
 ---- . ---- . 11 1111111 . 111111111 → 170.20.255.255

(ه) تعداد کامپیوترها در هر زیر شبکه را بنویسید:

برای به دست آوردن تعداد کامپیوتر در هر زیر شبکه تعداد Host ID را بدست آورده و دو را بتوان آن رسانده و در نهایت دو عدد بابت آدرس شبکه و Broadcast از آن کم می کنیم که در این مثال به صورت زیر خواهد بود.

$$32-18=14 \rightarrow 2^{14} - 2 \text{ Computer In Each Sub net}$$

(مثال) آدرس شبکه 172.20.64.0/18 را در نظر بگیرید؛

(الف) Subnet Mask ی را بدست آورید که شبکه فوق را به ۴ زیر شبکه تقسیم کند:

$$\log_2 4 = 2 \text{ bit} \quad 18+2=20 \text{ bit For Net ID} \quad 11111111.11111111.11110000.0000 \rightarrow 255.255.240.0$$

(ب) آدرس زیر شبکه ها:

همانند مثال قبل تا بیت ۱۸ که در مسئله بیان شده دست نمی زنیم و همان اعداد آی پی اصلی را می نویسیم، بیت ۱۹ و ۲۰ را که زیر شبکه ها هستند تمام حالت های شان را می نویسیم.

172.20.01 00 0000. 00000000 → 172.20.64.0
 172.20.01 01 0000. 00000000 → 172.20.80.0
 172.20.01 10 0000. 00000000 → 172.20.96.0
 172.20.01 11 0000. 00000000 → 172.20.112.0

(ج) آدرس Broadcast در زیر شبکه ها را بنویسید:

172.20.79.255
 172.20.95.255
 172.20.111.255
 172.20.127.255

(د) آدرس 172.20.91.18 در کدام زیر شبکه قرار دارد؟

172.20.80.18



ه) اگر کامپیوتری با آدرس 172.20.86.18 بخواهد به تمام اعضای زیر شبکه‌ی خود داده ارسال کند در آدرس مقصد چه آدرسی باید قرار دهد؟

برای پاسخ به سؤال فوق آدرس زیر شبکه را پیدا کرده و Broadcast آن را می‌نویسیم:

172.20.80.0 → Broadcast= 172.20.95.255

مثال) آدرس شبکه‌ای به صورت 190.2.160.0/20 است. مطلوب است؛

الف) می‌خواهیم شبکه‌ی فوق را طوری تقسیم‌بندی کنیم که هر زیر شبکه ظرفیت ۶۰۰ میزبان را داشته باشد: همانگونه که می‌دانیم برای نمایش عدد ۶۰۰ در مبنای دو به ۱۰ بیت (حالت) نیاز است. پس ۱۰ بیت سمت راست ۳۲ بیت آی‌پی را برای Host در نظر می‌گیریم.

ب) Subnet Mask جدید برای زیر شبکه‌سازی را بنویسید:

برای اینکار ۱۰ بیت سمت راست را صفر و مابقی بیتها را یک می‌کنیم؛

11111111 . 11111111 . 111111 00.00000000 → 255.255.252.0

ج) تعداد زیر شبکه‌ها را بیابید:

زیر شبکه خواهیم داشت $2^2=4$ ، بیت برای زیر شبکه‌ها $32-10-20=2$

۱۰ بیت بابت (Host) - ۲۰ بیت نیز در خود مسئله به عنوان Subnet بیان شده است.

د) آدرس زیر شبکه‌ها را بنویسید:

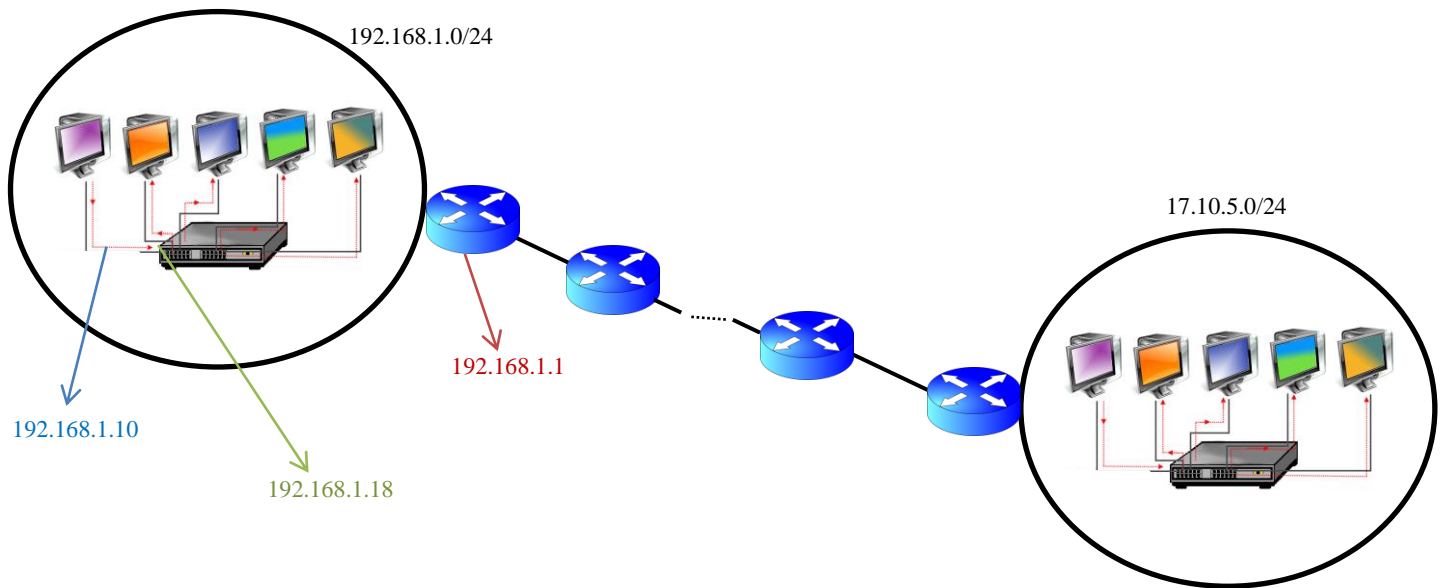
همانطور که در مسئله بیان شده ۲۰ بیت اول (سابنت) Net ID هستند و ۲ بیت نیز بابت زیر شبکه‌ها خواهد بود، پس بیت

۲۱ و ۲۲ را برای تمام حالت تغییر داده و Host ID را مانند قبل صفر قرار می‌دهیم؛

190 . 2 . 1010 00 00 . 00000000	→	190.2.160.0
190 . 2 . 1010 01 00 . 00000000	→	190.2.164.0
190 . 2 . 1010 10 00 . 00000000	→	190.2.168.0
190 . 2 . 1010 11 00 . 00000000	→	190.2.172.0

ه) آدرس Broadcast در هر زیر شبکه را بنویسید:

192.2.163.255
192.2.167.255
192.2.171.255
192.2.175.255



* هنگامی که در شبکه‌ی داخلی مبادله صورت گیرد، با آدرس MAC (از لایه‌ی ۲) کار می‌کند اما اگر خارج از شبکه بود IPها مقایسه می‌شود و بسته‌ی ارسالی را به روتر می‌فرستند، چرا که در شبکه‌ی داخلی آن آی‌پی وجود ندارد. در شبکه داخلی با پروتکل ARP آدرس MAC کارت شبکه را پیدا می‌کند. این پروتکل بوسیله‌ی Broadcast کردن آدرس مقصد را پیدا کرده و آدرس MAC و IP را روی بسته قرار می‌دهد و می‌فرستد. در خارج از شبکه آدرس MAC روتر روی بسته‌ها اضافه می‌شود و ادامه کار را انجام می‌دهد. در صورتی که شبکه خارجی بود آدرس شبکه خود با آدرس شبکه مقصد مقایسه می‌شود و در صورت نابرابری بسته را برای روتر ارسال می‌کند.

مثال) کدامیک از بسته‌های زیر اگر وارد شبکه اینترنت شود به مقصد خواهد رسید؟

- ۱) 192.168.18.2 به مقصد نمی‌رسد چرا که جز آی‌پی‌های Private است.
- ۲) 172.32.18.12 به مقصد می‌رسد.
- ۳) 127.18.12.20 به مقصد نمی‌رسد به دلیل اینکه آی‌پی Loop Back است.
- ۴) 0.0.0.0 به مقصد نمی‌رسد چرا که آدرس خود کامپیوتر برای شناسایی اولیه است.
- ۵) 255.255.255.255 به مقصد نمی‌رسد زیرا روترها آدرس Broadcast را از خود عبور نمی‌دهند.
- ۶) 192.186.12.18 به مقصد می‌رسد. IP معتبر است.

دستورات:

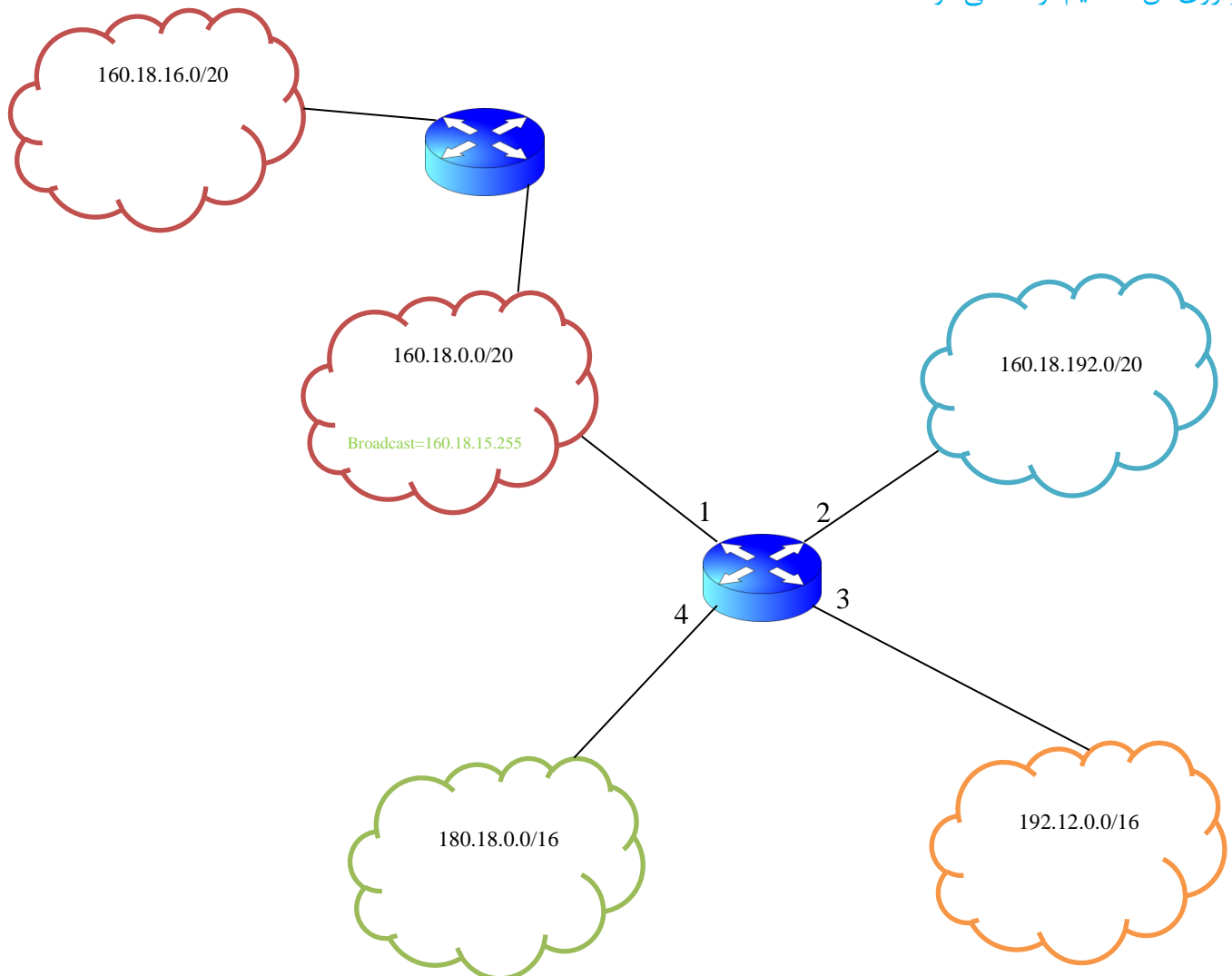
Ping آدرس مقصد (Net Bios Name)

ARP -Print



Super netting (ادغام زیر شبکه‌ها)

این عمل بلعکس Sub netting است. در ادامه و با ذکر مثال برای شبکه‌ی زیر یک Subnet تعیین خواهیم کرد که برای تمام زیر شبکه‌ها در هنگام AND کردن پاسخ یکسان دهد. Super netting در جداول مسیریابی روترها کاربرد دارد. جدول تصمیم‌گیری ارسال بسته از روتر اینترنت به روتر شبکه‌ی داخلی را Forwarding Table گویند که برای ارسال بسته‌ها از روی آن تصمیم گرفته می‌شود.



* شبکه‌های Stub فقط اطلاعات داخلی خود را رد و بدل می‌کنند مانند کوچه‌ی بن بست و شبکه‌های Transit علاوه بر داده‌های خود، داده‌های شبکه‌های دیگر را نیز انتقال می‌دهند.

قبل از شروع به Super netting ابتدا موارد زیر را در نظر می‌گیریم؛

برای شبکه‌ی فوق یک جدول Forward وجود دارد که به شکل زیر است:

* فیلد اول آدرس شبکه را نگه می‌دارد.

آدرس شبکه	Subnet Mask	Next Hop	Port	Metric



* فیلد دوم سابنت شبکه (ادغام شده یا نشده)

* Next Hop: گام بعدی (مسیریاب) در ارسال به مقصد.

* Port: شماره‌ی هر درگاه روتر است. (مانند کارت شبکه‌ها در کامپیوتر)

* Metric: در هنگام جستجو مسیر برای روتر بعدی ممکن است چند مسیر وجود داشته باشد که کمترین مسیر در اینجا قرار می‌گیرد؛ یا اینکه ممکن است هزینه‌های Metric گام‌ها باشد، یا پهنای باند و یا هزینه‌ی واقعی و ...

و اما کاربرد Super netting در مثال فوق: همانگونه که مشاهده می‌شود Port شماره ۱ روتر خود دارای دو زیر شبکه است و برای جستجوی راحت‌تر و مناسب‌تر در جدول فقط یک زیر شبکه ثبت می‌شود فلذا برای اینکار باید Subnet Mask را در نظر بگیریم که در صورتی که هر IP از هر زیر شبکه را با آن AND کنیم پاسخ آدرس شبکه یکی باشد و در نهایت آن آدرس شبکه و Subnet را به جای آن دو زیر شبکه در جدول ثبت خواهیم کرد. دقت شود همیشه باید بزرگترین Subnet در نظر گرفته شود.

برای پیدا کردن این Subnet آدرس دو زیر شبکه را یادداشت کرده و بیت‌های مشابه را نگه داشته و بقیه‌ی بیت‌ها را صفر خواهیم کرد که در مثال فوق برای پورت یک به صورت زیر خواهد شد؛

160 . 18 . 0 . 0

160 . 18 . 16 . 0

در دو آدرس بالا ۱۶ بیت اول مطمئناً مشابه هستند پس تا اینجا ۱۶ بیت اول برای سابنت شدن یک خواهند بود. در بایت سوم (۸ بیت سوم) دو عدد را در مبنای دو می‌نویسیم:

0 → 00000000

16 → 00010000

در بالا نیز سه بیت اول یکسان است پس به ۱۶ بیت قبلی ۳ بیت دیگر اضافه خواهد شد که به عبارتی ۱۹ بیت اول یک خواهند شد و مابقی بیت‌ها صفر که در نهایت جواب پایانی برای Subnet Mask مشترک برای زیر شبکه‌ها به صورت زیر است:

11111111.11111111.111 00000.00000000 → 255.255.224.0

این سابنت در صورت AND شدن با هر کدام از زیر شبکه‌های مورد مناقشه پاسخ یکسان خواهد داد که آنرا آدرس شبکه می‌نامیم و به صورت زیر است؛

IP=	160.18.16.0	→	160.18. 00010000.00000000
Subnet Mask=	255.255.224.0	→	255.255.11100000.00000000
AND=	160.18.0.0	→	160.18.0.0

یا

IP=	160.18.0.0	→	160.18. 00000000.00000000
Subnet Mask=	255.255.224.0	→	255.255.11100000.00000000
AND=	160.18.0.0	→	160.18.0.0

در جدول Forward آدرس شبکه فوق با سابنت ۱۹ ثبت خواهد شد، پس جدول Forward به شکل زیر تکمیل می‌شود:

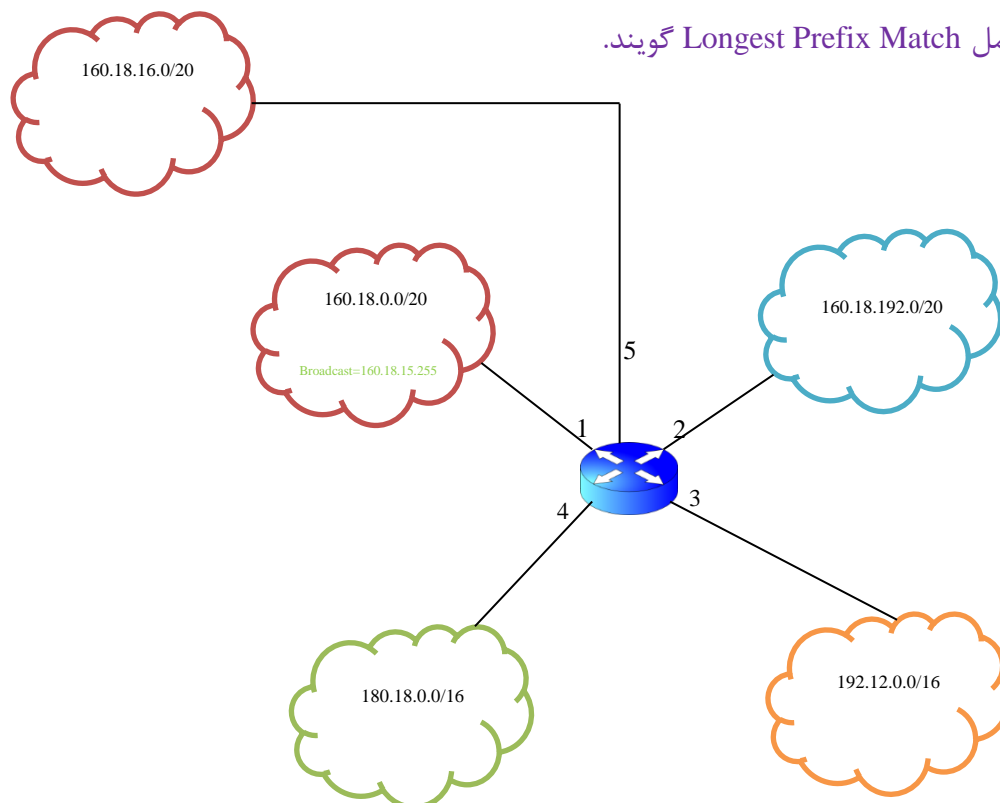


آدرس شبکه	Subnet Mask	Next Hop	Port	Metric
160.18.0.0	/19		1	
160.18.0.0	/20		2	
192.12.0.0	/16		3	
180.18.0.0	/16		4	
0.0.0.0	/0			

Default Gateway

کار مسیریاب را انجام می‌دهد، به این صورت که اگر میزبان بخواهد به یک میزبان دیگر داده ارسال کند به طوری که فرستنده و گیرنده در دو شبکه‌ی مجزا قرار داشته باشند (Net ID متفاوت داشته باشند)، فرستنده دادخ را به دروازه‌ی پیش فرض (Default Gateway) ارسال می‌کند. (یعنی آدرس MAC دروازه‌ی پیش فرض را روی فریم ارسالی خود قرار داده ولی آدرس IP مقصد نهایی بر روی بسته درج می‌شود) و دروازه‌ی پیش فرض آن را به سمت گیرنده هدایت می‌کند. آدرس IP گیرنده در کل مسیر ثابت است ولی آدرس MAC گیرنده، گام به گام تغییر می‌کند. در جدول فوق دروازه‌ی پیش فرض به صورت 0.0.0.0 ثبت شده است.

(مثال) بسته‌ای با IP مقصد 160.18.18.12 به روتر ارسال می‌شود که با تک تک IP شبکه‌ها (Subnet) موجود در شبکه AND می‌شود و پس از پیدا شده آدرس شبکه در صورتی که دو آدرس یکسان پیدا شد، آنرا به آدرس شبکه‌ای می‌فرستند که Subnet بزرگتری داشته باشد. اگر با فیلد آخر جدول (D.G)، AND شد و پاسخ یکی شد باز آنرا به بزرگترین Subnet ارسال می‌کند؛ به این عمل Longest Prefix Match گویند.





به شکل بالا توجه کنید، با تغییر در زیر شبکه و تبدیل آن به پورت ۵ اکنون باید آی پی و سابنت آن به جدول Forward نیز اضافه گردد، اکنون اگر مثال قبل را دوباره بررسی کنیم چون این آدرس شبکه بزرگتر است (از نظر شماره پورت)، بسته به آن ارسال می گردد.

پس برای تعریف نهایی Super netting می توان گفت که با استفاده از کوچک کردن Subnet، چندین شبکه را باهم ادغام کرده تا شبکه‌ها دارای آدرس واحدی گردند.

مثال) چهار زیر شبکه زیر را در نظر بگیرید، Subnet Mask ی را پیدا کنید که آنها را به شبکه‌ی واحدی تبدیل کند:

102.8.20.1

102.18.30.1

102.65.18.7

102.120.12.8

برای راحت تر شدن کار بزرگترین عدد متغییر و کوچکترین را در نظر گرفته و پس از تبدیل به مبنای دو آنها را باهم مقایسه می کنیم؛ (هشت بیت اول ثابت است، از ۸ بیت دوم کوچکترین ۸ و بزرگترین ۱۲۰ است)

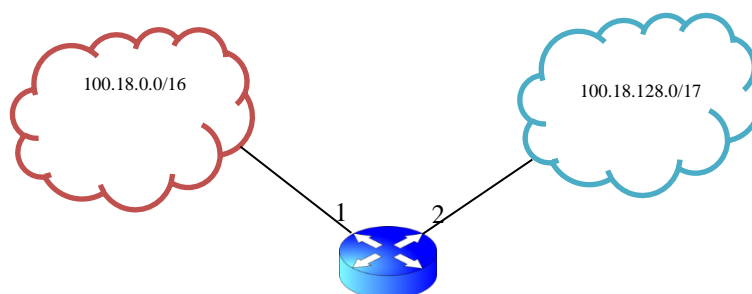
8 → 00001000

120 → 01111000

همانگونه که مشاهده می شود در این دو عدد فقط یک بیت دارای تشابه است، در نتیجه هشت بیت اول بعلاوه‌ی یک بیت از بایت دوم که جمعاً ۹ بیت خواهند بود را یک کرده (برای سابنت) و مابقی را صفر در نظر می گیریم:

11111111 . 1 0000000 . 00000000 . 00000000 → 255.128.0.0

مثال) در شکل زیر نیز اگر بسته‌ای به روتر برسد و آدرس شبکه پس از مقایسه یکسان گردد، بسته به شبکه‌ای ارسال می گردد که Subnet بزرگتری داشته باشد.



دستورات

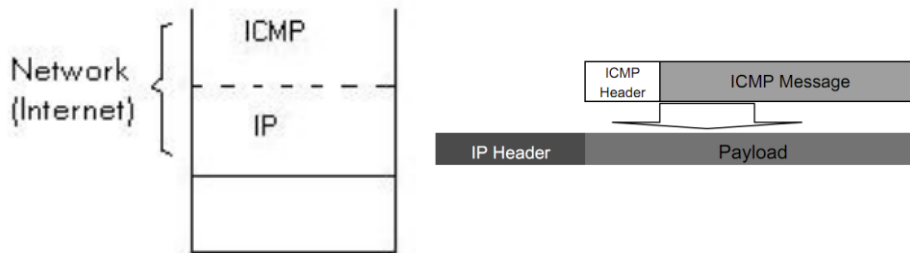
دستور زیر جدول مسیریابی را نشان می دهد:

Rout-Print



پروتکل ICMP (Internet Control Message Protocol)

این پروتکل کار گزارش خطا را انجام می‌دهد. در حقیقت پروتکل مدیریتی برای IP است و بر روی IP قرار می‌گیرد. در حقیقت مانع از ادامه یافتن خطا می‌شود اما خطا را تصحیح نمی‌کند، شکل آنرا در زیر می‌توانید مشاهده کنید:



ICMP Header

این پروتکل نیز همانند سایر پروتکل‌ها دارای یک سرآیند است که اطلاعات زیر را در خود نگه‌داری می‌کند:

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
Type		Code		Checksum																											
Parameters																															
Data																															

Type: نوع پیغام در آن قرار می‌گیرد و ساختار فیلدهای Parameters و Data بسته به عددی که در این فیلد قرار می‌گیرد متفاوت خواهد بود.

Code: کد زیر گروه پیغام را مشخص می‌کند.

Checksum: کنترل خطا (مانند سایر سرآیندها)

Parameters: هر دستور ممکن است پارامتری لازم داشته باشد مانند تأخیر در ارسال بین دو بسته

Data: محتویات پیام خواهد بود.

برخی از پیغام‌های ICMP

۱) Destination Unreachable: زمانی صادر می‌شود که گیرنده در دسترس نباشد. (وقتی یک بسته به هر دلیلی به مقصد

نرسد، بسته از بین می‌رود و پیغام ICMP با این نوع (Type) به فرستنده‌ی پیغام ارسال می‌شود)

خود این پیغام زیرگروه‌هایی دارد؛ مانند Timeout

* دستور Ping از ICMP برای انجام عملیات خود استفاده می‌کند.

۲) Time Exceed: این پیام زمانی صادر می‌شود که مهلت قانونی یک بسته منقضی شده باشد، به عبارت دیگر در زمان TTL

تعیین شده به مقصد نرسد، در نتیجه بسته دور ریخته شده و به مبدأ پیام فوق صادر می‌شود.



۳) Source Quench: این بسته زمانی برای ماشین فرستنده ارسال می‌شود که از آن خواسته شود حجم ارسال بسته‌هایش را کاهش دهد چرا که در غیر اینصورت ازدحام پیش خواهد آمد. اگر ماشین میزبان پس از طی مدت مشخصی دیگر این پیام را دریافت نکرد می‌تواند سرعت تولدی بسته‌های خود را به حالت اولیه بازگرداند.

فرستنده در کاهش سرعت مختار است و در صورت عدم انجام این کار پس از پر شدن صف روتر بسته‌های ارسالی بعدی از بین رفته و فرستنده مجبور خواهد بود دوباره آنها را ارسال کند. (برای حملات D.O.S استفاده می‌شود).

۴) Redirect: اگر بسته‌ای به روتری برسد و روتر پس از بررسی متوجه شود که از همان پورت وارد شده باید بازگردد، متوجه مشکل شده؛ پس پیغام فوق را به فرستنده (یا یک مرحله قبل) ارسال می‌شود.

۵) Echo Request & Reply: فرستنده یک پیغام به گیرنده می‌فرستد و گیرنده همان را به فرستنده باز می‌گرداند. (در ping کاربرد دارد؛ برای بررسی رسیدن داده به مقصد. «زمان رفت و برگشت در Ping، RTT نام دارد»).

۶) Time Stamp Request & Reply: همانند پیام بالایی عمل کرده با این تفاوت که گیرنده علاوه بر پاسخ ارسالی مهر زمانی به بسته زده و آنرا بر می‌گرداند. زمان ارسال و دریافت را نمایش می‌دهد. (به دلیل تفاوت زمانی دو سیستم معمولاً محاسبه‌ی درست زمانها به تفکیک سخت است).

* دستور Trace Route باعث می‌شود که تمام Nodeهای بین مبدأ و مقصد استخراج گردند (کل مسیر). در ویندوز از دستور: (آدرس مقصد Tracert) استفاده می‌شود.

این دستور از دو پیغام شماره‌ی ۲ (Time Exceed) و ۵ یا ۶ (Echo Request & Reply OR Time Stamp Request & Reply) در ICMP استفاده می‌کند.

برای اینکار ابتدا یک بسته با شماره ۵ یا ۶ و $TTL=1$ ارسال می‌کند، که در روتر اول پیغام ۲ به مبدأ ارسال می‌شود؛ این عمل و افزایش گام به گام TTLها آنقدر ادامه پیدا خواهد کرد تا به مقصد برسد، در نهایت آدرس تمام روترها (مسیر) را بر می‌گرداند. به این صورت کل Nodeهای مسیر استخراج خواهند شد.

پروتکل ARP (Address Resolution Protocol)

هرگاه بخواهیم آدرس MAC یک کامپیوتر را از روی آدرس IP آن بدست آوریم از این پروتکل استفاده می‌کنیم. برای این کار کامپیوتر فرستنده یک ARP Request تولید کرده و داخل آن پیامی با این مضمون (چه کسی آدرس MAC کامپیوتری با آدرس IP ... را دارد؟) را در شبکه Broadcast می‌نماید. (یعنی آدرس MAC آنرا یک می‌گذارد). تمام کامپیوترهای شبکه این پیام را دریافت کرده و تنها کامپیوتری به آن پاسخ می‌دهد که صاحب آدرس IP مذکور باشد. گیرنده یک پیام ARP Reply تولید می‌کند و آدرس MAC خود را در آن قرار می‌دهد و آنرا به تولید کننده‌ی پیام ARP Request ارسال می‌کند.

اگر آدرس فیزیکی سوال شده در شبکه‌ی محلی جاری نباشد قاعده‌تاً پاسخی به درخواست ARP داده نمی‌شود در این حالت دو راه وجود دارد:



الف) وقتی مسیریابی که به آن شبکه متصل است می‌بیند آدرس مقصدی که توسط ARP سوال شده روی یک شبکه‌ی محلی دیگر واقع است، در پاسخ به آن، آدرس فیزیکی خودش را به ایستگاه ارسال می‌کند، که به این روش Proxy ARP گفته می‌شود.

ب) ایستگاه‌ها خود موظفند که محلی یا خارجی بودن ماشین مقصد را با توجه به الگوی زیر شبکه تشخیص داده و در صورت خارجی بودن، آدرس فیزیکی یک مسیریاب مناسب را انتخاب کنند.

آدرس‌های بدست آمده از طریق پروتکل ARP برای جلوگیری از دوباره کاری در یک جدول با نام ARP Table یا ARP Cache ذخیره می‌شوند که باعث بالا رفتن سرعت این پروتکل می‌شود.

جدول فوق هر دقیق یکبار به‌روز می‌شود.

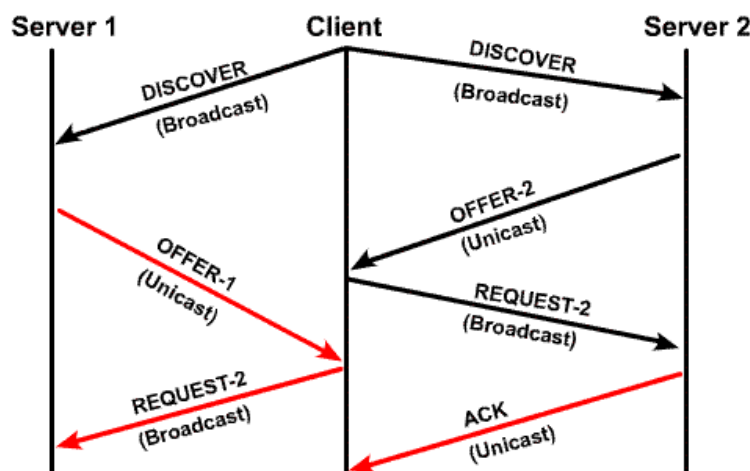
IP	MAC	Expire
192.168.1.18	EF-FB-AB-00-AA	...
192.168.1.31	ED-99-09-33-00-09	...

* آدرس MAC در لایه‌ی پیوند داده‌ها استفاده می‌شود.

پروتکل DHCP (Dynamic Host Configuration Protocol)

آدرس دهی خودکار به سیستم‌های درون شبکه را انجام می‌دهد. این پروتکل یک نسخه‌ی Client و یک نسخه‌ی Server دارد. تمام دستگاه‌های سرویس گیرنده نسخه‌ی کلاینت را دارا هستند؛ نسخه‌ی سرور نرم‌افزاری مانند DHCP Server و ... است که می‌تواند بر روی سرور نصب گردد و به هر سرویس گیرنده که می‌خواهد وارد شبکه شود برای مدت معینی یک IP اجاره می‌دهد.

روال کار این پروتکل به صورت زیر است:



کلاینت ابتدا دو درخواست برای دو سرور مجزای دارای DHCP به صورت Broadcast ارسال می‌کند، سپس سرورها به صورت یک پیام Offer به کلاینت پاسخ آمادگی تحویل IP می‌دهند در اینجا کلاینت یکی را به اختیار انتخاب کرده و از وی درخواست IP نموده که سرور با ارسال ACK، آی‌پی را برای مدت معین اختصاص (اجاره) می‌دهد.



پروتکل DHCP می‌تواند موارد زیر را Set کند؛

IP, Subnet Mask, Default Gateway, DNS Name, DNS Server

* در صورتی که زمان اجاره‌ی IP در حال اتمام باشد ($1/2$ زمان گذشته باشد)، کلاینت مجدداً روال فوق را البته برای تمدید اجاره‌ی IP انجام می‌دهد. اگر توافق بین سرور و کلاینت صورت نگیرد، کلاینت پس از گذشت ($7/8$) زمان کل مجدداً روال فوق را این بار برای درخواست IP جدید انجام داده و درخواست IP جدید می‌کند.

* مدت زمان اجاره‌ی IP در DHCP Server قابل تنظیم است.

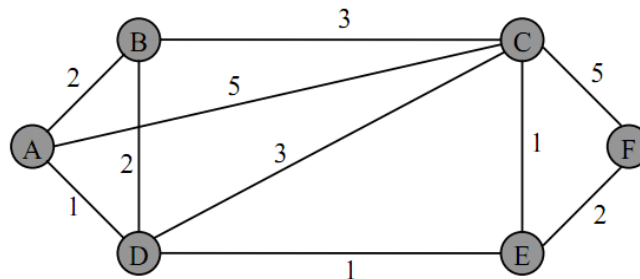
* به دلیل اینکه پروتکل DHCP بر اساس Broadcast کار می‌کند در زیرشبکه‌ها به ازای هر شبکه یک DHCP Server جدا نیاز است، چرا که روترها آدرس Broadcast را از خود عبور نمی‌دهند.

البته می‌توان روتری با قابلیت DHCP Relay قرار داد (عبور پیام DHCP از روتر و ارسال به DHCP Server). و یک سرور در یکی از زیر شبکه‌ها برای DHCP قرار می‌دهیم. البته امروزه روترها کار DHCP Server را نیز به طور کامل انجام می‌دهند.

* DHCP در لایه‌ی کاربرد است و از پروتکل‌های قدیمی تر مانند RARP (برعکس ARP عمل می‌کند) استفاده می‌کند.

مسیریابی (Routing)

در شکل زیر فرض کنید قرار است بسته ای از گرهی A پس از طی مسیری به گرهی F تحویل گردد؛ اصلی ترین وظیفه‌ی الگوریتم‌های مسیریابی، پیدا کردن مسیری بهینه از A تا F است به گونه‌ای که هزینه‌ی کل مسیر به حداقل برسد.



دو مسئله‌ی مهم در مسیریابی مطرح است:

۱- هر یک از مسیریاب‌ها چگونه از پارامتر هزینه‌ی کل کانال‌ها مطلع شوند؛ تا بتوانند گراف زیر ساخت ارتباطی شبکه را تشکیل داده و بهترین مسیر را انتخاب کنند؟

۲- چه الگوریتمی برای یافتن مسیر بهینه استفاده شود که از لحاظ پیچیدگی زمانی الگوریتم بهینه باشد؟

مسیریابی در دو بخش صورت می‌گیرد:

الف) Routing: الگوریتم‌های مسیریابی بهینه را اجرا می‌کند و اطلاعات شبکه‌ها را بین مسیریاب‌ها مبادله می‌نماید. و بر اساس این اطلاعات جداول Forwarding (جداول مسیریابی) را تشکیل می‌دهد. و هر چند ثانیه یکبار اطلاعات بین مسیریاب‌ها رد و بدل می‌شود به عبارت دیگر همیشه در حال اجرا است.

ب) Forwarding: در این فاز پورت (درگاه) خروجی بر اساس اطلاعات موجود در جداول Forwarding انتخاب می‌شود. این بخش همیشگی نیست و فقط زمانی اجرا می‌شود که یک بسته به مسیریاب برسد و قرار است پورت خروجی‌اش انتخاب گردد.

تفاوت Routing و Switching

در روتینگ مسیریابی انجام می‌شود و بحث بر سر روابط بین شبکه‌ها است و در لایه‌ی Inter Network فعالیت می‌کند؛ در صورتی که Switching در شبکه‌ی داخلی صورت می‌گیرد و در لایه‌ی Data Link و با آدرس MAC فعالیت می‌کند.

انواع Switching

۱- Circuit Switching: در شبکه‌های تلفن استفاده می‌شود. (ارتباط دو نفر در مرکز با یکدیگر)

۲- Message Switching



۳- Packet Switching: در شبکه‌های کامپیوتری استفاده می‌گردد؛ در این نوع سوییچ کردن داده‌ها را به قطعات کوچکتری شکسته و آدرس مقصد روی بسته‌ها قرار می‌گیرد و به گام بعدی ارسال می‌شود. بسته‌ها جداگانه و از روترهای مختلف به سمت مقصد حرکت می‌کنند (مسیرهای مختلف) و به صورت نامنظم به مقصد می‌رسند.

مسیریابی به روش Virtual Circuit Switching (VC)

در دنیای امروز مخابرات و دنیای شبکه به یکدیگر نزدیک و متصل هستند و برای مسیریابی از Virtual Circuit Switching استفاده می‌کنند؛ به این شکل که بر روی Packet انجام می‌شود و بر خلاف سوییچ کردن مورد اول داده‌ها در اینجا به بسته‌های کوچکتر شکسته می‌شوند و بر خلاف مورد ۳ که از روترهای مختلف بسته‌ها عبور می‌کردند، در این حالت ابتدا با انجام مسیریابی یک مسیر انتخاب شده و تمام بسته‌ها از همان مسیر عبور خواهند کرد.

مسیریابی به روش Datagram

این روش بر اساس حالت سوم عمل می‌کند، یعنی برای هر بسته مسیریابی جداگانه صورت می‌گیرد و ممکن است بسته‌های مختلف از مسیرهای متفاوتی به سمت مقصد ارسال گردند.

الگوریتم‌های مسیریابی

بر اساس جمع‌آوری اطلاعات

۱- متمرکز (سراسری)

در این الگوریتم یک Node یا گره برای تصور شمای کلی گراف زیرساخت باید تمام اطلاعات سایر گره‌ها را داشته باشد؛ به این الگوریتم‌ها Link State نیز می‌گویند.

۲- غیر متمرکز

در این الگوریتم‌ها مسیریاب اطلاعات کاملی از شبکه ندارد و فقط قادر است هزینه‌ی ارتباط به مسیریاب‌هایی که به طور مستقیم با آنها در ارتباط است دست پیدا کند. به عبارت دیگر هر مسیریاب جداول مسیریابی خود را برای مسیریاب‌های مجاور ارسال می‌کند. (Distance Vector)

بر اساس هوشمندی

۱- ایستا (Static)

در این حالت مسیر به صورت دستی تنظیم می‌شود و تمام مسیریابی‌ها از روی مسیر تنظیم شده صورت می‌گیرد، مشکل این روش این است که در صورت زیاد بودن روترها مدیر باید تمام مسیرها را برای آنها تعریف کند که وقت‌گیر است و دیگر عیب آن عدم مدیریت و توجه به وضعیت ترافیک مسیرها در حین مبادله است.



۲- پویا (Dynamic)

در این الگوریتم بستگی به وضعیت شبکه ترافیک و مسیرهای موازی تعریف شده؛ و برای پیدا کردن شمای کلی شبکه بر اساس مبادله‌ی داده بین لینک‌ها (مسیریاب‌ها) و بررسی تاخیر (Delay) در هر لینک اطلاعاتی بدست خواهد آمد که آنرا به دیگر روترها ارسال می‌نماید تا در نهایت شمای کلی بوجود آید. عیب روش پویا ایجاد ترافیک مسیریابی است (Routing Traffic). به دلیل اینکه ترافیک مسیریابی جز داده‌های اصلی محسوب نمی‌گردد سر بار در شبکه ایجاد خواهد شد. اگر تعداد روترها کم باشد می‌توان از الگوریتم‌های ایستا استفاده کرد اما اگر روترها در کل مسیر فعال و غیر فعال گردند (متغیر) روش ایستا نامناسب خواهد بود و باید از روش پویا استفاده کرد.

بر اساس زمان اجرا

۱- پیش دستانه (Pro Active)

از آنجا که بسته‌ها بر اساس جدول مسیریابی ارسال می‌شوند، روش پیش دستانه قبل از اینکه بسته‌ای ارسال شود، جدول Forwarding را ایجاد و نگهداری می‌کند تا اگر بسته‌ای آمد بر اساس آن عمل کند، این جدول در بازه‌های زمانی معین به روزرسانی می‌گردد. از مزایای این الگوریتم در دسترس بودن فوری مسیرها در آغاز برقراری ارتباط بین گره‌ها می‌توان نام برد.

۲- واکنشی (On Demand یا Re Active)

در این روش اگر بسته‌ای نباشد جدولی نیز نگهداری نمی‌شود (یعنی اطلاعات از مسیریابها ارسال نمی‌شود) و اگر بسته‌ای وارد شود آنگاه جدول تشکیل خواهد شد. تغییرات (بروزرسانی) در آن کم است. این الگوریتم برای شبکه‌های بزرگ مناسب است.

در شبکه‌هایی که شکل آن زیاد تغییر پیدا نمی‌کند از Pro Active استفاده شده و در شبکه‌هایی که شکل آن زیاد تغییر می‌کند از Re Active استفاده می‌گردد، چرا که وقتی بسته‌ای وارد می‌شود و در جدول مسیر آن یافت نمی‌گردد، جدول جدید تشکیل می‌شود و در هنگام وصول بسته‌ی بعدی معمولاً چون جدول جدید است، تغییرات نیز در آن اعمال شده است و نیازی به تغییر جدید نیست. در شبکه‌های وایرلس از Re Active استفاده می‌شود. در شبکه‌های سیمی نیز از Pro Active استفاده می‌شود.

Wi-Fi:

DCF

نماینده‌ی این الگوریتم Adhoc است؛ در این حالت تجهیزات وجود ندارند و هر وقت نیاز باشد بر اساس تجهیزات موجود شبکه برپا می‌شود. از مثال‌هایی که می‌توان برای Adhoc بیان کرد در بحث نظامی است که در میدان جنگ با تجهیزات روی وسائط نقلیه‌ی نظامی (بیسیم‌ها) ارتباط شبکه‌ای برقرار می‌شود. معمولاً شبکه‌ی Adhoc داده‌ها را به شبکه‌ی Infra structure (اینترنت و ...) ارسال می‌کند.



* برای پیدا کردن گراف شبکه و ایجاد جدول مسیریابی از دو روش Link State (وضعیت پیوند) که متمرکز است و از Distance Vector (بردار فاصله) که غیر متمرکز است استفاده می‌شود.

مسیریابی سیل آسا (Flooding)

در این روش مسیریابی هر بسته‌ی ورودی روی تمام لینک‌های خروجی غیر از لینکی که بسته از آن وارد شده است ارسال می‌شود. تمام مسیرهای بین راه نیز همین کار را انجام می‌دهند تا بسته به مقصد برسد. مزیت این روش این است که بسیار سریع عمل می‌کند و اگر مسیری باشد آنرا پیدا می‌کند. از معایب این روش نیز این است که یک بسته ممکن است چند بار به یک روتر برود (حلقه) و ترافیک بیهوده بالا برود + اضافه طی شدن مسیر، عیب دیگر آن این است که یک بسته ممکن است بیش از یکبار به مقصد برسد.

روش Link State

در روش Link State هر روتر وظیفه دارد همسایه‌های خود را پیدا کرده (یعنی آدرس‌ها را پیدا کرده؛ کدام روتر به کدام پورت متصل است). از دیگر مسائل مهم هزینه‌ی هر پورت است که معمولاً تاخیر (delay) دریافت و ارسال است. و در ادامه برای هر روتر یک جدول ایجاد می‌کنیم تا اطلاعات در آن ثبت شود و شمای کلی گراف برای ایجاد جدول Forwarding بدست آید.

به صورت خلاصه پنج عمل زیر در این روش صورت می‌گیرد:

- ۱- بدست آوردن آدرس مسیریاب‌هایی که در مجاورت قرار دارند و به صورت فیزیکی با آنها در ارتباط است.
- ۲- اندازه‌گیری تاخیر (هزینه) مسیریاب‌های مجاور
- ۳- ایجاد یک بسته و قرار دادن اطلاعات بدست آمده از مسیریاب‌های مجاور
- ۴- بسته‌های ایجاد شده را برای تمام مسیریاب‌های شبکه ارسال کند (از طریق روش سیل آسا)
- ۵- بدست آوردن مسیر بهینه بین دو مسیریاب در شبکه با استفاده از الگوریتم‌های موجود.

محتویات بسته‌های State Link:

- ۱- آدرس جهانی مسیریاب تولید کننده‌ی بسته
- ۲- شماره ترتیب هر بسته (برای تشخیص قدیمی یا جدید بودن بسته و یا غیر تکراری بودن بسته)
- ۳- طول عمر بسته
- ۴- آدرس مسیریاب‌های مجاور و هزینه‌ی رسیدن به آنها



روش Distance Vector

در این روش هر مسیر یاب اطلاعاتی را که از کل شبکه جمع آوری کرده است فقط به همسایه‌های خود ارسال می‌کند. و به ازای هر مسیر یاب یک رکورد وجود دارد که در این رکورد دو فیلد پورت خروجی و هزینه‌ی تقریبی رسیدن به یک مقصد خاص وجود دارد.

مراحل این الگوریتم:

۱- محاسبه هزینه‌ی خطوطی که یک مسیر یاب به صورت فیزیکی با مسیر یابهای دیگر دارد و ثبت آنها در جدول. هزینه‌ی خطوطی که مسیر یاب با آنها ارتباط مستقیم ندارد بی‌نهایت در نظر گرفته می‌شود.

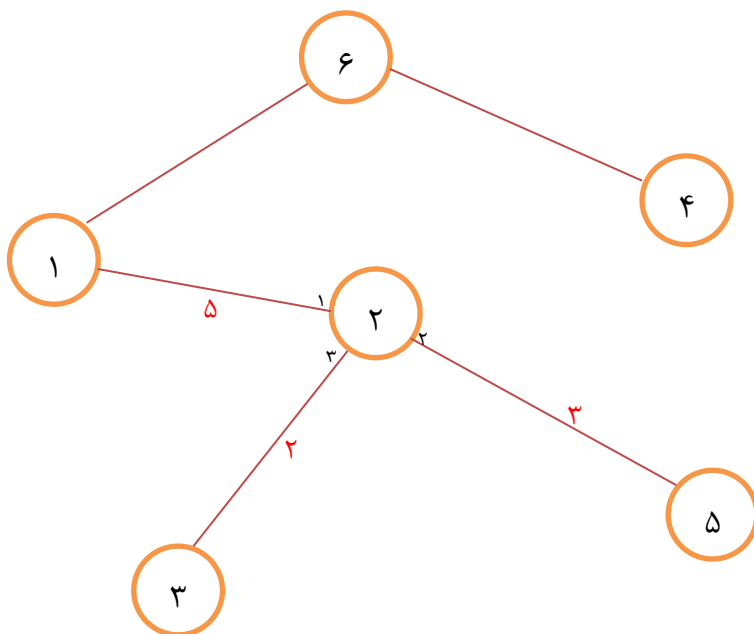
۲- هر مسیر یاب در زمان‌های مشخص (دوره‌ای) جدول مسیر یابی خود را برای مسیر یاب‌های مجاور ارسال می‌کند.

۳- هر مسیر یاب پس از دریافت جداول مسیر یابی از روترهای همسایه، هزینه‌ی مسیرها را محاسبه می‌کند.

* این الگوریتم به عنوان الگوریتم پویا نیز شناسایی می‌شود.

* مشکل این الگوریتم این است که در هنگام خرابی یک مسیر جداول مسیر یابی با یکدیگر همگرایی سریعی نخواهند داشت.

در شکل زیر چگونگی ایجاد جدول برای یک مسیر یاب نشان داده شده است؛



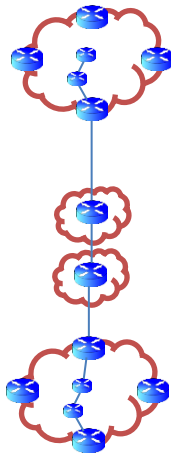
جدول نمونه برای روتر ۲

نام روتر	پورت	هزینه
1	1	5
3	3	2
5	2	3

نمایندگی روش‌های OSPF, Link State است و نمایندگی روش‌های RIP, Distance Vector است.

مسیریابی در اینترنت (مسیریابی سلسله مراتبی)

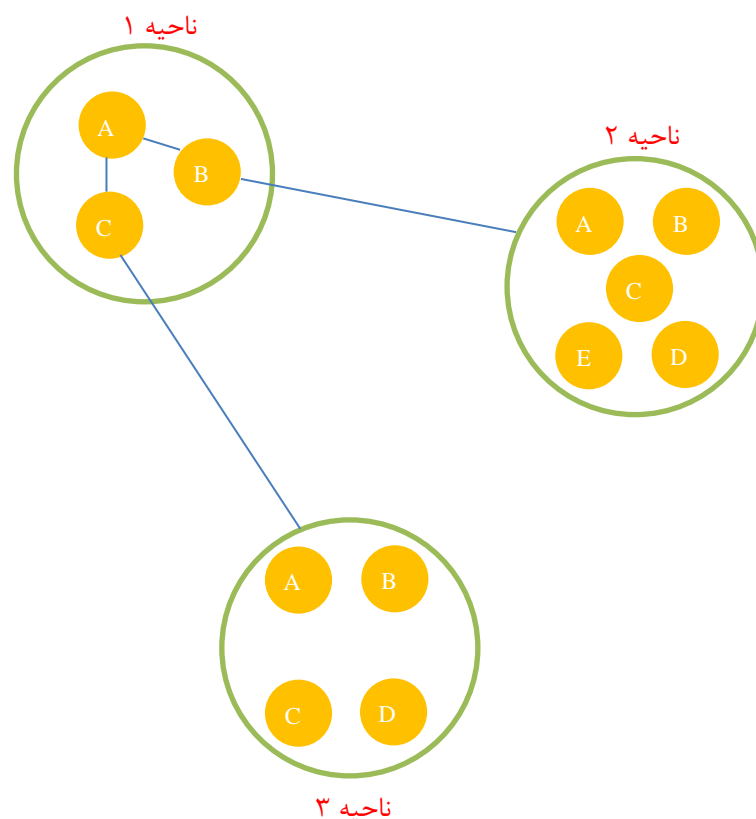
در این نوع مسیریابی، مسیر یاب‌ها در گروه‌هایی به نام ناحیه دسته‌بندی می‌شوند و هر مسیر یاب نواحی و مسیر یاب‌های درون ناحیه‌ی خود را می‌شناسد و اطلاعاتی در مورد مسیر یاب‌های نواحی دیگر ندارد.



این نوع مسیریابی سه فاز دارد:

- ۱- مسیریابی از مبدا تا دروازه‌ی ناحیه‌ی مبدا
- ۲- مسیریابی بین ناحیه‌ی مبدا تا ناحیه‌ی مقصد
- ۳- مسیریابی بین دروازه‌ی ناحیه‌ی مقصد تا مقصد نهایی

روترهای بین مبدا و مقصد فقط اطلاعات مسیره‌ها برای آنها کفایت می‌کند.



در شکل فوق در ناحیه‌بندی به جای اینکه اطلاعات ۱۲ روتر را یکجا داشته باشد، کافی است هر روتر فقط ارتباطات خود را بداند.

روش‌های Link State و Distance Vector درون ناحیه‌ای هستند که به آنها IGP نیز گفته می‌شود. در بین ناحیه‌ها نیز از پروتکل‌های متفاوتی استفاده می‌شود که به‌طور کلی به آنها EGP گفته می‌شود. از پروتکل‌های معروف بین ناحیه‌ای BGP را می‌توان نام برد.

در پروتکل EGP به ناحیه‌ها AS (خودمختار) گفته می‌شود. ASها مجموعه ناحیه‌هایی هستند که تحت یک مدیریت واحد هستند؛ هر AS یک ID دارد. ASها به‌طور مثال برای کشورها ثبت می‌شوند و بین آنها BGP کار مسیریابی را انجام داده و و حداقل یک مسیر با AS دیگر پیدا خواهد کرد.



هر BGP یک Policy دارد که مدیر شبکه می‌تواند تنظیم کند که ترافیک از یک AS گذر کند یا خیر. در اینجا برای ASها تعریف Stub (بن بست) و Transit (انتقال) صدق می‌کند.



لایه‌ی انتقال

توصیف کلی این لایه این است که جزئیات را از دید برنامه‌نویسان مخفی می‌کند و کارهای Socket و Http را خود لایه انجام می‌دهد. در حقیقت این لایه مشکلات لایه‌ی شبکه را که پروتکل IP در آن قرار دارد برطرف می‌کند و معمولاً خدمات این لایه به صورت مطمئن و با رفع خطا صورت می‌گیرد.

وظایف تعریف شده برای لایه‌ی Transport

در این لایه دو سرویس ارائه می‌شود؛ سرویس اتصالگرا و سرویس مطمئن البته بسته به نیاز برنامه‌ها دو پروتکل مهم در این لایه TCP که سرویس اتصالگرا و مطمئن ارائه می‌دهد و دیگری UDP که از لحاظ نوع سرویس مشابه IP است یعنی اینکه غیر اتصالگرا و نامطمئن است.

کاستی‌های IP

- ۱- عدم آگاهی از آمادگی گیرنده
- ۲- عدم حفظ ترتیب بسته‌ها
- ۳- IP مکانیزمی برای توزیع داده‌ها بین برنامه‌ها ندارد
- ۴- عدم اطمینان از رسیدن بسته‌ها
- ۵- عدم هماهنگی سرعت فرستنده و گیرنده
- ۶- عدم تشخیص بسته‌های تکراری

* در لایه‌ی انتقال داده‌ها به صورت Segment هستند.

* کاستی‌های فوق در حقیقت جز وظایف IP نیستند.

پروتکل TCP راهکارهایی برای برطرف کردن کاستی‌های بیان شده دارد. مطلبی که در ادامه بیان می‌شود TCP استاندارد است؛ امروزه این پروتکل اصلاح گردیده است.

راهکارهای TCP

- ۱- 3-Way Handshaking یا دست تکانی سه مرحله‌ای
- ۲- درج شماره ترتیب؛ Seq.No
- ۳- شماره پورت برای برنامه‌ها؛ به هر برنامه یک Port.No اختصاص داده می‌شود.
- ۴- ارسال Ack از گیرنده به فرستنده

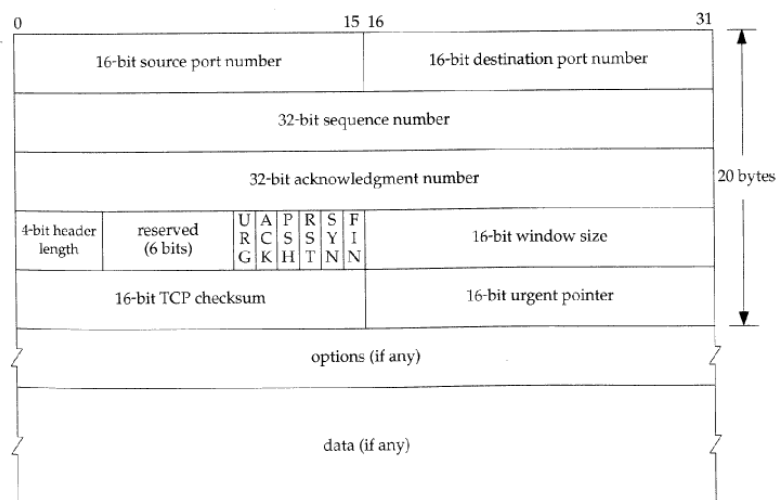


۵- اعلام Windows Size از گیرنده به فرستنده (Windows Size بافر خالی گیرنده است که همراه با Ack اعلام می‌شود). فرستنده اجازه‌ی ارسال داده بیشتر از اندازه‌ی Windows Size را ندارد، البته در هنگام اعلام حجم، حجم بسته‌ی ارسالی قبلی از آن کم می‌شود.

۶- Seq.No؛ وقتی که Seq.No و Port.No دو بسته مشابه هم باشد، این پروتکل تشخیص به تکراری بودن بسته می‌دهد. در ادامه چگونگی انجام وظایف توسط TCP بررسی می‌شود.

سرآیند پروتکل TCP

همانگونه که مشاهده می‌کنید ۲۰ بایت اول این سرآیند اجباری است.



فیلد Source Port

این فیلد ۱۶ بیتی آدرس پورت فرستنده را به همراه دارد

فیلد Destination Port

این فیلد ۱۶ بیتی آدرس پورت مقصد را که باید تحیل شود را دارا است.

** ۱۰۲۴ پورت اول، پورت‌های استاندارد هستند که مشخص شده است هر برنامه از چه پورتهی استفاده کند؛ برای مثال: http پورت ۸۰؛ Ftp پورت ۲۰ و ۲۱؛ Smtip پورت ۲۵؛ Telnet پورت ۲۳ و ...
 ۱۰۲۴ پورت دوم، پشتیبان پورت‌های فوق هستند و مابقی برنامه‌ها از شماره‌ی ۲۰۴۸ شماره‌گذاری می‌شوند.



فیلد Sequence Number

این فیلد ۳۲ بیتی شماره ترتیب آخرین بایتی را که در فیلد داده از بسته جاری قرار دارد را نشان می‌دهد. در پرتکل TCP شماره ترتیب، بر حسب شماره آخرین بایتی است که در بسته جاری قرار دارد. به عنوان مثال اگر در این فیلد عدد ۱۹۳۴۱ قرار بگیرد به این معناست که داده‌ها تا بایت ۱۹۳۴۱ درون این فیلد داده‌ها قرار دارد. برای حل مشکلات شماره ۲ و ۶ بیان شده در فوق (IP) استفاده می‌شود.

فیلد Acknowledgment number

این فیلد ۳۲ بیتی شماره ترتیبی بایتی است که گیرنده بسته برای تأیید به فرستنده ارسال می‌کند که داده‌ها تا بایتی که در این فیلد قرار دارد به درستی دریافت شده است. به عنوان مثال اگر در این فیلد عدد ۱۲۳۶۵ قرار گرفته شود به این معنی است که داده‌ها تا بایت ۱۲۳۶۵ صحیح و کامل دریافت شده است و در انتظار بایتهای ۱۲۳۶۶ به بعد می‌باشد. ($Ack=Seq+1$)

فیلد TCP Header Length

عددی که در این فیلد قرار می‌گیرد طول کل سرآیند بسته TCP بر مبنای کلمات ۳۲ بیتی تعیین می‌کند. به عنوان مثال اگر در این فیلد عدد ۷ قرار بگیرد طول سرآیند بسته برابر است با $7 \times 4 = 28$ بایت خواهد بود (این فیلد کلاً چهار بیتی است).

بیت‌های Flag

❖ **بیت URG:** در صورتی که در این بیت عدد ۱ قرار گیرد معین می‌شود که در فیلد Urgent Pointer مقدار قابل معتری قرار دارد و بایستی مورد پردازش قرار گیرد. به عبارت دیگر قسمتی از این سگمنت باید فوری ارسال گردد و محل ارسال نیز در Urgent Pointer قرار می‌گیرد.

❖ **بیت ACK:** اگر در این بیت عدد ۱ قرار داشته باشد به این معناست که در فیلد Acknowledgment number عدد معتبری قرار دارد. بیت‌های ACK و SYN نقش دیگری نیز دارند که در ادامه بدان اشاره خواهد شد.

❖ **بیت PSH:** اگر این بیت مقدراً ۱ قرار گرفته باشد از گیرنده تقاضا می‌شود که داده‌های موجود را بافر نکرده و در اسرع وقت تحویل برنامه کاربردی صاحب آن شود.

❖ **بیت RST:** اگر در این بیت عدد ۱ قرار گرفته شود به این معناست که این ارتباط به صورت یک طرفه خاتمه یافته است.

❖ **بیت SYN:** این بیت نقش اساسی در ارتباط یک بسته TCP بازی می‌کند. برقراری ارتباط یک طرفه TCP از روند زیر تبعیت می‌کند؛

الف) شروع کننده ارتباط یک بسته TCP بدون هیچ داده‌ای و با تنظیم بیت‌های $ACK=0$ و $SYN=1$ تقاضای یک ارتباط جدید می‌کند.

ب) در صورتی که طرف مقابل تمایل به برقراری ارتباط داشته باشد برای طرف مقابل یک بسته با قرار دادن بیت‌های $ACK=1$ و $SYN=1$ خود را برای برقراری ارتباط به طرف مقابل اعلام می‌کند.



❖ **بیت FIN:** اگر یکی از طرفین هیچ داد دیگر برای فرستادن نداشته باشد این بیت را در آخرین بسته برابر ۱ قرار می‌دهد و ارتباط را یک طرفه قطع می‌کند باید توجه داشته که ارتباط هنوز به طور کامل قطع نشده است و باید طرف مقابل نیز در آخرین بسته خود این فیلد را برابر ۱ قرار داده تا ارتباط کامل قطع شود.

فیلد Window size

مقدار قرار گرفته در این فیلد مشخص می‌کند که مقدار بافر گیرنده چند بایت دیگر فضای خالی دارد. در نسخه‌های جدید TCP یک ضریب در Option برای Windows Size تعیین می‌شود که بتوان اطلاعات بیشتری ارسال کرد.

فیلد Checksum

در این فیلد ۱۶ بیتی کد کشف خطا قرار می‌گیرد.

فیلد TCP Segment length

در آن طول کل بسته TCP قرار می‌گیرد.

فیلد Urgent Pointer

در این فیلد عدد به عنوان اشاره‌گر قرار می‌گیرد که موقعیت داده‌های اضطراری را درون بسته مشخص می‌کند. این داده‌ها زمانی اتفاق می‌افتد و ارسال می‌شود که عملی شبیه وقوع وقفه در هنگام اجرای یک برنامه کاربردی رخ دهد. بدون آنکه ارتباط قطع شود داده‌ها درون همین بسته جاری قرار گرفته و ارسال می‌شود. لازم به ذکر است که از این فیلد، لایه‌های بالاتر استفاده می‌کنند.

پروتکل دست تکانی سه مرحله‌ای (3-Way Handshaking)

برای برقراری ارتباط در پروتکل TCP از روش «دست تکانی سه مرحله‌ای» استفاده می‌شود. این سه مرحله عبارتند از:

در مرحله اول از طرف شروع کننده ارتباط یک بسته TCP خالی از داده ارسال خواهد شد که در آن بیت $SYN=1$ و بیت $ACK=0$ است و درون فیلد شماره ترتیب عدد x قرار داده شده که در آن x یک عدد تصادفی است. در حقیقت با این شماره به طرف مقابل اطلاع داده می‌شود که ترتیب داده‌های ارسالی از $x+1$ شروع می‌شود. در پروتکل TCP برای پیشگیری از مشکلات ناشی از مساوی بودن شماره‌ی ترتیب بسته‌های ارسالی، داده‌ها از شماره‌ی صفر شروع نمی‌شوند، بلکه از یک عدد تصادفی که به صورت خودکار تولید می‌شود، آغاز می‌گردد و در همان مرحله‌ی اول، این شماره ترتیب به طرف مقابل اعلام خواهد شد.



در مرحله دوم طرف مقابل با دریافت بسته ای با مشخصات فوق الذکر اگر تمایل به برقراری ارتباط نداشته باشد با ارسال یک بسته خالی که در آن بیت RST به ۱ تنظیم شده این تقاضا را رد می کند. ولی اگر متمایل به برقراری ارتباط بود یک بسته ی خالی از داده با مشخصات زیر تولید می کند:

- بیت SYN را یک می کند.
- بیت ACK را یک می کند.
- مقدار فیلد Acknowledgement Number را $x+1$ قرار می دهد.
- مقدار فیلد Sequence Number را مقدار تصادفی y قرار می دهد.

در این مرحله که به معنای پذیرش ارتباط است طرف مقابل با قرار دادن مقدار فیلد $Ack=x+1$ نشان می دهد که شماره ترتیب x را پذیرفته و منتظر داده ها از شماره ترتیب $x+1$ به بعد است. در ضمن خودش عدد تصادفی y را در فیلد Seq.No قرار می دهد و به طرف مقابل اعلام می کند که شماره ترتیب داده های ارسالی از y خواهد بود.

در مرحله سوم شروع کننده ارتباط با قرار دادن مقادیر زیر شروع ارتباط را تصدیق می کند:

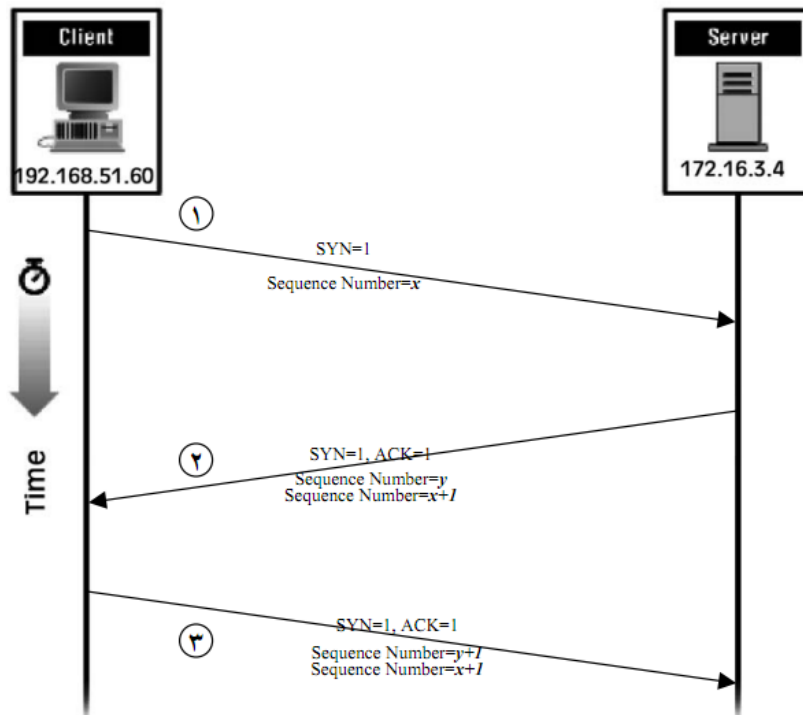
- بیت SYN را یک می کند.
- بیت ACK را یک می کند.
- فیلد $Seq.No=x+1$ را قرار می دهد.
- فیلد Ack را $y+1$ قرار می دهد.

پس از این مرحله ارسال و دریافت داده ها توسط طرفین تا هنگامی که ارتباط با اطلاع طرفین خاتمه داده نشده است آزاد است.

برای خاتمه ارتباط روند زیر صورت می گیرد:

طرفی که داده هایش برای ارسال تمام شده است یک بسته ی TCP ارسال می نماید که در سرآیند آن بیت FIN را یک قرار داده است. طرف مقابل این درخواست را دریافت می کند و با ختم یک طرفه ی آن موافقت می کند. ولی چون ارتباط به صورت یک طرفه ختم می شوند طرف مقابل می تواند تا جایی که داده دارد آن را ارسال کند و در نهایت در آخرین بسته بیت FIN را یک بگذارد تا پس از تصدیق آن، ارتباط به صورت دو طرفه ختم شود.

کلیه ی مراحل انجام شده در پروتکل Three-Way Handshake بین دو ترمینال در شکل زیر به تصویر کشیده شده است؛



نکته‌ای که وجود دارد آنست که اگر یکی از طرفین ارتباط در اثر بروز مشکلی سخت‌افزاری یا نرم‌افزاری ارتباط را بدون هماهنگی طرف مقابل قطع کند، حق ندارد تا ۱۲۰ ثانیه به ارتباط مجدد با همان پروسه اقدام کند. دلیل این کار این است که مطمئن باشد بسته‌های قبلی که ارسال کرده یا آنکه برایش ارسال شده از زیر شبکه حذف شده‌اند.

* TCP به جای ایجاد ارتباط Full Duplex دو ارتباط Half Duplex ایجاد می‌کند.

یکی از حملات D.O.S، حمله‌ی SYN flood است که حمله‌کننده تعداد زیادی SYN ارسال می‌کند.

مکانیزم کنترل ازدحام در TCP (Congestion Control)

در کنترل جریان هماهنگ کردن فرستنده و گیرنده مطرح است، اما در کنترل ازدحام، شبکه مطرح است. از آنجا که کنترل ازدحام در شبکه ایت و روترها با لایه‌ی IP کار می‌کنند، پس TCP به طور مستقیم نمی‌تواند سرعت ارسال خود را با شبکه هماهنگ کند. برای اینکار با یک سرعت خاصی بسته‌ها را ارسال می‌کند، اگر برای TCP مشخص شود که بسته‌ها در بین راه خراب یا گم شده‌اند متوجه می‌شود که نباید با سرعت بیشتر از این ارسال صورت گیرد و ممکن است حتی سرعت را کاهش دهد؛ در غیراینصورت (گم نشدن بسته‌ها) سرعت ارسال را افزایش داده و مکانیزم فوق را دنبال می‌کند.

نسخه‌های مختلف پروتکل TCP عمده‌ترین تفاوت را در مکانیزم‌های کنترل ازدحام دارا هستند.

کنترل ازدحام دارای سه فاز است:

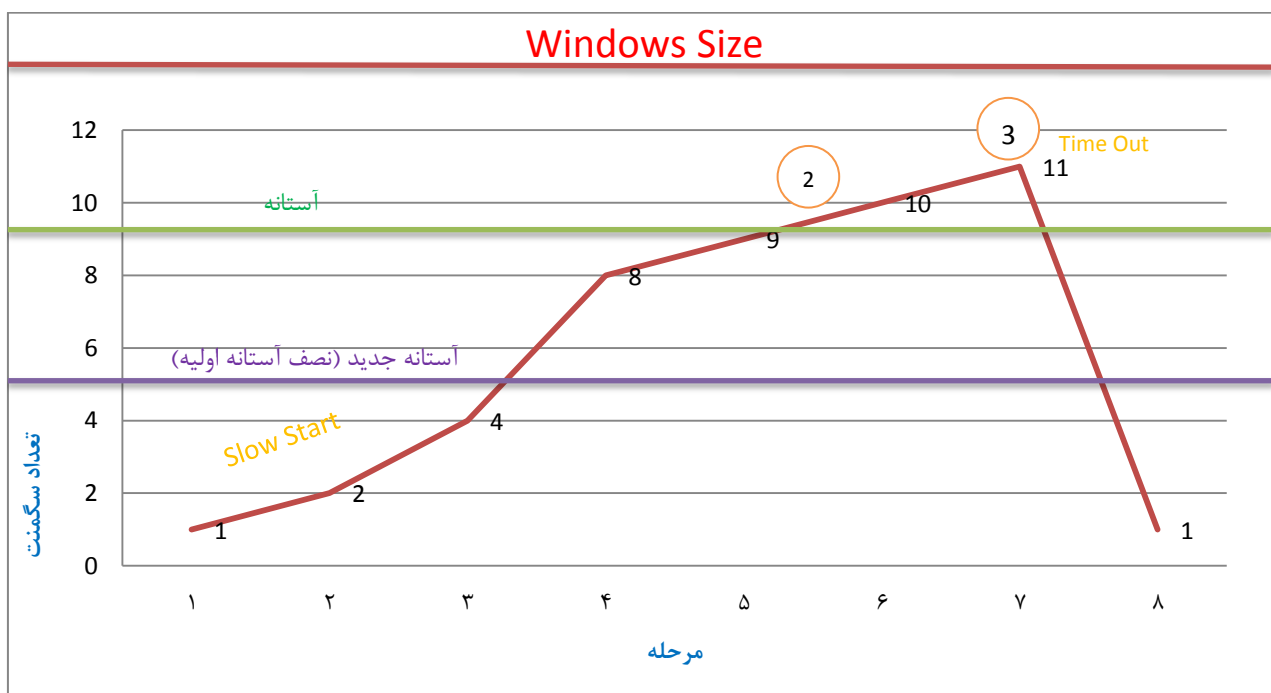
۱- شروع آهسته (Slow Start)

۲- اجتناب از ازدحام

۳- مقابله با ازدحام

در Slow Start در حالت استاندارد TCP از ارسال یک سگمنت شروع می‌کند، در صورتی که Ack بازگشت آنرا دوبرابر می‌کند و این روال تا زمانی که تشخیص دهد در حال بوجود آمدن ازدحام است ادامه می‌یابد به محض این تشخیص به جای ارسال دوبرابر فقط یک سگمنت به قبلی اضافه کرده و ارسال می‌نماید؛ اگر ازدحام رخ داد، مجدداً از یک شروع می‌کند و یک سگمنت ارسال می‌کند.

کار ارسال دوبرابر تا زمانی که به پارامتر آستانه (Threshold) برسد انجام می‌شود.



محدودیت دیگر در شکل فوق Windows Size است که ارسال نباید بیشتر از آن انجام گردد.

فرض TCP این است که هنگامی که Time Out (عدم دریافت بسته توسط گیرنده) رخ دهد، ازدحام رخ داده است. البته این فرض در شبکه‌های بیسیم نادرست است.

در مرحله سوم ضمن اینکه از اول شروع می‌کند، آستانه را نصف مرحله‌ای که خطا رخ داده کاهش می‌دهد که در مثال فوق $11/2$ است؛ به عبارتی دیگر نصف پنجره‌ی ارسال می‌گردد. (از صفر تا آستانه‌ی اولیه را پنجره‌ی ارسال گوییم).

مکانیزم فوق باعث می‌شود که TCP خود را با شرایط تطبیق دهد (Adaptive). و مکانیزم را آنقدر ادامه می‌دهد تا ارسال تقریباً خطی گردد و خطایی رخ ندهد.



زمان سنج‌ها در TCP

برای کنترل شرایط از تایمرهای مختلف در این پروتکل استفاده می‌شود.

(RT) Retransmission Timer

هر سگمنتی که ارسال می‌کند یک تایمر شروع می‌شود. اگر سگمنت در زمان مقرر به مقصد برسد و Ack توسط فرستنده دریافت گردد داده درست ارسال شده است، در غیراینصورت داده مجدداً باید ارسال شود. مدت زمان انتظار برای دریافت Ack توسط این تایمر (RT) مشخص می‌شود.

به طور خلاصه برای محاسبه‌ی تایمر از میانگین وزن دار استفاده می‌شود؛ به ازای هر سگمنتی که ارسال می‌شود و Ack دریافت می‌گردد RTT محاسبه و میانگین وزن دار آن بدست می‌آید.

Keep Alive

زنده نگه داشتن ارتباط. اگر ارتباط بین طرفین برای مدت زمان خاصی Idle بماند، یک پیغام برای طرف مقابل ارسال می‌گردد، اگر Ack داده شد، ارتباط ادامه پیدا خواهد کرد؛ در غیراینصورت پس از چند پیغام بدون Ack ارتباط قطع می‌گردد. مقدار پیش فرض این تایمر بین ۵ الی ۴۵ ثانیه است.

Persistent Timer

اگر بافر گیرنده پر شود، قاعدتاً فرستنده چیزی را ارسال نمی‌کند. و همانطور که می‌دانیم باید منتظر بماند تا گیرنده فضای بافر خود را اعلام کند که همراه Ack ارسال می‌شود، اما از آنجا که ارسالی صورت نمی‌گیرد پس بافری نیز اعلام نمی‌شود، این تایمر مدت زمانی را منتظر می‌ماند و سپس برای گیرنده یک بسته ارسال می‌کند و وضعیت گیرنده را جویا می‌شود، اگر جوابی از گیرنده باز گردد به کار خود ادامه می‌دهد یعنی یا بافر خالی شده و به ارسال می‌پردازد و یا اینکه کماکان باید منتظر بماند. در صورتی که به این بسته پاسخی داده نشود ارتباط قطع می‌گردد.

Quite Timer

پس از بسته شدن یک ارتباط با یک شماره پورت، بقیه‌ی برنامه‌ها تا مدتی حق استفاده از آن پورت را ندارد. این مدت زمان توسط این تایمر مشخص می‌شود. دلیل این انتظار این است که ممکن است یک ارتباط بسته شده باشد اما کماکان بسته‌های سرگردان و یا به مقصد نرسیده‌ی آن ارتباط در شبکه وجود داشته باشند و پس از اتمام ارتباط به مقصد برسند. مقدار پیش فرض این تایمر بین ۳۰ الی ۱۲۰ ثانیه است.

Idle Timer

این تایمر برای آن است که اگر تلاش برای ارسال مجدد یک بسته بیش از حد متعارف انجام گردد، ارتباط به صورت یکطرفه از سمت فرستنده قطع شود. مقدار پیش فرض این تایمر ۳۶۰ ثانیه است.



سوکت (Socket)

مفهوم سوکت در اینجا نرم‌افزاری است، مانده مفهوم پورت در شبکه است. یعنی اینکه یک شماره است. (IP+Port را End Point یا نقطه‌ی انتهایی گویند).

سوکت‌ها API‌هایی هستند که برای برنامه‌نویسی شبکه استفاده می‌شوند.

منظور از سوکت همان Berkley Socket است؛ چرا که اولین بار توسط این دانشگاه ارائه شد.

سوکت واسطی است بین لایه‌ی Application (کاربرد) و لایه‌های پایین است؛ که لایه‌های زیرین را از دید برنامه‌نویس مخفی می‌کند.

API (Application Programming Interface): توابع و آبجکت‌های آماده‌ای هستند که جهت تسهیل در برنامه‌نویسی برای انجام کارهای مختلف مورد استفاده قرار می‌گیرند.

سوکت برکلی بر روی سیستم‌عامل‌های Linux نوشته شده است که در این سیستم‌عامل‌ها کار با ابزار جانبی مانند یک فایل دیده می‌شود. پس این سوکت، شبکه را مانند یک فایل می‌پندارد.

سوکت‌ها با لایه‌ها و پروتکل‌ها کار می‌کنند؛ مانند: سوکت TCP، UDP و ...

سوکتی وجود دارد که آنرا Raw Socket می‌نامند، این سوکت یک سوکت خام است و می‌توان سرآیند لایه‌ها را توسط خود برنامه‌نویس در آن ایجاد و تکمیل کرد. این سوکت برای انجام کارهای Professional استفاده می‌شود. مثلاً برای نوشتن Sniffer (نمایش مشخصات بسته‌های رسیده) از این سوکت استفاده می‌کنند. توابع کتابخانه‌ای آماده‌ی نصب بنام [win] pcap وجود دارد که کار با سوکت خام را راحت‌تر می‌کند.

ایجاد TCP Socket از نوع سنکرون و تک نخ (Single Thread)

برنامه‌ای تحت شبکه و به مدل Client/Server است. برنامه‌ی سمت سرور اجرا شده و همیشه در حال اجرا می‌ماند و حتی ممکن است واسط کاربری گرافیکی نداشته باشد.

نسخه‌ی سرور برای چت دو نفره‌ی ساده

اول ایجاد یک Socket()، پس از آن مشخص کردن پورت مورد استفاده و اعلام آن به سیستم‌عامل است؛ Bind(). البته اگر در هنگام ایجاد سوکت خطایی رخ ندهد این مرحله اجرا می‌شود. این مرحله تخصیص پورت از طرف سیستم‌عامل به برنامه است. اگر این مرحله انجام داده شد، پورت مورد نظر برای برنامه رزرو می‌شود. در مرحله‌ی بعد برنامه برای دریافت درخواست‌ها به سیستم‌عامل اعلام آمادگی می‌نماید؛ Listen() تابع این مرحله است. در TCP درخواست ارتباط یعنی SYN=1. وقتی Listen فراخوانی می‌شود، ممکن است برنامه سنکرون باشد یا آسنکرون. اگر برنامه سنکرون و از نوع تک نخ باشد، و Listen فراخوانی گردد، یعنی منتظر درخواست است؛ پس وقتی که این تابع صدا زده می‌شود، سیستم‌عامل برنامه را Suspend (موقتاً غیرفعال) می‌کند و تا آمدن درخواست آن را به همین حالت نگه می‌دارد و پس از رسیدن درخواست آن برنامه فعال می‌شود. اما اگر از نوع آسنکرون باشد مورد گفته شده (غیرفعال شدن) پیش نمی‌آید.

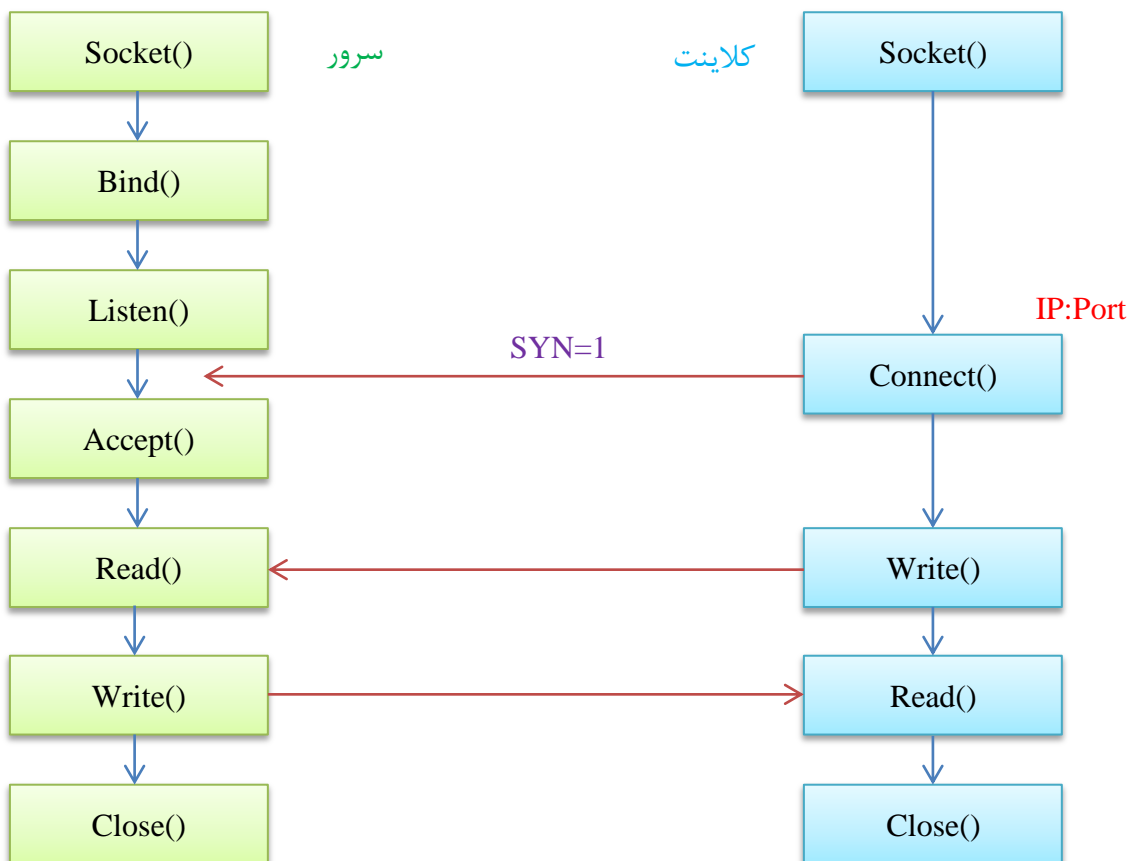
در سنکرون وقتی که برنامه خود هر مرتبه بررسی می‌نماید که آیا درخواستی آمده یا خیر را Pushing گویند. در مرحله‌ی بعد باید برنامه آزمایش شود که درست کار می‌کند یا خیر؟! برای اینکار در Command دستور `netstat -na` را اجرا می‌کنیم که اگر پورت مورد نظر در حالت Listen باشد، برنامه به درستی وظیفه‌ی خود را انجام می‌دهد. در هنگام رسیدن درخواست (SYN) سیستم‌عامل برنامه را فعال کرده و اگر برنامه موافق ارتباط باشد تابع `Accept()` را صدا می‌زند؛ که ارتباط برقرار شود و امکان ارسال و دریافت داده بوجود خواهد آمد. (توابع `Read()` و `Write()`). و مرحله‌ی آخر تابع `Close()` فراخوانی شده تا سوکت بسته شود. اگر سوکت بسته نشود در اجرای بعدی برنامه به دلیل عدم آزاد بودن پورت، برنامه دچار خطا می‌شود.

در حال حاضر برنامه‌ی سمت سرور به پایان رسیده است و در ادامه باید برنامه کلاینت نوشته شود؛ لیکن برای آزمایش صحت فعالیت برنامه‌ی سرور بدون کلاینت باید از دستور (شماره پورت آدرس مقصد `telnet`) استفاده کرد و اگر دو نسخه‌ی برنامه روی یک کامپیوتر باشد به صورت (`telnet 127.0.0.1 8080`) از دستور فوق استفاده خواهد شد. اگر هیچ پیغامی بر روی صفحه‌ی Command نشان داده نشد، یعنی اینکه برنامه‌ی سمت سرور به درست کار کرده و دارای خطا نمی‌باشد.

نسخه‌ی کلاینت برای چت دونفره‌ی ساده

در سمت کلاینت نیز ابتدا باید سوکت ایجاد شود، سپس اتصال به مقصد (سرور) توسط تابع `Connect()` که IP و پورت به آن ارسال می‌شود صورت می‌گیرد (IP:Port مقصد). پس از قبول ارتباط از جانب سرور می‌توان به ارسال و دریافت داده مشغول شد و در نهایت سوکت را `Close()` می‌کنیم.

در شکل زیر ارتباط سرور/کلاینت مشخص شده است: (فرض استفاده از پورت ۸۰۸۰ برای ارتباط است)





سیستم نام‌گذاری دامنه (Domain Name System) یا DNS

جزو پروتکل‌های لایه‌ی Application می‌باشد. یک پروتکل Client/Server است که کامپیوترهای شخصی بیشتر از نسخه‌ی کلاینت استفاده می‌کنند.

این سرویس یک پیغام ایجاد کرده و از سرور خواستار پیدا کردن IP نام وارد شده را می‌خواهد و سرور نیز IP را برای این سرویس بر می‌گرداند.

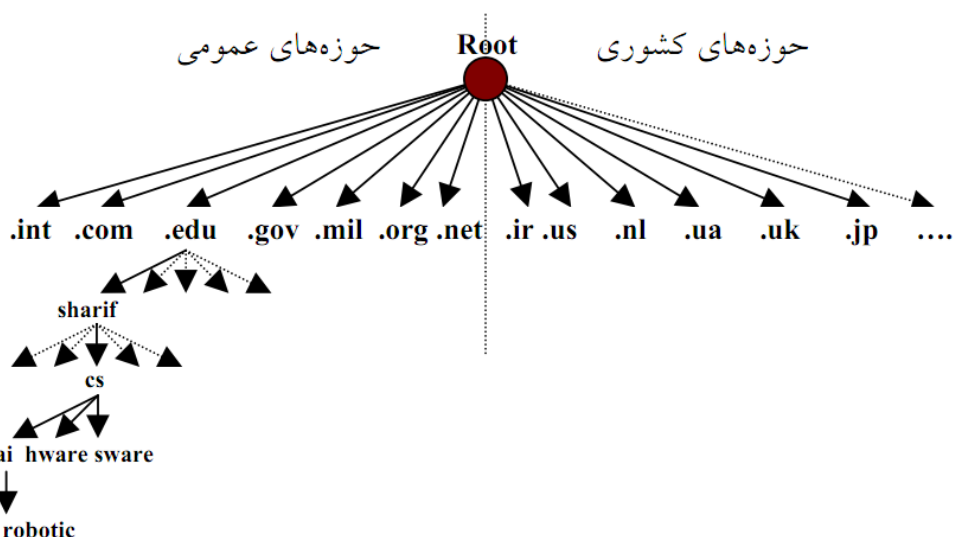
از آنجا که این سیستم بر مبنای درخواست و پاسخ است پس نتیجه می‌گیریم که بر مبنای UDP عمل می‌کند. و پروتکل UDP در هر مرحله تقاضا را ارسال و منتظر دریافت پاسخ می‌شود.

DNS مبتنی بر TCP فقط وقتی که دو سرور DNS خواهان سرویس‌دهی به یکدیگر هستند استفاده می‌گردد.

DNS یک پروتکل سلسله‌مراتبی است و برخلاف آدرس‌دهی بر روی هارد که با / عمل می‌کند، در این پروتکل برای جداسازی سطوح (حوزه‌ها) از یکدیگر از نقطه استفاده می‌شود. و شیوه‌ی بازبایی آن نیز برخلاف آدرس‌دهی کامپیوتر است و به صورت پایین به بالا است.

DNS Type . Domain Name . Domain Type

شکل زیر بیانگر نوع سلسله‌مراتب در DNS است:



در شکل فوق برای مثال برای دست‌یابی به حوزه‌ی (ماشین) robotic به صورت زیر آدرس آنرا می‌نویسیم؛

robotic.ai.cs.sharif.edu

البته در انتها یک نقطه نیز وجود خواهد داشت که نوشته نمی‌شود و متعلق به Root است.

برای پیدا کردن یک آدرس در سرور دیگر باز به صورت سلسله‌مراتبی و از انتهای آدرس وارد شده کار جستجو انجام می‌شود. عملیات نگاشت یا تبدیل DNS Name به آدرس IP را Resolve گویند.

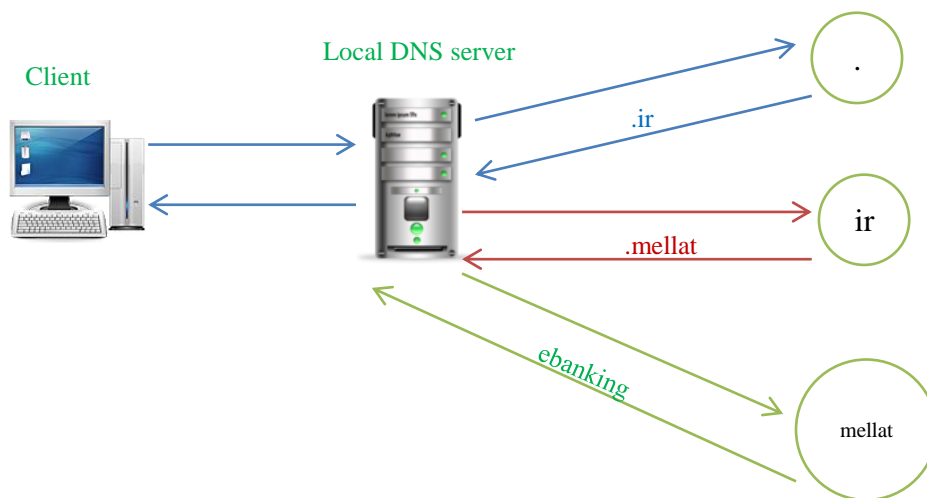


دو روش اصلی برای عملیات نگاشت وجود دارد:

روش تکراری (Iterative)

در این روش DNS Server محلی مسئول پیدا کردن آدرس نهایی است و تمام بار پردازش بر عهده‌ی آن است. این روش برای مقیاس بزرگ خوب است.

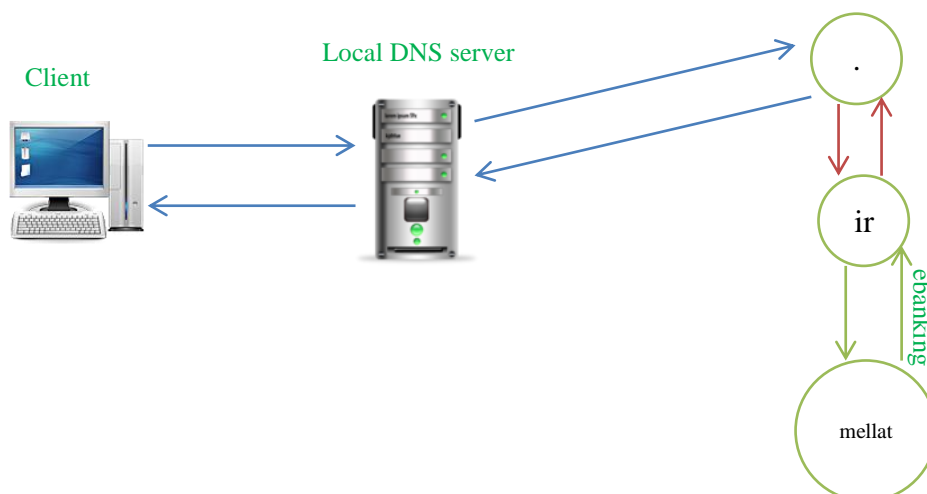
شکل زیر نمایانگر این نوع نگاشت (جستجو) است؛ فرض بر این است که ترمینال خواستار مراجعه به ebanking.mellat.ir است:



روش بازگشتی (Recursive)

در این روش بار پردازشی بر عهده‌ی Root یا بالاترین حوزه قرار می‌گیرد و عملاً می‌توان DNS Server محلی را حذف کرد، چرا که کار خاصی انجام نمی‌دهد! برای مقیاس‌های کوچک مناسب است.

شکل زیر نمایانگر این مدل جستجو است؛ (با همان فرض بالا)





اگر از دید DNS سرور محلی نگاه کنیم که مورد بازگشتی مناسب است ، چرا که حالت تکرار برای سرور محلی سربار دارد. در اینترنت از روش تکرار (iterative) استفاده می‌شود، چرا که اگر از روش بازگشتی استفاده شود تمامی سربار بر دوش Root قرار می‌گیرد و در صورت وصول تعداد زیادی درخواست، در خواستها باید منتظر بمانند تا Root به آنها پاسخ دهد، به عبارت دیگر در یک نقطه ازدحام بوجود می‌آید و تمام درخواستها به آن وابسته خواهند بود. در شبکه‌های محلی همواره یک DNS سرور محلی وجود دارد.

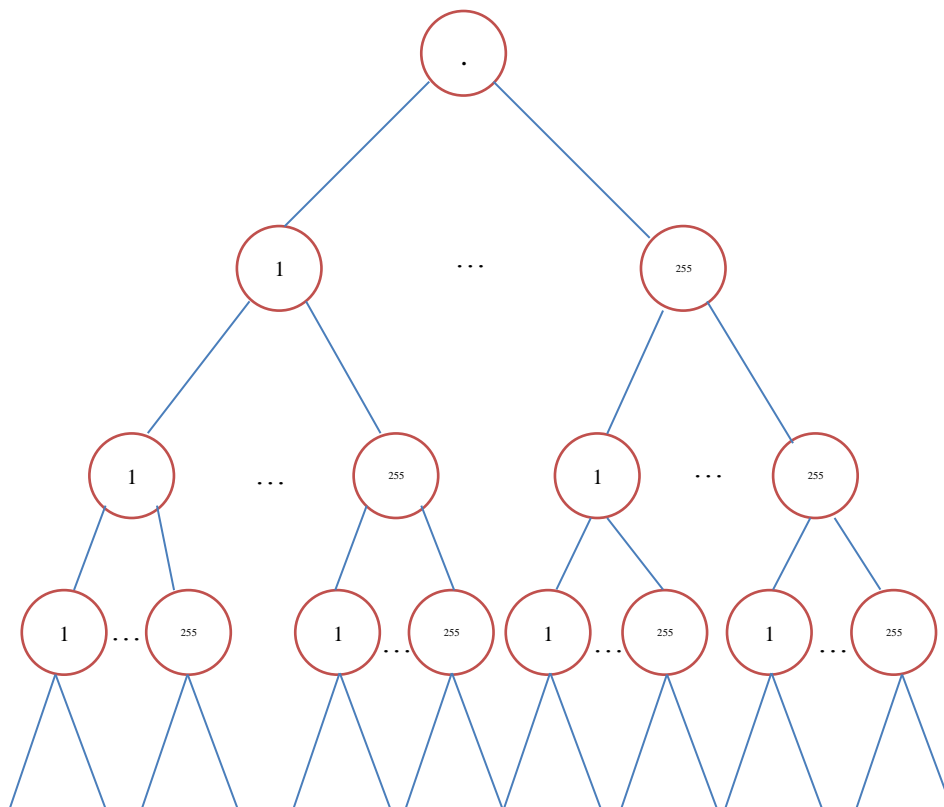
در جهان اینترنت DNS سروهایی وجود دارند که به صورت عمومی عمل کرده و به درخواستهای خارج از شبکه‌ی خود (خارج از حوزه) نیز پاسخ می‌دهند. دو DNS معروف در این زمینه:

گوگل با IP آدرس: 8.8.8.8

ياهو با IP آدرس: 4.2.2.4

روش معکوس (Reverse)

این روش برخلاف عملیات‌های نگاشت فوق عمل می‌کند؛ به عبارت دیگر آدرس IP موجود است و نام دامنه‌ی آنرا خواهیم. در نتیجه درخت تشکیل آن نیز برعکس است و سلسله مراتب آن در ۴ سطح خواهد بود. (چرا که IP چهار بخشی است) در این روش IP را از ابتدا و از بالا شروع به جستجو کرده تا به انتهای آن برسد، به دلیل یکپارچه بودن درخت در این حالت به روزرسانی آن دشوار و مشکل‌زا است.





مثال) آدرس IP دامنه‌ای بنام ut.ac.ir را چگونه بدست می‌آورید؟

استفاده از Ping ساده‌ترین راه برای اینکار است. راه دیگر استفاده از سایت‌های Who is!؟ است. (معروفترین سایت How is!؟ می‌توان <http://www.dnsstuff.com> را نام برد) دستور IP lookup، باعث برگرداندن تمام زیر مجموعه‌های یک DNS سرور می‌گردد.

DNS Cache

در هر DNS سرور محلی وقتی درخواستی برای آن برسد ابتدا کش خود را چک می‌کند، اگر IP نام حوزه‌ی درخواست شده در آن وجود داشت، دیگر درخواست را برای ریشه ارسال نمی‌کند و از روی خود سرور برای پاسخ‌گویی استفاده می‌شود که اینکار باعث افزایش سرعت پاسخ‌گویی به کلاینت می‌گردد.

URL

آدرس مستنداتی که نیاز است آن را در اختیار داشته باشیم؛ (سیستم آدرس‌دهی یکپارچه) شکل کلی آن به صورت زیر است:

آدرس سند روی ماشین مقصد/شماره پورت : آدرس ماشین مقصد//پروتکل
<http://192.168.130.100:8080/chat>

ساختار بانک اطلاعاتی DNS Server

اطلاعات به صورت text ذخیره می‌شوند. در این دیتابیس داده‌های لازم برای تحلیل یک نام نمادین، ذخیره می‌شوند. بانک اطلاعاتی مذکور می‌تواند در هر سرویس دهنده مقداری متفاوت باشد اما برخی اطلاعات آن همواره ثابت است. به این بانک اطلاعاتی Resource Records می‌گویند. هر رکورد در این فایل معمولاً دارای ۵ فیلد است:

Domain Name	Time to live	Class	Type	Value
-------------	--------------	-------	------	-------

Domain Name: نام حوزه یا نام مربوط به یک ماشین

Time to live: مدت اعتبار و استناد به رکورد

Class: مشخص کردن شبکه‌ی ماهیت نام نمادین (هر شبکه می‌تواند روش خاص خود را برای تعریف نام نمادین در شبکه محلی خود داشته باشد). اگر رکوردی مربوط به یک نام در شبکه ی اینترنت باشد در این فیلد رشته ی in قرار می‌گیرد.

Type: این فیلد نوع رکورد و معنای آنرا مشخص می‌کند.

برخی از موارد مهمی که در این فیلد قرار می‌گیرند:

A: آدرس IP شبکه

NS: نام سرور (آدرس DNS Server)

CNAME: نام مستعار (یک سایت می‌تواند چندین نام مستعار داشته باشد)



PTR: اشاره گر (IP را گرفته و آدرس دامنه را برمی گرداند).

MX: آدرس های ایمیل سرور

SOA: اطلاعات ابتدایی پیرامون ناحیه ی آدرس نمادین

مثال زیر نمایش یک رکورد از اطلاعات RR است:

```
ns.nic.ddn.mil 99999999 IN A 192.112.36.4
```

پروتکل انتقال فایل (File Transfer Protocol [FTP])

پروتکل FTP توانایی مشاهده دایرکتوری‌ها و مدیریت آنها و همچنین مدیریت دسترسی به فایل‌ها را دارا می‌باشد. این پروتکل از دو پورت استفاده می‌کند. از طریق پورت ۲۱ برای انتقال دستورات و از طریق پورت ۲۰ برای انتقال داده‌ها استفاده می‌نماید.

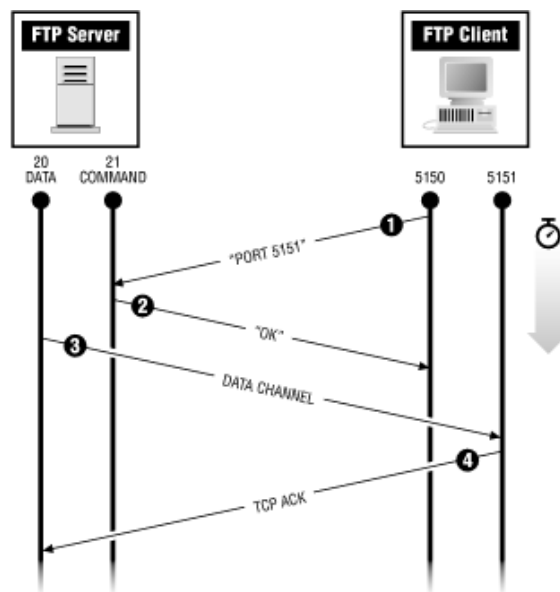
این پروتکل در دو مد *normal* و *passive* اتصال را برقرار می‌کند.

حالت Normal

در این حالت سرور روی پورت ۲۱ آماده‌ی دریافت اطلاعات است. در این حالت، کلاینت به سرور وصل شده و دستور پورت را به سرور ارسال می‌کند به این معنا که بر روی پورت x (عددی بیشتر از ۱۰۲۴ است؛ در مثال زیر ۵۱۵۱ است.) منتظر یک فایل است؛ سرور نیز تایید کرده؛ اکنون سرور پورت ۲۰ را فعال کرده و یک کانال داده به کلاینت می‌فرستد که کلاینت نیز تاییدیه صادر کرده و عملیات مبادله‌ی اطلاعات آغاز می‌شود.

این مد با *Fire wall* مشکل دارد. چرا که کانال ارسال داده از طرف سرور فعال می‌شود و فایروال جلوی این پورت را خواهد گرفت. یک راه مقابله با این مشکل این است که فایروال کار پردازش انجام دهد و پورت و دستورات FTP را شناسایی کند؛ البته این روش مشکلی خواهد داشت و آن این است که فایروال مدت زیادی برای پردازش نیاز دارد و نگهداری دستورات نیز فضای زیادی را می‌طلبد که در صورت وجود دستورات FTP زیاد فایروال دچار مشکل و ازدحام خواهد شد.

شکل زیر نمایانگر اتصال به حالت Normal است:



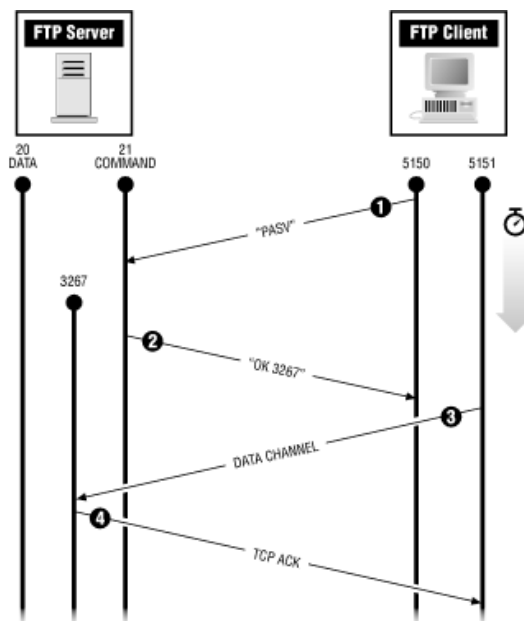
همانطور که گفته شد FTP دو نوع دستور (Command) دارد؛ اولی دستورات کاربری مانند (*dir*, *cd*, ...) و دیگری دستورات پروتکلی هستند که بین سرور و کلاینت رد و بدل می‌شوند. برای مثال دستورات در شکل بالا از نوع پروتکلی هستند.

حالت Passive

برای حل مشکل بیان شده از این مد استفاده می‌شود. در حقیقت این مد بحث اختصاص کانال توسط کلاینت صورت می‌گیرد نه سرور به این شکل مشکلی با فایروال نخواهد داشت.

و تمام کارها نیز روی یک پورت صورت خواهد گرفت! چرا که سرور همراه با تایید برقراری ارتباط پورت را نیز اختصاص می‌دهد و کلاینت کانال ارتباطی را فعال کرده که در نهایت سرور آنرا تایید می‌کند.

شکل زیر نمایش این مد اتصال است:



پروتکل TFTP (Trivial File Transfer Protocol)

این مدل بر خلاف FTP که بر مبنای TCP و کاملاً مطمئن عمل می‌کند؛ بر مبنای UDP پیاده سازی شده است و غیرمطمئن است؛ در حقیقت مدل ساده شده‌ی FTP است.

این مدل برخی از قابلیت‌های FTP را ندارد و به دلیل فعالیت بر اساس پروتکل UDP بسیار سبک‌تر از FTP است. در TFTP عملیاتی نظیر فهرست‌گیری از فایل‌ها و شاخه‌ها، تغییر شاخه‌ی جاری و احراز هویت کاربر امکانپذیر نیست.

از این پروتکل معمولاً در دستگاه‌های سخت‌افزاری که قدرت پردازش کامپیوترهای معمولی را ندارند (مانند ماشین‌هایی که بدون دیسک فعالیت می‌کنند و از ROM بهره می‌برند) استفاده می‌شود.

از آنجا که TFTP بر مبنای UDP است و به دلیل عدم برقراری ارتباط دست‌تکانی سه مرحله‌ای و عدم ثبت IP کاربر امکان سوء استفاده‌ی هکرها را نیز فراهم می‌کند.