به نام خدا

درس برنامه‌سازی پیشرفته

توضیحات شبکه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۳ـ۰۲

استاد:

دکتر محمدامین فضلی

فهرست

مفاهیم شبکه

شبکه چیست؟

شبکه به طور کلی به معنای ارتباط و اشتراک اطلاعات بین چند فرد یا چند شیء است. در زمینه فناوری اطلاعات هم به ارتباط بین چند سیستم کامپیوتری، نرم افزار یا سخت افزار با وسایل ارتباطی مثل کابل یا تجهیزات بیسیم، شبکه می گویند. در واقع شبکه مجموعه‌ای از سیستم‌ها است که از طریق یک کانال ارتباطی می‌توانند داده و منابع را به اشتراک بگذارند. در ادامه، به معرفی مفاهیم مربوط به شبکه می‌پردازیم.

مفاهیم اولیه شبکه و ارتباط

- **Node:** در شبکه‌های کامپیوتری به هر کدام از سیستم‌ها Node هم گفته می‌شود. Nodeها می‌توانند شامل کامپیوترهای شخصی، تلفن‌ها، پرینت سرورها و دیگر سخت افزارهای مرتبط با شبکه باشند.
- **Packet:** به طور کلی در شبکه‌های کامپیوتری برای تبادل اطلاعات، داده‌های بزرگ به قسمت‌های کوچکتر تقسیم می‌شوند و هر یک از این قسمت‌ها را بسته یا packet می‌نامند. هر packet که در شبکه ارسال می‌شود، دارای مشخصات و IP مربوط به مبدا و مقصد است.
- **IP Address:** هر میزبان در شبکه، یک آدرس مخصوص به خود به نام IP Address دارد. این آدرس یکتاست و از ۴ عدد ۸ بیتی تشکیل شده است. وقتی سایتی را در مرورگر جست و جو می‌کنید، درخواستی همراه با IP شما به آن سایت فرستاده می‌شود. سرور آن سایت نیز اطلاعات درخواستی را با کمک IP آدرس شما برایتان ارسال می‌کند.
- **Port**: فرض کنید کامپیوتر شما توسط برنامه‌های (process) مختلفی نیاز به برقراری ارتباط با شبکه دارد. در این صورت IP Address به تنهایی پاسخگو نخواهد بود و از port استفاده می‌شود. پورت عددی است که برای شناسایی process خاصی که قصد دسترسی به شبکه را دارد استفاده می‌شود. در واقع به هر برنامه (process) که منتظر دریافت پیام از شبکه است (listening)، یک پورت نسبت داده می‌شود که با استفاده از آن، برنامه مقصد به صورت یکتا در کامپیوتر مشخص می‌شود.
پورت و IP آدرس را می شود مانند شماره تلفن در نظر گرفت که IP آدرس کد شهر یا کشور است و پورت، باقی شماره تلفن در نظر گرفته می شود. کد شهر یا IP آدرس جهت شناسایی محدوده و منطقه تماس به کار می رود و قسمت باقی‌مانده یا پورت، شماره اختصاصی و یکتایی است که تماس با آن برقرار می‌شود.
- **Socket:** به شکل ساده، سوکت ترکیبی از پورت و IP آدرس است. یک سوکت شامل هر دو گروه آدرس آی‌پی میزبان و پورت مربوط به یک برنامه است که با یک علامت جداکننده این دو مقدار از یکدیگر جدا شده اند (مثلا 172.0.0.1:8000).
به تعبیر تخصصی‌تر، سوکت نقطه انتهایی یک ارتباط دوطرفه بین دو برنامه در حال اجرا در شبکه است. سوکت به یک عدد پورت متصل می شود تا لایه TCP شبکه بتواند برنامه مورد نظر برای ارسال اطلاعات را تشخیص دهد. در مثال شماره تلفن، سوکت مانند گوشی تلفن است. به این شکل که شماره مورد نظر و کد ناحیه را در گوشی وارد کرده و تماس را برقرار می‌کنید. زمانی که تماس پاسخ داده می‌شود، در واقع یک کانال ارتباطی بین شما و فردی که با او تماس گرفته‌اید، ایجاد می‌شود؛ به تعبیر ساده‌تر،

کار سوکت ایجاد این کانال است. از طریق کانال ارتباطی ایجاد شده توسط سوکت، داده‌هایی در طول شبکه ارسال و دریافت میشوند.
به شکل ساده‌تر می‌توان socket را جایی در نظر گفت بسته‌هایی که از شبکه برای یک برنامه آمده‌اند، در آن قرار گرفته و سپس آن برنامه می‌تواند این بسته‌ها را از socket خود بردارد.

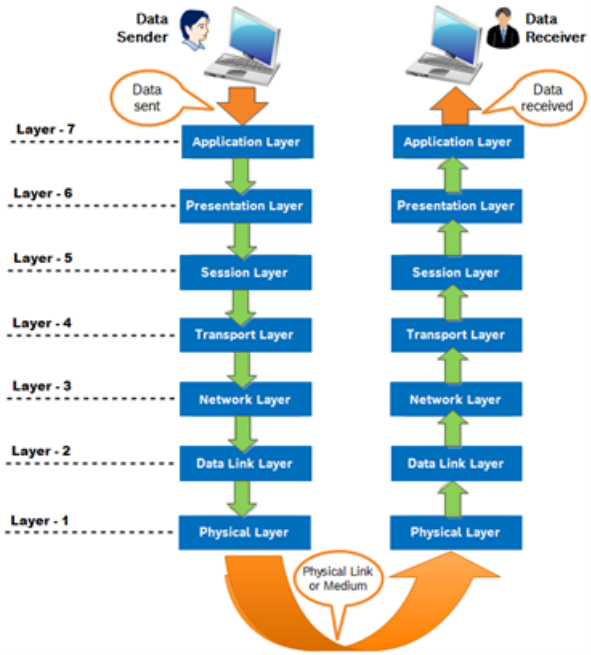
- **Protocol:** کامپیوترها برای تبادل اطلاعات باید به توافق اولیه برسند که چطور اطلاعات ساختاربندی شود و هر طرف چطور و چه مقدار داده ارسال و دریافت می‌کند. تصور کنید کامپیوتری بسته ۸ بیتی می‌فرستد در حالی که کامپیوتر مقصد منتظر بسته ۱۶ بیتی است! پروتکل‌ها قوانین و استانداردهایی هستند که جهت جلوگیری از بروز چنین مشکلاتی در ایجاد ارتباط بین دو کامپیوتر در شبکه به کار می‌روند.

لایه‌های شبکه

فرایند ارتباط بین دو Node در شبکه را در نظر بگیرید. مهندسی این ارتباط بسیار پیچیده است. برای سهولت، کل فرایند شبکه به وظایف کوچک تقسیم شده است. هر کار کوچک به یک لایه خاص اختصاص داده شده و هر لایه طوری کار می کند که تنها همان کار خاص را انجام دهد. هر لایه داده‌های ورودی را گرفته و داده‌های خروجی را به لایه بعدی ارسال می‌کند.

مدل OSI یا Open System Interconnection یك مدل مرجع براي ارتباط بين دو كامپيوتر است. طبق این مدل، ارتباط فرستادن و دریافت کردن پیام هر یک به هفت لایه تقسیم می‌شوند.

زمانی که این لایه ها شروع به کار می‌کنند و از کامپیوتر مبدا می‌خواهند ارسال شوند لایه‌ها از بالا به پایین یعنی از لایه Application تشکیل شده و به لایه فیزیکی برای ارسال می‌رسند. اما در کامپیوتر مقصد دقیقا برعکس این موضوع است. از لایه فیزیکی به مقصد رسیده و در نهایت به لایه Application ختم می‌شود.



در ادامه به طور مختصر کارکرد لایه‌ها توضیح داده شده است:

۱. **Application Layer:** این لایه رابط بین کاربر و سیستم عامل محسوب می شود و همانطور که از اسمش پیداست، شما به وسیله این لایه با نرم‌افزارهای کاربردی ارتباط برقرار می‌کنید. در این جا

داده‌هایی که باید در شبکه منتقل شوند تولید می‌شوند. پروتکل‌های لایه اپلیکیشن شامل دستوراتی برای اپلیکیشن‌های خاص هستند مثل HTTP و IMAP و .FTP مرورگرهای وب از https برای دانلود امن محتوا از وب سرور استفاده می‌کنند.

۲. **Presentation Layer (Translation Layer):** این لایه مانند مترجم داده‌های لایه application را به فرمتی که برای انتقال در شبکه مورد نیاز است تبدیل می‌کند. به عنوان مثال تعیین می‌شود که اطلاعات چگونه رمزنگاری شوند، فشرده‌سازی شوند و قالب‌بندی شوند.

۳. **Session Layer:** این لایه مسئول پایه گذاری ارتباط (connection)، نگهداری از نشست‌ها (session)، احراز هویت (authentication) و مسائل امنیتی است.

۴. **Transport Layer:** همان طور که می‌دانیم، در کامپیوتر ما چندین برنامه در حال ارتباط با شبکه هستند. پس اگر برنامه اول (از کامپیوتر اول) پیامی را به سمت کامپیوتر دوم می‌فرستد، این کامپیوتر باید به نحوی بفهمد که این پیام برای کدام برنامه است. این عملیات در لایه‌ی انتقال یا Transport پیاده سازی شده است. پروتکل‌های لایه انتقال، مشخص می‌کنند بسته‌ها چگونه ارسال و دریافت و تایید می‌شوند مانند TCP و UDP.

۵. **Network Layer:** لایه Network مسیر انتقال داده از یک میزبان (host) به دیگری را پیدا می‌کند. در واقع لایه‌ی network مسئول مسیریابی بسته‌ها یا packet routing بر اساس آدرس مبدا و مقصد هر packet است. از پروتکل‌های مربوط به این لایه می‌توان به IPv4 و IPv6 اشاره کرد.

۶. **Data Link Layer:** وظیفه این لایه آدرس‌دهی فیزیکی و ایجاد ارتباط مطمئن و بدون مشکل بین نودها است. این لایه مکانیزم‌های مختلف برای مقابله با collision ها و error ها دارد.

۷. **Physical Layer:** این لایه مرتبط با سخت افزار، کابل کشی سیمی، توان خروجی، تعداد پالس‌ها و غیره است و وظیفه انتقال سیگنال را بر عهده داشته و به محتویات هیچ کاری ندارد. سخت افزار موجود در این لایه از جنس انواع کابل ها، کارت شبکه و هاب است.

معماری‌های شبکه

معماری شبکه در واقع مجموعه‌ای از قوانین و متودهایی برای طراحی منطقی و فیزیکی نرم افزار، سخت افزار، پروتکل‌ها و انتقال داده در شبکه است. دو مورد از معروف‌ترین معماری‌ها عبارتند از:

- **Server/Client:** در این معماری یک سرور مسئول پاسخ‌گویی به درخواست‌های(requests) کاربران یا کلاینت‌ها می‌باشد و با دریافت درخواست‌ها، پاسخ‌های (response) مناسب را به کاربران برمی‌گرداند. همچنین سرورها برنامه‌های خاصی را اجرا کرده و منابع خود را با کاربران به اشتراک می‌گذارند. سرور به منابع و اطلاعات اصلی برنامه دسترسی دارد و پردازش‌های اصلی داده ها در آن انجام شده و در نهایت نتیجه به شکل مناسبی به کلاینت اطلاع داده می‌شود.

Client-Servers Network Model

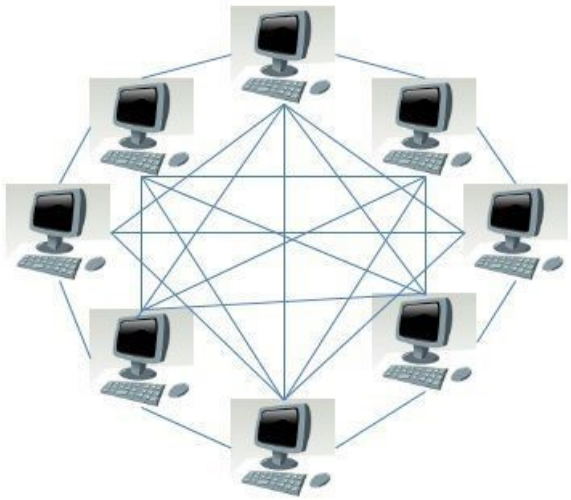
- **Peer-to-Peer (P2P):** نقطه مقابل معماری Client-Server را می‌توان معماری Peer To Peer یا نظیر به نظیر دانست. در معماری P2P، دو یا چند کامپیوتر منابع خود را در قالب یک سیستم غیرمتمرکز به اشتراک می‌گذارند. در این سیستم دیگر ساختار سلسله مراتبی وجود ندارد. تمامی Nodeهای موجود، منابع خود را در اختیار دیگر Nodeها قرار می‌دهند. به همین دلیل هیچ Nodeای به Node دیگر ارجحیت ندارد. به هر Node در این شبکه، Peer می‌گویند. در واقع هر Node می‌تواند هم در نقش سرور (تامین کننده) و هم در نقش کلاینت (مصرف کننده) ظاهر شود.

یکی از مزیت‌های اصلی شبکه‌های P2P به نسبت شبکه‌های Client-Server، این است که در ساختار Client-Server اگر سرور دچار مشکل شود، کل خدمت‌رسانی دچار مشکل می‌شود. ولی در شبکه‌های P2P اگر یک سیستم خراب شود، به‌راحتی سیستم دیگری را می‌توان جایگزین آن کرد. به اصطلاح می‌توان گفت که در معماری Client-Server، Single point of failure داریم.

هم‌چنین برای گسترش شبکه Client-Server، باید هزینه زیادی را صرف ارتقا سرور کنیم. ولی این کار در شبکه P2P به دلیل غیرمتمرکز بودن آن، به راحتی قابل انجام است.

از طرف دیگر در سیستم client-server یک سرور فایل اختصاصی سطوح دسترسی متفاوتی را برای کلاینت‌ها فراهم می‌کند که به نسبت سیستم‌های peer to peer که امنیت توسط کاربر نهایی اداره می‌شود از امنیت بیشتری برخوردار است. همچنین شبکه‌های peer to peer با افزایش تعداد نودها دچار اختلال در سطح عملکرد می‌شوند، اما سیستم‌های client-server باثبات تر هستند.

در نهایت می‌توان گفت که گسترش شبکه‌های P2P هزینه بیشتری دارد چون با اضافه کردن هر نود باید آن را با کل شبکه هماهنگ کنیم اما در معماری Server-Client این کار صرفا با متصل کردن Client جدید به سرور ممکن می‌شود و هزینه کمتری نسبت به P2P دارد.

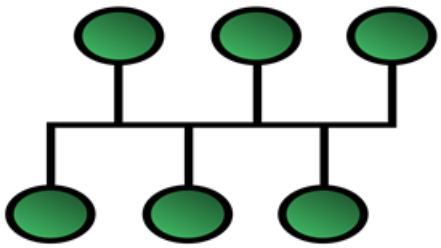


Peer-to-Peer Network Model

توپولوژی‌های شبکه

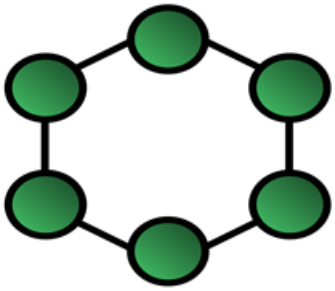
توپولوژی‌های شبکه ارتباط اجزای موجود در شبکه و ارتباط آن‌ها را با همدیگر نشان می‌دهد. در این بخش به بررسی برخی توپولوژی‌های موجود می‌پردازیم:

- **Bus Topology:** در این توپولوژی، هر Node به یک کابل متصل می‌باشد و داده‌ها در یک جهت منتقل می‌شوند و به Node های دیگر می‌رسند. از مزایای این روش می‌توان به مقرون به‌صرفه و ساده بودن پیاده سازی آن اشاره کرد. در مقابل، از کار افتادن کل سیستم در صورت قطع شدن کابل اصلی و کندتر بودن شبکه به نسبت توپولوژی‌های دیگر اشاره کرد.

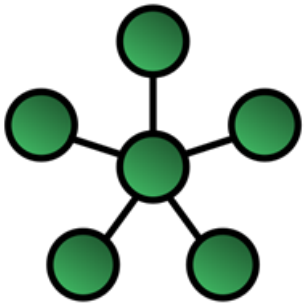


- **Ring Topology:** در این توپولوژی، هر Node دقیقا به دو Node دیگر وصل می‌شود و ساختاری حلقه‌ای پدید می‌آورد. در این توپولوژی یک توکن در حال حرکت است و هر Nodeای که توکن را در اختیار داشته باشد قادر به ارسال پیام است. پیام‌ها در این توپولوژی در یک جهت در حال حرکت هستند و شبکه ابتدا و انتهایی ندارد و اطلاعات در یک حلقه منتقل می‌شوند. همچنین به دلیل اینکه اطلاعات از همه هاNode در حال عبور هستند، از repeaterها استفاده می‌شود که مانع از دست رفتن اطلاعات در این تبادلات طولانی باشد.
از معایب این روش می‌توان به حساس بودن آن به کارکرد درست همه Node هاست. از آنجایی که داده‌ها از همه هاNode عبور می‌کنند، ضروری است که همه Nodeها اطلاعات را به درستی دریافت و سپس ارسال کنند و در غیر اینصورت کل شبکه از کار می‌افتد. همچنین عیب‌یابی در این توپولوژی هم سخت‌تر از دیگر توپولوژی‌هاست.

از طرفی راحتی نصب و نگه‌داری این شبکه و همچنین امنیت اطلاعات در شبکه‌هایی با حجم بالای داده به دلیل استفاده از توکن از مزایای این روش به حساب می‌آیند.



- **Star Topology:** در این توپولوژی نیز، تمام هاNode به صورت جداگانه به یک Hub متصل می‌شوند. Hubها اجزایی در شبکه هستند که داده‌ها را به همه Nodeهای متصل به خودشان ارسال می‌کنند. همچنین خود Hub به‌عنوان repeater نیز عمل می‌کند. از مزایای این روش نیز می‌توان به عیب‌یابی ساده‌تر به دلیل متمرکز بودن سیستم، عملکرد سریع شبکه با Nodeهای کم و وابسته نبودن شبکه به عملکرد درست همه Nodeها می‌باشد. همچنین می‌توان به وابسته بودن شبکه به عملکرد درست Hub و هزینه‌بالای پیاده سازی آن به عنوان معایب آن اشاره کرد.

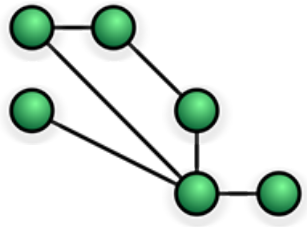


- **Mesh Topology:** در این توپولوژی، هر Node به مانند یک Router عمل می‌کند و با اتصالاتی که به دیگر Nodeها دارد اطلاعات را در شبکه منتقل می کند. توپولوژی Mesh بسته به اینکه Nodeها به چند Node دیگر وصل شده‌اند به ۲ نوع تقسیم می‌شوند:
  - Partial: در این توپولوژی، Nodeها به همه Nodeهای دیگر متصل نیستند.
  - Full: برخلاف partial، هر Node به همه Nodeهای دیگر وصل شده است.

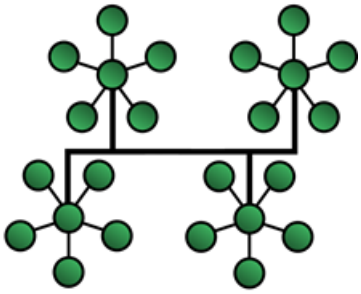
  همچنین بسته به منطق انتقال داده نیز می‌توان دو دسته‌بندی زیر را ارائه داد:

  - Rountig: دستگاه ها بر اساس نیاز شبکه، یک منطق مسیریابی (routing) خواهند داشت. به عنوان مثال یکی از منطق های مسیریابی می تواند انتخاب بر اساس کوتاه ترین مسیر باشد
  - Flooding: اطلاعات یکسان به تمامی دستگاه های شبکه منتقل می شود، بنابراین به هیچ مسیر یابی منطقی نیاز نیست

  در این توپولوژی، امنیت داده‌ها و Nodeها بالاست و همچنینرعیب‌یابی سیستم نیز ساده است. درمقابل، به دلیل تعداد زیاد ارتباطات، هزینه و حجم کابل‌ها زیاد است.

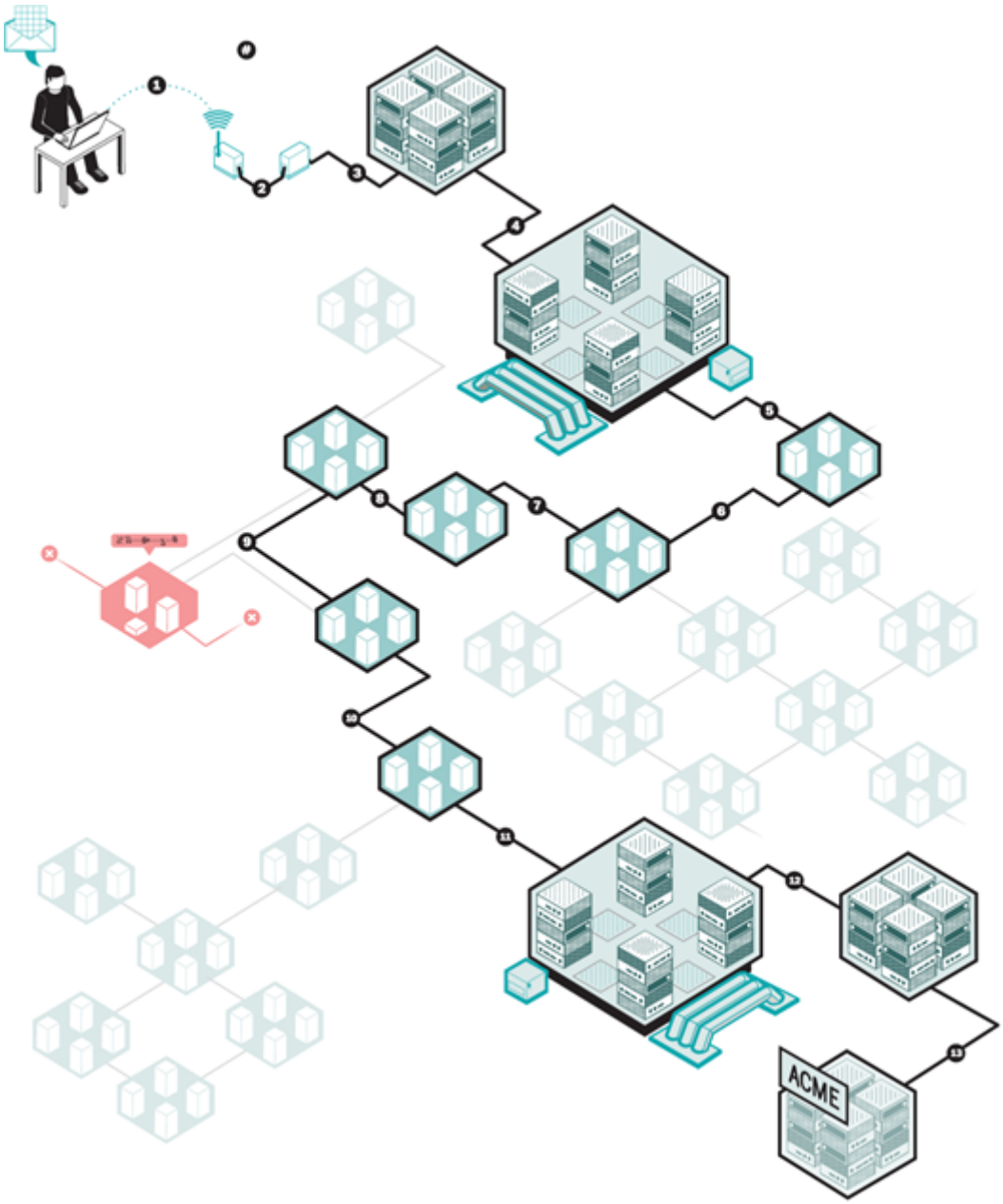


- **Tree Topology:** این توپولوژی یکی از بهترین انتخاب‌ها برای شبکه‌های بزرگ است. در این توپولوژی Nodeها به‌صورت سلسله مراتبی به یکدیگر وصل شده‌اند و در راس آن‌ها، یک دستگاه مرکزی وجود دارد. این شبکه حداقل باید ۳ سطح داشته باشد.
مدیریت و نگه‌داری این شبکه بسیار ساده است و همچنین اضافه کردن Node جدید به شبکه نیز بدون دردسر است. در مقابل، این سیستم وابسته به دستگاه مرکزی است و همچنین حجم کابل استفاده شده نیز بالاست.



اینترنت به عنوان یک شبکه و جابه‌جایی اطلاعات در اینترنت

اطلاعات برای انتقال در بستر اینترنت، ابتدا به بسته‌های کوچکی به نام Packet تقسیم می‌شوند. این بسته‌ها طی یک سفر طولانی به مقصد می‌رسند و دوباره با هم جمع شده و پیام اصلی را تشکیل می‌دهند. ابتدا پکت‌ها با وارد شدن به routerها و switchهایی که به ارسال کننده پیام آن متصل است، وارد شبکه ISP (Internet Service Provider) می‌شوند. شبکه‌ای که ISP ها باهمدیگر تشکیل می‌دهند، یک شبکه تقریبا بزرگ است که اکثر پیام‌ها در این شبکه قرار دارند. پکت‌ها با وارد شدن در این شبکه با توجه به دیتایی که در Header خود دارند، مسیریابی می‌شوند. الگوریتم‌های مسیریابی باید پکت‌ها را در کمترین زمان به مقصد برسانند. درنهایت، پکت‌ها از شبکه ISP ها خارج شده و با ورود به routerها، به دستگاه مقصد که متصل به آن روتر است فرستاده می‌شود. در نهایت، یک Acknowledgment از گیرنده به فرستنده پیام ارسال می‌شود که نشان دهد پیام به درستی به گیرنده رسیده است. شکل زیر یک شمای کلی از مسیری که پکت طی می‌کند را نشان می‌دهد. (این لینک)


ACME