- شبکه ها در core اینترنت ، سوییچ ها و روتر ها رو با لینک ها به هم متصل می کنن.وظیفه ی اصلی شون هم ایجاد ارتباط بین شبکه های محلی یا شبکه های دسترسی به یکدیگر هست.
- از لحاظ عملکرد شبکه هایی که در core اینترنت هستن به دو دسته تقسیم میشن : circuit switching و packet switching
- در packet switching ، اون end system یا host ، اون پیامی رو که از اپلیکشن (برنامه ی کاربردی) دریافت می کنه و به جای اینکه این پیام رو مستقیم به داخل شبکه تزریق کنه،اون ها رو در قالب یه سری بسته می شکنه و به شبکه می فرسته.
- هر بسته(packet) علاوه بر اینکه قسمتی از پیام(payload) رو داخل خودش داره ، یه سری اطلاعات کنترلی(header) رو هم حمل می کنه.

  آدرس فرستنده و گیرنده هم داخل این header بسته ها وجود داره.

  بعد شبکه ای که این بسته ها رو دریافت می کنه ، میاد این بسته ها رو مبتنی بر آدرسی که دارن از روتر به روتر و از لینک به لینک هدایت می کنه تا اون بسته به مقصد برسه.
- توی مثال packet switching ۶ تا روتر داریم که با ترتیب خاصی به هم وصل شدن.ما یه الگوریتمی داریم به اسم routing algorithm و یه جدولی داریم به اسم forwarding table که به صورت محلی عمل میکنه و هر بسته ای که روتر دریافت می کنه با توجه به آدرس

مقصدی که توی header اون بسته وجود داره ، تعیین می کنه که این بسته باید به کدوم یک از پورت های خروجی روتر ارسال بشه.
اما سوالی که هست اینه که مقادیر این جدول از کجا میاد؟
اگه ابعاد شبکه کوچیک باشه و شبکه هم مدام در حال تغییر نباشه میشه این کارو به صورت دستی انجام داد ، به این شکل که بین هر فرستنده و گیرنده یه مسیر به صورت استاتیک تعیین کرد و بعد جدول هایی که داخل روتر ها هستن رو جوری مقدار دهی کرد که اگه یه سورسی برای یه مقصدی بسته ای ارسال می کنه این فورواردینگ نهایتا این بسته رو به مقصد برسونه.
اما نکته اینه که شبکه ها معمولا بزرگن و داینامیک زیادی دارن.
داینامیک زیاد به خاطر خرابی لینک ها و روتر ها و سوییچ ها میتونه باشه ، به دلیل لود شبکه و ترافیک شبکه هم میتونه باشه. این قضیه باعث میشه که تعیین مقادیر جدول فورواردینگ به صورت دستی امکان پذیر نباشه و ما از یه سری الگوریتم های توزیع شده(پروتکل) استفاده می کنیم .این پروتکل های routing ، از طریق رد و بدل پیام های کنترلی میتونن مقادیر جدول های فورواردینگ رو به طور بهینه تعیین کنن ، مسیر یابی رو بین مبدا و مقصد های مختلف انجام بدن و بعد متناظر با این مسیریابی بهینه ، مقادیر جدول فورواردینگ برای هر روتر تعیین میشه.

- اگه می خواستیم این کارو به صورت متمرکز (centralized) انجام بدیم ینی مثلا یه سروری در شبکه باشه که مسئول مسیریابیه، و همه اطلاعاتشونو به اونجا می فرستن ، مسئله راحت تر میشه از نوع گرافی ، که بهترین مسیر بین دو نقطه رو می خواستیم انتخاب کنیم. اما پیچیدگی به خاطر اینه که ما میخوایم این کارو توزیع شده انجام بدیم.
- نمونه ی این کار توی حمل و نقل ما به طور روزمره هم دیده میشه. مثلا وقتی بخوایم از یه شهر به یه شهر دیگه سفر کنیم ، میخوایم بهترین مسیر رو پیدا کنیم(مثلا کم ترافیک ترین مسیر). توی هر منطقه بین تمام خروجی هایی که هست بهترینش رو انتخاب می کنیم و پیش میریم . بنابراین هم forwarding داریم و هم routing و مسیریابی!

- یکی دیگه از ویژگی های شبکه های مبتنی بر packet switching ، اینه که هیچ گونه رزرو کردن منابع قبل از اینکه ما داده هامون رو به شبکه بفرستیم صورت نمی گیره .
این منابع میتونن تعیین مسیر end-to-end ، و همچنین اختصاص مسیری از پهنای باند لینک ها به یک ارتباطی باشه که میخواد صورت بگیره. ولی توی packet switching بلافاصله بعد از اینکه بسته آماده شد به شبکه ارسال میشه.(بدون هیچ هماهنگی)

- مثال این قضیه میتونه تلفن کردن به یه شماره باشه. مثلا وقتی یه شماره ای رو داخل شماره گیر وارد می کنیم و تماس می گیریم، اول شبکه بررسی می کنه و می بینه که آیا منابع کافی برای برقراری این ارتباط وجود داره یا نه ، و اگه بود تماس برقرار میشه. در واقع به این signaling اولیه ، call setup میگن و شبکه های مبتنی بر packet switching ، call setup ندارن.
- مزیت نداشتن reservation : باعث میشه منابع به طور مشترک استفاده بشن و بهره برداری از منابع شبکه افزایش پیدا می کنه(جنس ترافیک bursty هست).به این قضیه statistic multiplexing نام داره. به طور کلی ینی یه چیزی که خواهان زیادی داره رو یا قسمت قسمت کنیم و در اختیار مشترکین قرار بدیم ، یا اینکه اگه براساس آماری زمان استفاده ی مشترکین از این سرویس sync نیست ، کلش رو در اختیار مشترک ها قرار بدیم. یه نمونه اش تلفن عمومی توی خیابون هاست!

تعریفش توی شبکه های کامپیوتری اینه که مثلا دوتا clientداریم که برای استفاده از اینترنت باید 1.5Mbps بیت ریت داشته باشه، و اگه بخوایم منابع رو به این دوتا کلاینت اختصاص بدیم ، اگه کلاینت A بخواد بسته بفرسته همواره نیازی نیست که از بیت ریت 1.5Mbps بهره مند بشه و همینطور کلاینت B، و چون اینا sync نیستن وقتی کلاینت اولی میخواد استفاده کنه احتمال اینکه کلاینت

دومی هم بخواد استفاده کنه کم هست و ما با قرار دادن یه سری بافر حالت های همزمان استفاده ی این دو رو پوشش میدیم.

خوبی این قضیه اینه که یه لینک Mbps۱.۵ به جای Mbps۳ برای ما کفایت می کنه.

- بدی نداشتن reservation : چون ما مسیری ست نمی کنیم و منبعی رو کنار نمیذاریم ، هرکدوم از بسته هایی که به شبکه ها ارسال می کنیم مستقلا باهاشون رفتار میشه و توی هر روتری با توجه به اون آدرس مقصدی که دارن مسیریابی میشن و این یه سری چالش هایی به دنبال داره :

این بسته ها ممکنه از مسیر های مختلفی ممکنه نهایتا به مقصد برسن و این مسیری که طی میکنن(از مبدا تا مقصد) متفاوت باشه.

(مثال: اساس کشی با دوتا کامیون با راننده های متفاوت که از مسیرای مختلف میرن!)

- مثال در شبکه های کامپیوتری: (۴ تا اند سیستم و یه پکت سوییچ)

اشکال وجود این مسیرای مختلف ، delay های مختلف و عوض شدن ترتیب هاست.

همچنین ممکنه بسته ها گم بشن(lost).چون منابع رزرو نمی کنیم ، در داخل روتر ها قبل از اینکه بسته ی ما برسه ممکنه بسته های دیگه در اون وجود داشته باشه و همون موقع بهش سرویس داده نمیشه ، ما از بافر استفاده می کنیم که تا یه حدی بسته ها بتونن

منتظر بمونن اما اگه تعداد بسته ها از گنجایش بافر بیشتر بشن ، طبق یه مکانیزم خاصی توسط روتر drop میشن(دور ریخته میشن). و به همین خاطره که معماری پکت سوییچ رو بهش میگن BEST EFFORT ! ینی در میزان تاخیر بسته ها و نه در رسیدن بسته ها هیچ تضمینی نیست و ممکنه بسته ها گم بشن!

- مثلا در packet switching سرویس های صوت و تصویر ارائه نمیشه چون هیچ تضمینی برای تاخیر نداشتن نیست!
  و اینکه چجوری در بستر packet switching بتونیم صوت و تصویر(به طور کلی اپلیکشین هایreal-time) ارائه کنیم و رفتاری مثل circuit switch داشته باشیم، هنوز یه مسئله ی حل نشده هست!

- در داخل شبکه های کامپیوتری مسیر های مختلف میتونن تاخیرای مختلف داشته باشن.تاخیر مولفه های مختلفی داره :

1 - تاخیر در انتشار امواج الکترو مغناطیس در لینک که به طول لینک و جنس لینک بستگی داره.

2 - تاخیر ارسال ، که بستگی به ظرفیت لینک داره

3 - تاخیر صف که در داخل روتر ها و سوییچ ها رخ میده و ناشی از تاخیر در ارسال و استفاده همزمان بیش از یک کاربر از شبکه هست.

- تاخیر ارسال : کاری که روتر انجام میده ، store & forward عه ، یعنی اول باید کل بسته ها رو دریافت کنه و بعد فوروارد کنه.

دلیل اینکه اگه روتر یه بیت دریافت کنه ، نمیتونه همون موقع ارسالش کنه اینه که ما توی packet switch اول باید آدرس رو ببینم چیه و تحلیلش کنیم و ببینیم خطایی وجود داره یا نه ، و بعد مبادرت به ارسال بسته بکنیم . بعد که مشخص شد توسط چه لینکی باید ارسال بشه ، ظرفیت اون لینک مشخص می کنه که چقد طول می کشه تا اون بسته ارسال بشه. مثلا اگر طول packet ، L بیت باشه و ظرفیت لینک R، B/s باشه ، L/R ثانیه طول می کشه تا اون بسته روتر رو ترک کنه و توسط لینک ارسال بشه . به این میگن تاخیر ارسال!

- به طور کلی اگه نرخ سرویس دهی(ریت خروجی) کمتر از نرخ تقاضا(ریت ورودی) باشه صف ایجاد میشه. اگه ریت ورودی از ظرفیت لینک بیشتر بشه ورودی های جدیدی که به روتر ارسال میشن دور ریخته میشن !

  این نوع تاخیر خیلی وابسته به میزان ترافیک شبکه هست و چون ممکنه از مسیرای مختلف به روتر برسن و مسیرای مختلف تاخیرای مختلف دارن، تاخیرای صف هم ممکنه متفاوت باشن!
- روش دیگه ای که برای ارتباط بین دستگاه ها در شبکه های کامپیوتری ازش استفاده میشه روش circuit switching هست.

- برخلاف پکت سوییچ ، منابع به صورت end-to-end برای یه ارتباطی که میخواد شکل بگیره رزرو میشه و این منابع به صورت انحصاری در اختیار اون ارتباط قرار میگیره.

خوبی این انحصاری بودن اینه که اگه منابع موجود بود و ارتباط شکل گرفت، ترافیک شبکه روی پرفورمنس اون ارتباط تاثیری نمیذاره.

بدی این انحصاری بودن اینه که از منابع خوب استفاده نمیشه و در داخل اون ارتباط پیام های زیادی رد و بدل نشه اما چون همچنان در جریان هست ، ما نمیتونیم از منابعی که به اون جریان اختصاص داده شده استفاده کنیم!

این نوع نحوه ارتباط بین دستگاه ها، در شبکه های تلفن روش اصلی هست.در داخل اینترنت هم شبکه های مبتنی بر circuit switch داریم.

- ما میتونیم به دو روش منابع یک لینک رو تقسیم بندی کنیم و در اختیار ارتباطات مختلف قرار بدیم :

۱-Frequency Division Multiplexing(FDM) : توی این روش، پهنای باندی که لینک داره ، به یه سری sub-bond یا کانال تقسیم میشه و هرکدوم ازین کانال ها در اختیار یک ارتباط قرار می گیره توی کل مدت زمان.

۲-Time Division Multiplexing(TDM) : ما منابع لینک رو توی حوزه ی زمان تقسیم می کنیم توی این روش.میایم یه فریمی

رو تعریف می کنیم که هر فریم تشکیل شده از یک سری slot ، و هر slot رو به یک کاربر اختصاص میدیم.(هر کاربر در مدت زمان محدود ، که این زمان بین کاربر های مختلف می چرخه.)

- این تقسیم بندی لینک ، توی هر دوتا روش دیده میشه.
- مزایا و معایب packet & circuit دوباره در اسلاید ۵۵ اورده شده!

- ساختار اینترنت :

- همونطور که گفتیم host ها یا end system ها توسط شبکه های دسترسی به ISP ها متصل میشن و خود ISP ها هم به هم متصل میشن تا end system هایی که در جاهای مختلف هستن بتونن برای همدیگه بسته ارسال کنن و باهم در ارتباط باشن.
  نتیجه یه شبکه یه پیچیده – ینی اینترنت – هست ! و در تکامل و نحوه ی اتصال شبکه هایی که در اینترنت هستند ، علاوه بر مولفه های فنی، مولفه های اقتصادی و سیاست های کشوری هم لحاظ شده!
- به صورت اولیه ، end system ها توسط شبکه های دسترسی به هم متصل میشن. یه روش ارتباط شبکه های دسترسی اینه که بین هر دوتایی ازون ها یه لینک وجود داشته باشه. به عبارتی یه گراف کامل داشته باشیم که نود ها access network ها هستن.

ولی این ساختار مقیاس پذیر نیست! ینی اگه شبکه های دسترسی زیاد باشن ، حجم لینک هایی(سیم کشی ها) که این شبکه ها رو به هم وصل می کنن خیلی زیاد میشه و قابل پیاده سازی نیست.
به جای این کار، یه global ISP ایجاد می کنیم و شبکه های دسترسی مختلف به روتر های این global ISP متصل میشن تا بتونن با هم در ارتباط باشن.
البته فقط یک global ISP نداریم و چندین شرکت وجود دارن که این سرویس رو ارائه میدن و شبکه های دسترسی مختلف هم این آزادی عمل رو دارن که هر کدوم ازین providerها و ISP هایی رو که میخوان انتخاب کنن.

- به این ISP هایی که scope شون گلوبال هست میگن ISP های tier1 .
- یه ساختار دیگه ای داخل اینترنت وجود داره به اسم IXP(Internet Exchange Point) که برای ایجاد ارتباط بین ISP ها ساخته شده. ISPها یا میتونن از لینک های مستقیم برای ارتباط بین هم استفاده کنن یا از یه point of present (مثلا روتر) استفاده کنن تا بدون واسطه ی هیچ شبکه یا چیز دیگه ای به هم متصل بشن.
- وقتی دوتا ISP هم رده، به هم متصل بشن و بابت سرویسی که به هم میدن از هم پول نگیرن ، به این ارتباط peering link گفته میشه.

- یه سری دیگه از ISP ها هستن به اسم regional ISP(tier2) که بین المللی نیستن و فقط یه منطقه یا کشور خاصی رو پوشش میدن. معمولا شبکه های دسترسی هم به این ISP ها متصل میشن و بعد این ISP ها به ISP های tier1 که بین المللی هستن متصل میشن. (البته اینا قواعد کلی هستن وگرنه در اینترنت هر شبکه ای به هر شبکه ی دیگه ای میتونه متصل بشه!)
- یه سری شبکه داریم مخصوص بعضی شرکت ها مثل گوگل ، فیسبوک ، مایکروسافت ، Akamai و ... که در زمینه محتوا و اپلیکیشن های اینترنت فعالیت می کنن به نام CDN(Content Delivery Network) . این شرکت ها برای اینکه بتونن سرویس های خودشون رو مناسب تر از بابت تاخیر و ارزون تر ارائه بدن ، دیتاسنترهاشون جاهای مختلف زمین هست و برای ارتباط بین دیتا سنتر هاشون از یه شبکه خصوصی استفاده می کنن. مثلا گوگل ۱۹ تا دیتا سنتر در قاره ها و کشور های مختلف داره و نتورک مخصوص خودشو داره و ترافیک این نتورک فقط مخصوص این دیتاسنتر ها هست.

این نتورک ها میتونن به بقیه ی ISP ها متصب بشن ، البته ترجیح این هست که به ISP های tier1 متصل نشن، هم بابت تاخیر هم به دلیل هزینه های زیاد ، واسه همین اگه ممکن باشه ، به ISP های tier2(regional ISP) ها متصل میشن.