

Chapter 6

The Link Layer and LANs

A note on the use of these PowerPoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part.

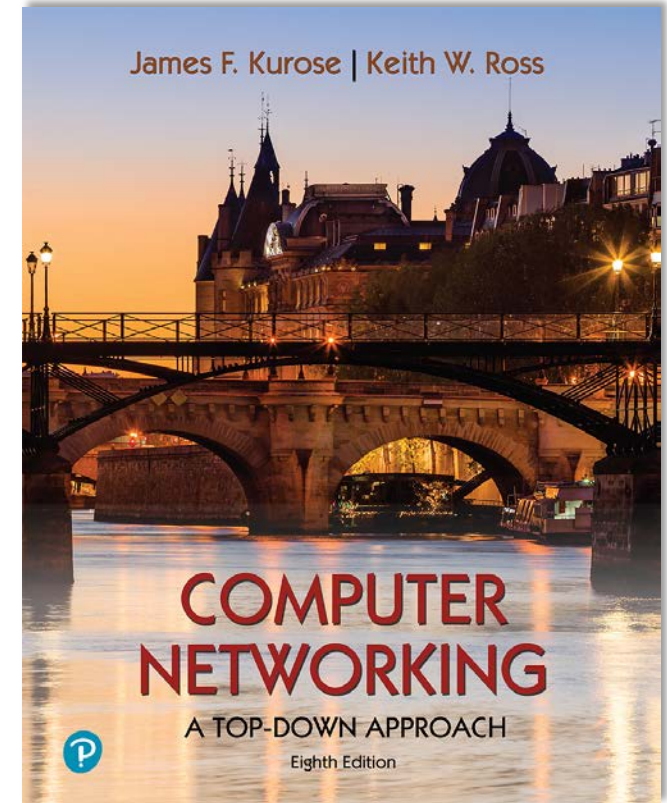
In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2020
J.F Kurose and K.W. Ross, All Rights Reserved



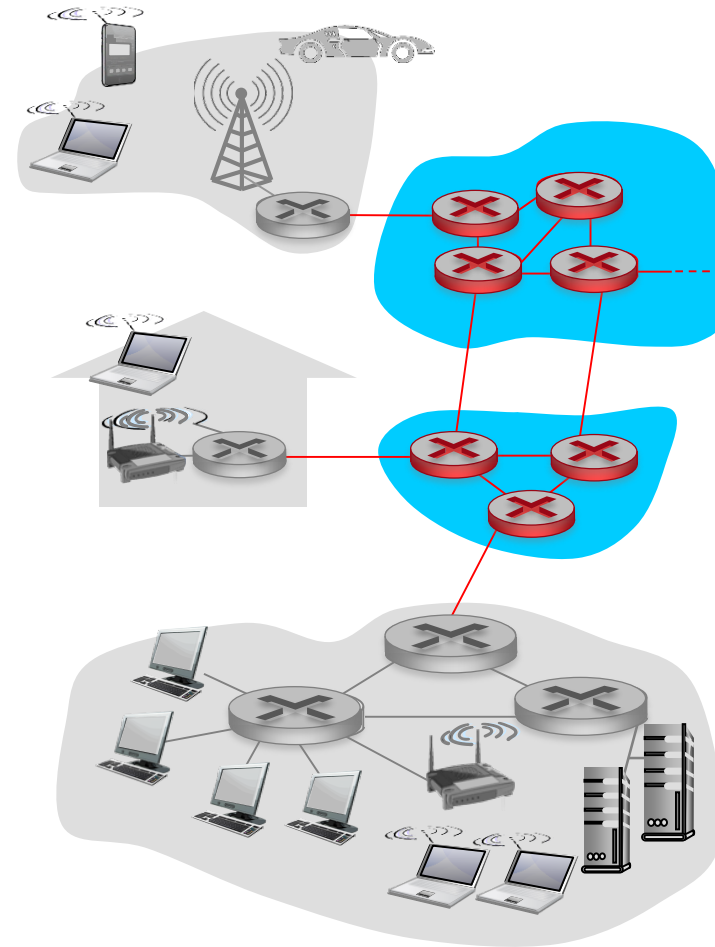
Computer Networking: A Top-Down Approach

8th edition

Jim Kurose, Keith Ross
Pearson, 2020

The network core

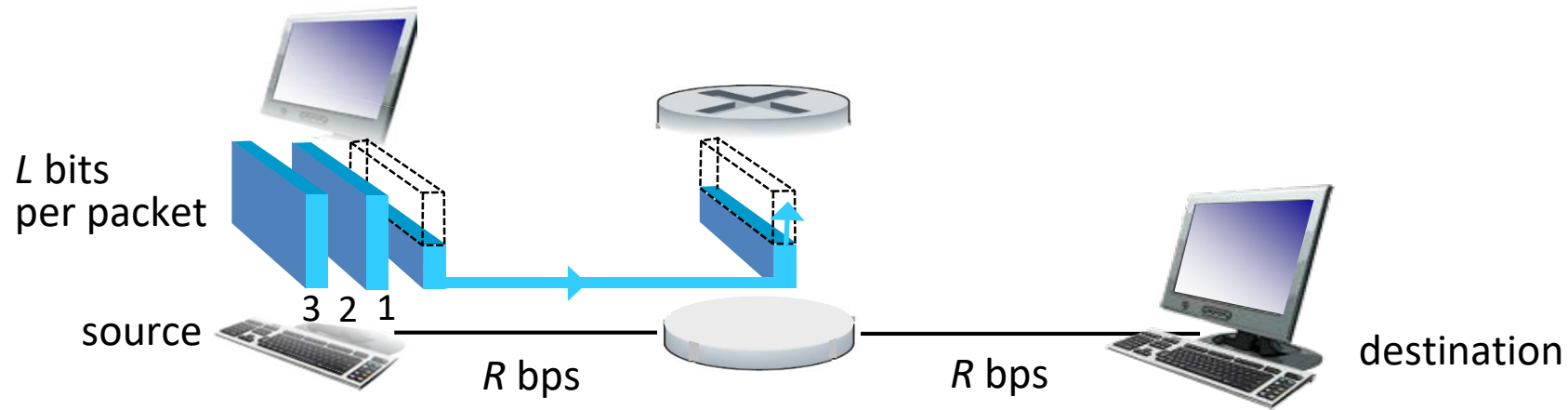
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
 - forward packets from one router to the next, across links on path from source to destination
 - each packet transmitted at full link capacity



packet-switching: شبکه اینترنت یک شبکه دیتا گرام هست مبتنی بر packet-switching و در این شبکه هاست های انتهایی اطلاعات اپلیکیشن ها رو در قالب پکت توی شبکه می فرستن این بسته ها از طریق روترهای شبکه دست به دست بهم دیگه داده میشه تا در نهایت به کامپیوتر مقصد رسونده بشن

در هر هاپ از این مسیر بین هر دوتا روتر یک لینک ارتباطی وجود داره با یک ریت ارسال مشخص و بسته ها یکی یکی توی این لینک ارسال میشن و هر بار که یک پکت به یک لینک ارسال میشه با سرعت کامل اون لینک ارسال اون انجام می گیره

Packet-switching: store-and-forward

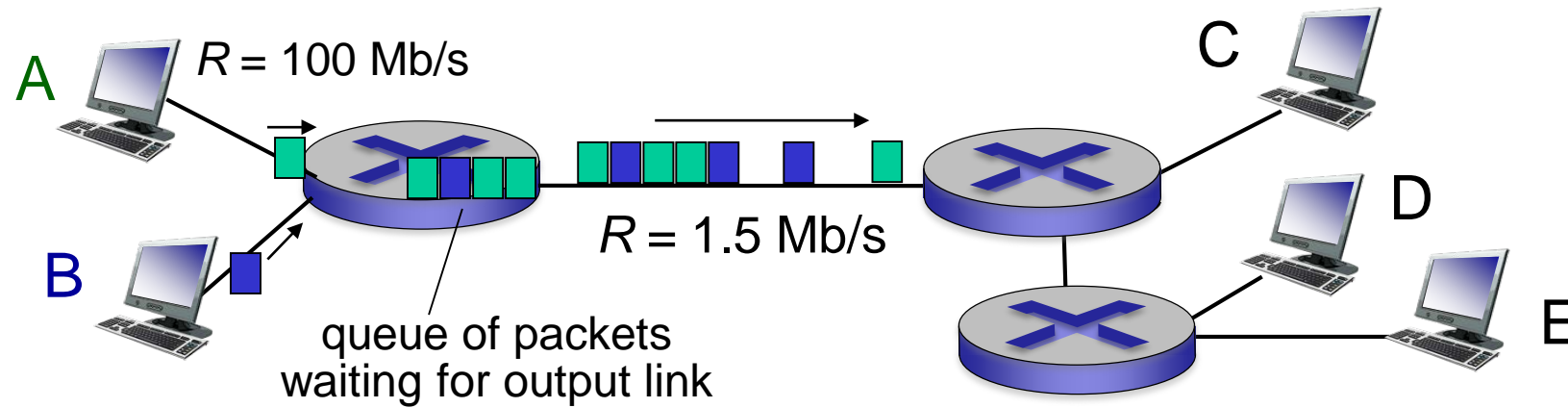


- takes L/R seconds to transmit (push out) L -bit packet into link at R bps
- *store and forward*: entire packet must arrive at router before it can be transmitted on next link

ارسال پکت در هر نود بین راه به صورت store-and-forward انجام میگیره

در این روش یک پکت به طور کامل باید توسط نود میانی دریافت بشه و بعد یک پردازش هایی باید روش انجام بشه واسه همین است که باید به طور کامل دریافت بشه به طور مشخص هدر اون باید بررسی بشه و براساس ادرس مقصد اون و روتینگی که قبلا انجام شده مسیر باید مشخص بشه و در نهایت پکت در مسیری که مشخص شده به پورت خروجی و لینک بعدی ارسال میشه این ارسال براساس ریت اون لینک صورت میگیره اگر اندازه بسته L بیت باشه و سرعت لینک R زمان ارسال بسته به لینک L/R ثانیه خواهد بود به این ترتیب برای هر نود میانی اینقدر تاخیر برای ارسال به لینک خواهیم داشت و در مجموع به تعداد نودهای میانی ما L/R تاخیر ارسال برای این بسته خواهد بود

Packet Switching: queueing delay, loss



queueing and loss:

- if arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
 - packets will queue, wait to be transmitted on link
 - packets can be dropped (lost) if memory (buffer) fills up

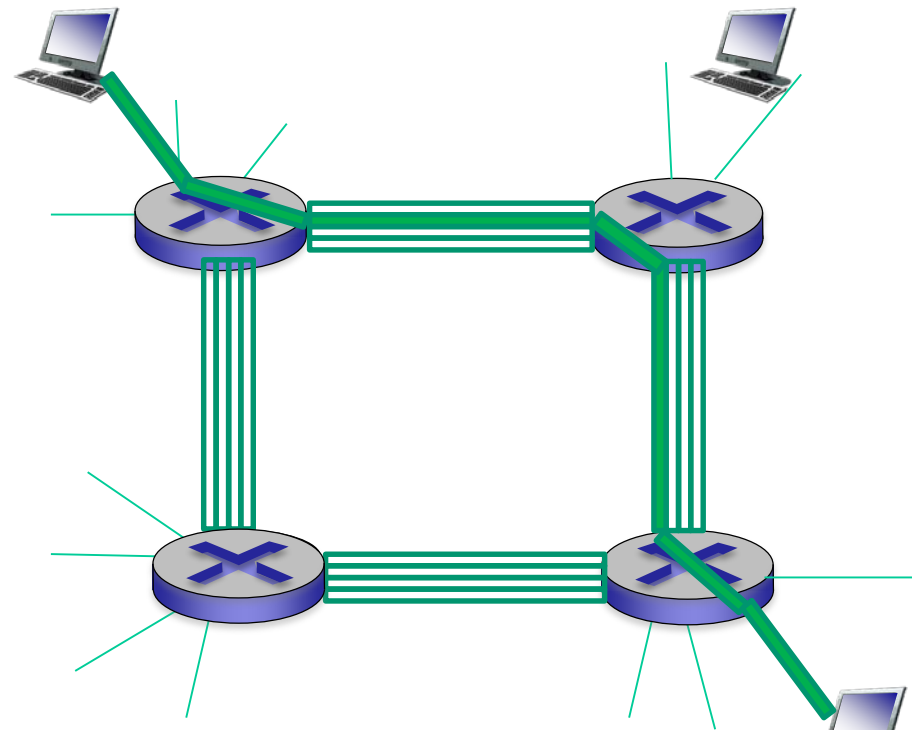
-

در نودهای میانی علاوه بر زمان ارسال ما تاخیر دیگه ای به نام تاخیر صف داریم
این تاخیر صف در روترهای بین راه اتفاق می افته

Alternative core: circuit switching

end-end resources allocated to, reserved for “call” between source & dest:

- in diagram, each link has four circuits.
 - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
 - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (*no sharing*)
- commonly used in traditional telephone networks



در مقایسه با شبکه اینترنت که از ابتدا براساس Packet Switching طراحی شد شبکه تلفن براساس circuit switching طراحی شد و در اون بین مبدا و مقصد یک circuit ایجاد میشه و وقتی که یک circuit بین مبدا و مقصد برقرار شد بعد مکالمه روی اون circuit انجام میشه ینی اطلاعات روی این circuit انجام میشه
این circuit در ابتدا یک سیم مسی یکپارچه بود و تمام عرض باند این سیم در اختیار اون مکالمه قرار میگرفت

در این مثال ما شبکه ای داریم از 4 سویچ و بین هر دوتا سویچ 4 تا لینک وجود داره اینجا برای مثال ما همچین مسیری ایجاد کردیم --> با استفاده از لینک شماره 2 بین اون دوتا سویچ و لینک شماره یک بین اون دوتا سویچ دیگه
نکته: منابع این مسیر به صورت کاملاً اختصاصی در اختیار این ارتباط قرار میگیره و هیچ نوع شیرینگی در اون صورت نمیگیره در نتیجه معادل همون circuit است
اگر مبدا و مقصد اطلاعاتی برای ارسال نداشته باشن چیزی منتقل نمیشه روی اون circuit و عملاً خالی می مونه

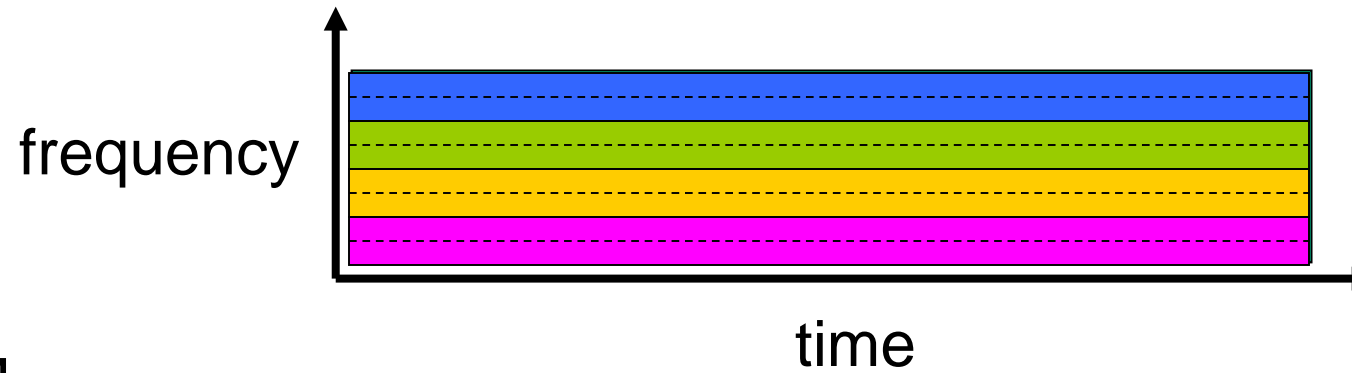
توی این حالت ما صف نداریم پس تاخیر صف هم نداریم --> پس تاخیرها محدود میشه به تاخیر ارسال و تاخیر انتقال --> پس تاخیر کم است و کیفیت سرویس خوبی اینجا داریم ولی این کیفیت سرویس خوب هزینه ای داره و اون هم این است که اگر این کانال چیزی برای ارسال نداشته باشه کاملاً idle می مونه و استفاده ای ازش نمی شه در نتیجه شیرینگ روی این کانال صورت نمی گیره پس همچین کانالی هزینه اش زیاد میشه

Circuit switching: FDM versus TDM

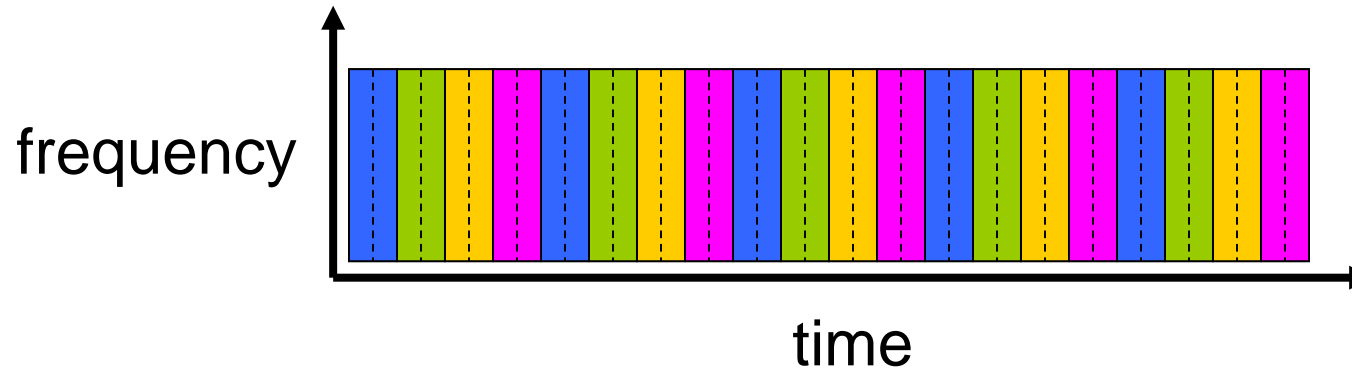
FDM

Example:

4 users



TDM



FDM: اینجا مثلا عرض باند رو به **circuit 4** تقسیم کردیم و در هر کدوم از این ها اطلاعات یکی از کانکشن ها رو ارسال می کنیم و در تمام این مدت تمام ظرفیت این قسمت از عرض باند مثلا قسمت ابی رنگ به طور اختصاصی در اختیار این ارتباط است

TDM: در میدان زمان عرض باند به کانکشن های مختلف تخصیص داده میشه و هر بار که لینکمون به یک کانکشن تخصیص داده میشه تمام رنج فرکانسی اون اون در اختیار اون کانکشن خواهد بود مثلا قسمت ابی رنگ به این ترتیب طی زمان ما ابتدا اطلاعات یک کانکشن رو به طور کامل می فرستیم و بعد اطلاعات کانکشن دیگر رو و همینطور به نوبت --> اینجا یک فریم داریم که به 4 تا تایم اسلات تقسیم شده که تایم اسلات اولی به یک کانکشن داده شده و دومی به بعدی و همین ترتیب و بعد از اون دوباره تکرار شده

در این حالت وقتی که ما یک تایم اسلات رو به یک کانکشن اختصاص میدیم در بازه زمانی اون تایم اسلات اون کانکشن با کل ریت لینکمون اطلاعات رو می تونه بفرسته ولی در حالت قبلی وقتی بخشی از عرض باندمون به یک کانکشن اختصاص داده میشه عملا سرعت ارسال اطلاعات در اون بخش محدود میشه حدود $1/4$ سرعت کل لینکمون به این ترتیب اگر کانالمون ریتش R باشد در اینجا هر کدوم از این ها $R/4$ ریت رو خواهند داشت توی **FDM**

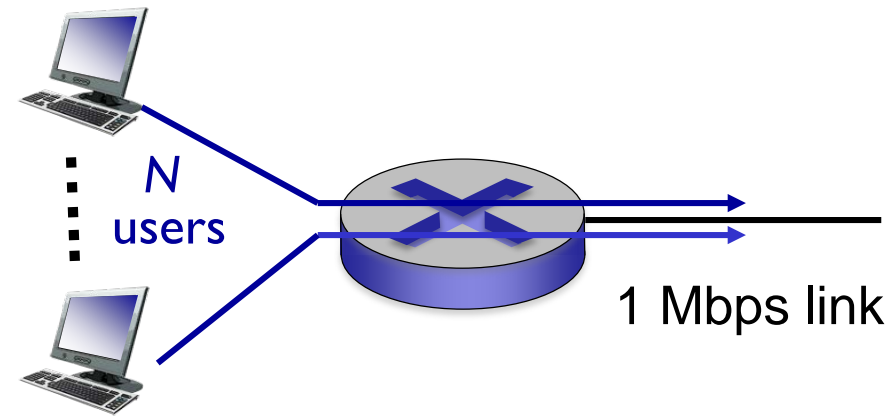
ولی توی **TDM** وقتی که اطلاعات ارسال میشه توی کانال نرخ ارسال R است ولی در $1/4$ زمان ها این ارسال انجام میشه پس در نتیجه در یک فریم اگر در نظر بگیریم هر از یک کانکشن هامون دوباره $R/4$ می تونن اطلاعات بفرستن

Packet switching versus circuit switching

packet switching allows more users to use network!

example:

- 1 Mb/s link
- each user:
 - 100 kb/s when “active”
 - active 10% of time
- *circuit-switching:*
 - 10 users
- *packet switching:*
 - with 35 users, probability > 10 active at same time is less than .0004 *



-
در مقابل در Packet switching ما اجازه میدیم که کانکشن های متعددی از لینک به صورت اشتراکی استفاده بکنن و این باعث میشه که اگر یکی از این کانکشن ها پکتی برای ارسال نداشت لینکمون idle نمونه بلکه پکت یک کانکشن دیگه ارسال بشه

مزیت: برای مشتری هزینه لینک تقسیم میشه بین اون ها پس هزینه ارتباط پایین میاد در شبکه های Packet switching سعی میشه تعداد این پکت ها اونقدر زیاد باشه که این لینک هیچ وقت خالی نمونه به عبارت دیگر سعی میشه که اینجا در پورت خروجی روتر ما یک صف کوچیکی داشته باشیم که در لحظاتی که هیچ بسته جدیدی نمیاد بتونیم ظرفیت لینک رو استفاده بکنیم

توی Packet switching تاخیر صف هم میتونه اتفاق بیوفته
برای مثال ما اگر یک لینک با عرض باند 1Mb/s داشته باشیم و هر یوزر وقتی که اکتیو است 100kb/s ارسال میکنه ولی میدونیم یوزر 10 درصد مواقع اکتیو است و 90 درصد مواقع idle اگر این لینک رو به صورت circuit switching استفاده بکنیم 10 تا یوزر داریم ولی اگر همین لینک رو به صورت Packet switching استفاده بکنیم میتونیم تا 35 تا یوزر داشته باشیم --> اینجا احتمال اینکه 10 تا یوزر همزمان فعال بشن و در حال ارسال باشن از 0.0004 کمتر است

Packet switching versus circuit switching

is packet switching a “slam dunk winner?”

- great for bursty data
 - resource sharing
 - simpler, no call setup
- **excessive congestion possible:** packet delay and loss
 - protocols needed for reliable data transfer, congestion control
- **Q: How to provide circuit-like behavior?**
 - bandwidth guarantees needed for audio/video apps
 - still an unsolved problem (chapter 9)

ایا همیشه ما باید Packet Switching رو استفاده بکنیم؟

بله اگر شبکه ما یک شبکه دیتا باشه و فقط دیتا توش منتقل بشه ما با Packet Switching از شبکه استفاه میکنیم و هزینه رو کم میکنیم با شیر کردن ظرفیت لینک هامون و استفاده مشترک از منابع که توی شبکه که عرض باند لینک ها هست

و Packet Switching به این ترتیب ساده تر است چون هیچ نوع call setup و برقرار مسیر توش وجود نداره بسته ها آماده میشن و ارسال میشن و به مقصد رسونده میشن

و اینجا می تونه صف تشکیل بشه و تاخیرها بالا بره اصطلاحا congestion می تونه اتفاق بیوفته و این به هر حال وقتی که congestion ایجاد بشه تاخیرها بالا می ره و امکان پکت لاس هم هست پس اولاً سعی میکنیم congestion ایجا نشه و سعی میکنیم اگر هم ایجاد شد اینو از بین ببریم و در عین حال از پروتکل های reliable data transfer استفاده میکنیم که اگر پکتی هم گم شد بتونیم اینو جبران بکنیم

علازم تمام این اقداماتی که برای بهتر کردن شرایط در این شبکه انجام میدیم این شبکه ها برای اپلیکیشن های ریل تایم که عمدتاً اپلیکیشن هایی هستن که audio/video باید توش منتقل بشه --> در این اپلیکیشن ها تاخیر از یک حدی نباید بیشتر بشه و باید تنظیم بشه که از اون حد بیشتر نمیشه و این مستلزم این است که یک حداقل عرض باندی گارانتی بشه برای این کانکشن ها و این ها در یک شبکه Packet Switching به صورت عادی امکان پذیر نیستن

نکته : می تونیم در یک شبکه Packet Switching یک رفتاری معادل circuit switching ایجاد بکنیم با مکانیزیم هایی که برای مدیریت تاخیر معرفی کردیم ولی یک شرطی داره که اون هم این است که مسیر ثابت باشه

Connection setup

- 3rd important function in *some* network architectures:
 - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
 - routers get involved
- network vs transport layer connection service:
 - *network*: between two hosts (may also involve intervening routers in case of VCs)
 - *transport*: between two processes

- این مستلزم Connection setup است

Connection setup در شبکه تلفنی همون شماره گیری است که همون ابتدا انجام میگیره و اون مراحلیه که طی میشه تا ارتباط برقرار بشه و بعد با گذاشتن تلفن این کانکشن قطع میشه

توی لایه شبکه هم اگر کانکشن ایجا بکنیم مستلزم Connection setup است در circuit switching این Connection setup انجام میگیره به صورت فیزیکی و در Packet Switching اگر ما بخوایم یک مسیر ثابت ایجاد بکنیم این مستلزم این است که یک Connection setup توی اون مسیر انجام بگیره و این ارتباط توی اون مسیر برقرار بشه

خود این مسیر چگونه برقرار میشه و چگونه برقرار می مونه و چجوری قطع میشه؟ یک مسیر مجازی در یک شبکه Packet Switching ینی Packet Switching که همه پکت ها از یک مسیر میرن و این برخلاف اون ویژگی است که Packet Switching داره --> این مستلزم کارهایی دیگه است که باید توی شبکه انجام بگیره بعضی از شبکه ها براین اساس طراحی شده بودن و توی اون ها این اتفاق می افتاد شبکه های ATM, frame relay, X.25 این ها شبکه هایی هستن که Connection setup توشون وجود داره برای یک ارتباط بین مبدا و مقصد یک کانکشن برقرار میشه و بعد توی اون کانکشن است که بسته ها ارسال میشه هر چند که خود بسته ها به صورت پکت ارسال میشه ینی این که این کانکشن ها، کانکشن های مجازی هستن و عرض باندشون شیر میشه بین پکت ها منتها بین اون پکت هایی که حداقل توی بخشی از مسیر با هم هستن پس به عبارت دیگه در عمل در هر لینک ما یک تعداد مسیر setup کرده ایم که این لینک رو شیر می کنن و مشترک هستن و عرض باند لینک بین اونها تقسیم میشه و هر کدوم بسته هاشون به هر حال از این لینک عبور داده میشه

یک چنین مسیری setup کردنش مستلزم این است که روترها درگیر باشن و در ابتدا قبل از اینکه پکت ها ارسال بشن توی شبکه روترها باید این مسیر رو establish بکنن که این یک مسیر virtual خواهد بود که برای بسته های مربوط به این کانکشن مشخص میشه که اینا دقیقا باید از چه مسیری باید منتقل بشن عملا در سطح شبکه به جای اینکه یک روتر تیپلی داشته باشیم که با روتر پروتکلی به طور داینامیک مسیرهای مناسب برای مقصد های مناسب توش ست میشه به ازای هر کانکشن توی هر نود مشخص میکنیم که بسته های این کانکشن توی چه مسیری باید به سمت مقصد هدایت بشن و این ثابت می مونه تا انتهای کانکشن که با قطع کانکشن این مسیر از بین می ره

این کانکشن در سطح شبکه بین هاست مبدا و مقصد ولی به طور فیزیکی از طریق روترهای شبکه برقرار میشه که در مقایسه با کانکشن tcp این کانکشن بین دوتا پروسس در هاست مبدا و مقصد برقرار میشه و به طور کامل مجازی است و در لایه شبکه هیچ معادل فیزیکی نداره این کانکشن

Network service model

Q: What *service model* for “channel” transporting datagrams from sender to receiver?

example services for individual datagrams:

- ❖ guaranteed delivery
- ❖ guaranteed delivery with less than 40 msec delay

example services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

لایه شبکه یک سرویسی به لایه transport میده و از طریق اون به اپلیکیشن که مدل سرویس شبکه رو مشخص میکنه و برای شبکه دیتاگرام این سرویس یک سرویس **best effort** است ینی هیچ تضمینی روی رسیدن بسته ها و تاخیر اونها توش وجود نداره در مقایسه حالا اگر شبکه ما یک شبکه ای باشه که که یک کانکشن ستاپ میکنه و یک **virtual circuit** ایجاد میکنه بین مبدا و مقصد و اطلاعات رو توی این کانکشن ارسال میکنه این شبکه چه سرویسی به لایه بالاتر میده؟ این سرویس در سطح دیتاگرام برای هر پکت می تونه این باشه که پکت حتما تحویل داده میشه چون یکی از ویژگی های **circuit** این است که صف تشکیل نمیشه و پکت لاس نداریم و یا اینکه گارانتی تحویل با حداکثر 40 میلی ثانیه تاخیر

در سطح **flow** سرویسی که داده میشه این است که ترتیب این ها حفظ میشه چون همشون از یک مسیر می رن و برای این کانکشن یا **Flow** یک حداقلی عرض باندی رو می تونیم تخصیص رو بدیم ینی مدل سرویسمون این است که یک حداقل عرض باندی رو گارانتی می کنه و ویژگی دیگر این است که تاخیر بین بسته ها ینی تاخیر هر بسته با بسته قبلیش یا بعدی هم تفاوت چندانی در این سرویس پیدا نمیکنه

Network layer service models:

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

ویژگی های سرویس لایه شبکه به اپلیکیشن:
توضیح یکیش:

در اینترنت که Packet Switching است --> مدل best effort است و عرض باند گارانتی
نمیشه و بقیش هم گارانتی نمیشه

جدول رو بخون و شبکه هیچ فیدبکی برای congestion نداره

ATM که براساس virtual circuit switching طراحی شده --> در CBR یک عرض باند
مشخصی رو شبکه اختصاص میده به کانکشن و در VBR در اینجا هم عرض باند اختصاص داده
میشه ولی اینجا ریت اپلیکیشن می تونه متغییر باشه و در ABR ما یک حداقل عرض باندی رو
گارانتی میکنیم نه الزاما عرض باندی که دیتامون لازم داره و در UBR هیچ عرض باندی گارانتی
نمیشه

در شبکه ATM برای این سرویس ها شبکه مکانیزیم هایی رو پیاده میکنه که با استفاده از اون
مکانیزیم ها این ویژگی های گفته شده می تونه به دست بیاد مثلا برای کانکشن CBR ما می تونیم
loss , timing رو گارانتی بکنیم حداکثرشو

ATM یک ویژگی دیگری هم دارد که باعث میشه که این مکانیزیم ها اونجا موثر واقع شده و اون
ویژگی این است که ATM یک شبکه virtual circuit switching است و شبکه ای است که در
اون مسیرها مشخص میشه و ثابت میشه

Connection, connection-less service

- ❖ *datagram* network provides network-layer *connectionless* service
- ❖ *virtual-circuit* network provides network-layer *connection* service
- ❖ analogous to TCP/UDP connection-oriented / connectionless transport-layer services, but:
 - *service*: host-to-host
 - *no choice*: network provides one or the other
 - *implementation*: in network core

در سطح شبکه میتوانیم یک سرویس connecton-oriented یا یک سرویس connectonless ارائه بدیم

سرویس connectonless همون همون شبکه های دیتاگرام است یا Packet Switching که در اون هیچ کانکشنی بین مبدا و مقصد از طریق شبکه برقرار نمیشه و پکت ها مستقلا به شبکه داده میشن و هر پکت مستقلا مسیر خودش رو طی میکنه به مقصد می رسه

سرویس connecton-oriented <-- شبکه های virtual circuit یک همچین سرویسی رو ارائه میدن که در اینجا بین مبدا و مقصد یک مسیر ثابتی مشخص میشه و هرچند که شبکمون هنوز Packet Switching است ولی این پکت ها برای یک کانکشن یک مسیر ثابتی رو طی میکنن <-- توجه این ها در سطح لایه شبکه است

TCP <-- هست connecton-oriented

UDP <-- هست connectonless

نکته: کانکشن های لایه transport بین پروسس ها در هاست مبدا و مقصد برقرار است در حالی کانکشن هایی ک در virtual circuit switching مطرح میکنیم در سطح شبکه بین هاست مبدا و هاست مقصد ایجاد میشه و ویژگی که این کانکشن داره این که است شبکمون باید یکی از این دو حالت یا باید دیتاگرام باشه یا circuit ولی تفاوتی که اینجا وجود داره در سطح شبکه این کانکشن های باید بین روترهای توی شبکه در core شبکه برقرار باشه و در نتیجه شبکه باید این قابلیت رو داشته باشه که این نوع سرویس ر ارائه بده و این کانکشن ها رو ایجاد بکنه و اگر این قابلیت وجود نداشته باشه ناچاریم که همون دیتاگرام رو استفاده بکنیم

Virtual circuits

“source-to-dest path behaves much like telephone circuit”

- performance-wise
 - network actions along source-to-dest path
-
- call setup, teardown for each call *before* data can flow
 - each packet carries VC identifier (not destination host address)
 - every router on source-dest path maintains “state” for each passing connection
 - link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

در یک شبکه virtual circuit switching عملاً ما شبکه یک Packet Switching داریم که توش virtual circuit switching طراحی میشه و در این صورت مسیر بین مبدا و مقصد مشابه بین circuit switching که توی شبکه تلفنی خواهد بود

این مشابهت از دو نظر است --> اولاً مسیر ثابت است و در نتیجه مکانیزم هایی رو توی این مسیر در نودهای بین راه می تونیم طراحی کنیم و از نظر پرفرمنس این شبیه circuit switching تلفن میشه ینی در واقع توی این مسیر با مکانیزم هایی که توش پیدا میکنیم می تونیم کاری بکنیم که اون شیرینگ کم بشه و رقابت نباشه و صف ها کوتاه بشن و تاخیرها کم بشن و لاس اتفاق نیوفته virtual circuit switching مستلزم این است که call setup انجام بشه و در انتها هم کانکشن قطع بشه و کال خاتمه پیدا بکنه و این مستلزم این است که هر پکت یک virtual circuit identifier با خودش حمل بکنه چون ما در اینجا در این روش پکت ها براساس این که مال کدام virtual circuit هستن توی نودهامون هدایت میشن به مسیری که برای اون virtual circuit ایجاد شده و دیگه اینجا مقصد نقشی نداره و این مستلزم این است که هر روتر در مسیر ستاپ شده یک استیتی رو برای هر کانکشنی نگهداری بکنه و براساس اون استیت مشخصاً بسته های اون کانکشن رو تعیین بکنه و هدایت بکنه به پورت خروجی مورد نظر و در virtual circuit این قابلیت به وجود میاد که منابع روترهامون در واقع عرض باند و بافر پورت خروجی بتونیم به یک کانکشن خاص اختصاص بدیم و با این ترتیب با اختصاص دادن منابع به اندازه کافی مثلاً با تخصیص عرض باند کافی برای یک کانکشن یا .. می تونیم تاخیر یا لاس رو تا حد مورد نظر محدود بکنیم

VC implementation

a VC consists of:

1. *path* from source to destination
 2. *VC numbers*, one number for each link along path
 3. *entries in forwarding tables* in routers along path
- ❖ packet belonging to VC carries VC number (rather than dest address)
 - ❖ VC number can be changed on each link.
 - new VC number comes from forwarding table

پیاده سازی virtual circuit یا VC به چه صورت انجام میگیره؟

ما در VC برای هر کانکشن یک virtual circuit ایجاد میکنیم

یک virtual circuit تشکیل میشه از یک مسیر از مبدا تا مقصد و یک id که بسته های این کانکشن که توی این virtual circuit منتقل خواهند شد باید براساس این number شناسایی بشن پس VC number یک id است برای بسته های virtual circuit مون

استیت در هر نود میانی براساس همین id که مشخص میکنه بسته های این virtual circuit به کدوم پورت خروجی باید منتقل بشن که این همون اطلاعاتی است که در کانکشن ستاپ باید توی یک جدول گذاشته باشه و تا انتهای این ارتباط این جدول مبنای فورواردینگ برای بسته های این virtual circuit در این نود خواهد بود

و در همه نودها باید یک چنین جدول هایی وجود داشته باشه برای اینکه مسیرمون از مبدا تا مقصد برقرار بمونه و تغییر نکنه

به این ترتیب پکت ها در چنین سرویسی VC number با خودش دارن به جای این که

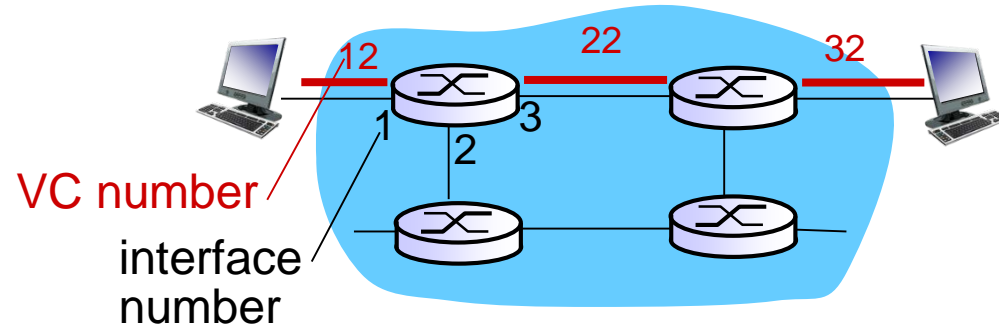
destination address رو داشته باشن و این VC number لازم نیست از مبدا تا مقصد یک نامبر مشخص برای یک کانکشن اختصاص داده بشه بلکه این می تونه پیرهاب اختصاص داده بشه

و در نتیجه ما یک VC داریم توی یک مسیری از مبدا تا مقصد ولی در فورواردینگ تیبل VC

number این virtual circuit برای لینک بعدی مشخص میشه و VC number ها جایگزین

میشن پس virtual circuit توی هر لینک VC number متفاوتی می تونه داشته باشه بدون اینکه خدشه ای به اون وارد بشه

VC forwarding table



*forwarding table in
northwest router:*

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

VC routers maintain connection state information!

مثال:

یک virtual circuit بین مبدا و مقصد ایجاد شده

این virtual circuit از طریق دوتا روتر بالایی برقرار شده پس این یک مسیر ثابتی است در مرحله call setup این مسیر برقرار میشه و بعد بسته ها توی این مسیر ارسال میشن و در انتها این call قطع میشه

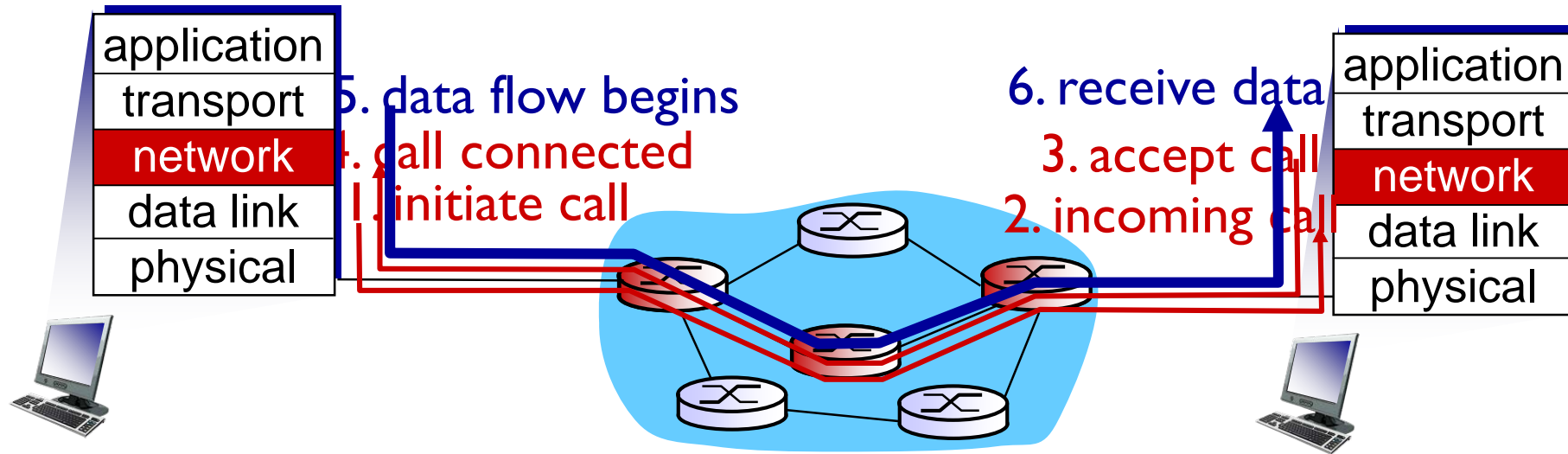
بسته های که توی این کانکشن ارسال می شن باید با VC number مشخصی ارسال بشن
نود بین راهمون براساس VC number تشخیص میده که این بسته ها مال کدوم virtual circuit هستن و بعد یک جدولی داره که توی این جدول برای هر virtual circuit اطلاعاتی لازم ثبت شده که این بسته ها برای هر virtual circuit از کدام اینترفیس میان و با چه VC number میان و بعد به کدوم پورت این ها باید ارسال بشن --> این مسیر رو مشخص میکنه
برای گام بعدی مسیرمون VC number میتونه جایگزین بشه که اینجا + مشخص میشه که ادامه مسیر با این VC number خواهد بود

پس روترها برای هر virtual circuit برای هر کانکشن این اطلاعات رو نگهداری می کنن توی این جدول و برای کانکشن دیگری هم که همزمان توی این روتر وجود داره بین پورت های ورودی مختلف و پورت های خروجی مختلف این اطلاعات پر میشه

نکته: تمام مسیرهایی که از ورودی های مختلف ممکنه توی روتر بیان و مقصدشون به گونه ای است که مسیری که براشون انتخاب میشه به اون سمتی که توی شکل هست برقرار میشه همه اونها این لینک رو شیر می کنن پس اینجا شبکمون هنوز شبکه Packet Switching است و شبکمون circuit switching نمیشه پس شبکمون Packet Switching است و پکت ها دارن منتقل میشن ینی پکت ها لینک ها رو به صورت اشتراکی استفاده می کنن منتها تفاوتی که داریم اینه که اینجا دیگه دیتاگرام نیست ینی بسته ها براساس اون روتینگ تیبل و فورواردینگ تیبل مبتنی بر مقصد است هدایت نمیشن بلکه براساس این تیپلی که مال virtual circuit switching است هدایت میشن

Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



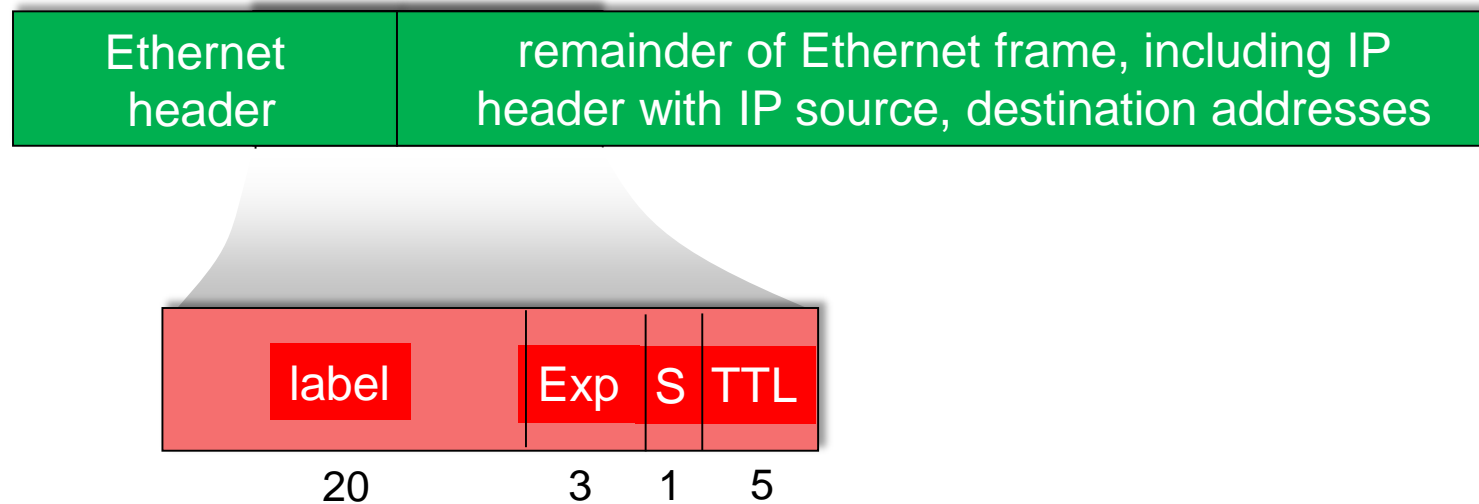
virtual circuit switching مستلزم سیگنالینگ است

اگر بخوایم مسیر ثابت بشه باید کانکشن ستاپ انجام بشه و کانکشن ستاپ روی مسیر مشخصی انجام بگیره که این به هر شکلی می تونه تعیین بشه چه مسیری ولی باید روترها با هم صحبت کنن و این مسیر رو ستاپ بکنن که این مستلزم یک پروتکلی برای صحبت کردن بین روترها است که بهش سیگنالینگ می گیم ینی اینجا اطلاعات کنترلی بین روترها رد و بدل میشه این سیگنالینگ برای ستاپ کردن و همینطور maintain کردن کانکشن و در نهایت قطع کانکشن استفاده میشه

maintain کردن ینی اگر در حین ارتباط مثلا یک لینکی قطع شده یا یک روتری یک مشکلی براش پیش اومد این پروتکل با سیگنالینگ که انجام میده سعی میکنه مسیر جایگزین پیدا بکنه در نتیجه maintain کردن مسیرهای ایجاد شده هم بخشی از پروتکل است شبکه های ATM و frame-relay و x.25 هر کدوم سیگنالینگ خودشون رو دارن ولی این ها شبکه ای هستن که استفاده نمیشن و منسوخ شدن کاری که پروتکل های سیگنالینگ انجام میدن این است که ابتدا اون مبدا یک کانکشن رو درخواست میکنه و این یک call رو شروع میکنه و این درخواست به اولین نود و نودهای دیگه منتقل میشه و به مقصد رسونده میشه و به عنوان یک incoming call اونجا محسوب میشه و مقصد باید کال رو قبول بکنه و وقتی که کال قبول شد این توی همین مسیر برگردونده میشه و به مبدا می رسه و منجر به برقراری کانکشن می شه و بعد دیتا می تونه توی این مسیر که ایجاد شده ارسال و دریافت بشه

Multiprotocol label switching (MPLS)

- **goal:** high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
 - faster lookup using fixed length identifier
 - borrowing ideas from Virtual Circuit (VC) approach
 - but IP datagram still keeps IP address!

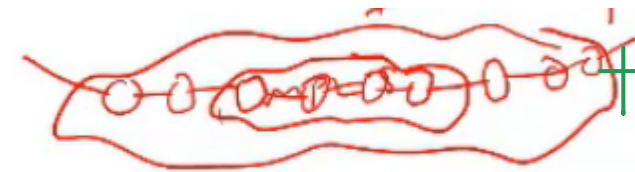
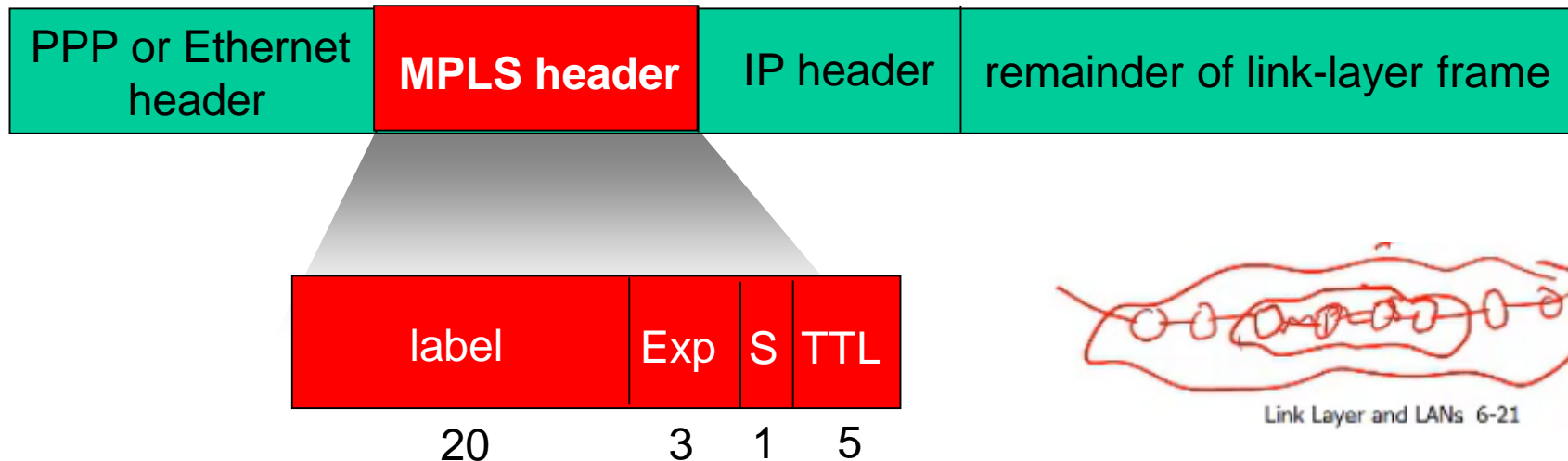


-

پروتکل MPLS معادل ATM است در شبکه های ip ینی همون کارهایی که ATM می خواست انجام بده رو ما با MPLS می تونن در شبکه های ip انجام بدیم
با استفاده از MPLS یک شبکه virtual circuit switching می تونیم ایجاد بکنیم و شبکمون می تونه MPLS رو ساپورت بکنه یا کلا MPLS باشه

Multiprotocol label switching (MPLS)

- **goal:** high-speed IP forwarding among network of MPLS-capable routers, using fixed length label (instead of shortest prefix matching)
 - faster lookup using fixed length identifier
 - borrowing ideas from Virtual Circuit (VC) approach
 - but IP datagram still keeps IP address!



Link Layer and LANs 6-21

هدف اصلی در اینجا این است که پکت فورواردینگ با استفاده از virtual circuit id یک لیبیل انجام بگیرد. توی هر پکت به جای اینکه ادرس مقصد مبنا باشد این روش virtual circuit switching غیر از اینکه مسیر رو ثابت میکنه پروسس بسته ها رو هم می تونه تصحیح بکنه

ما در این حالت به جای longest prefix match که در شبکه های ip استفاده میکردیم اینجا میایم یک lookup خیلی ساده روی همون تیبل VC انجام میدیم و براساس این VCid یا لیبیلی که روی پکت است و بعد اون لایینی که برای این لیبیل ستاپ بشه و در نتیجه مسیر رو به دست میاریم و بسته رو به اون مسیر می فرستیم پس با استفاده از MPLS می تونیم virtual circuit switching رو پیاده سازی بکنیم منتها تفاوتی که هست اینه که MPLS روی ip قرار بگیره و عملا با ip همزیستی داره به عبارت دیگر ما برای MPLS هدر دیتاگرام رو جایگزین نمیکیم بلکه یک هدر MPLS بهش اضافه میکنیم به این ترتیب ما پکتمون رو در لایه شبکه که هدر ip مابرای ارسال این توی یک شبکه MPLS یک هدر MPLS بهش اضافه میکنیم پس عملا MPLS معادل یک لایه پروتکلی است که اینجا اضافه شده و یک هدر کامل است و فیلدهای مختلف داره برای اون کارهایی که پروتکل MPLS بتونه انجام بده

توجه: در شبکه ATM لایه نت ورک کلا متفاوت است این برداشته میشه و لایه ATM جایگزین میشه ولی در صورتیکه ما از MPLS استفاده بکنیم این اتفاق نمی افته بلکه ما عملا یک لایه اضافه میکنیم اینجا زیر ip network به نام لایه MPLS --> شکل توی اسلاید برای درک بهتر: اگر شبکه ip به صورت عکس نشون بدیم توی شبکه ip مثلا core شبکه ip میتونه MPLS باشه و به این ترتیب بسته با همون ip میاد تا برسه به core و وارد core که شد این هدر بهش اضافه میشه و توی خود core براساس این هدر از روتر عبور میکنه تا انتها تا edge core و اینجا با ip هدایت میشه تا به مقصد رسونده بشه توی این فاصله ما یک هدر اضافه میکنیم و در انتهای قسمت + این هدر باید حذف بشه و ادامه مسیر باید با همون هدر ip انجام بگیره هدر MPLS چه قسمت هایی داره؟

label: اساس فورواردینگ در هر نود همین label است که 20 بیتی است

MPLS capable routers

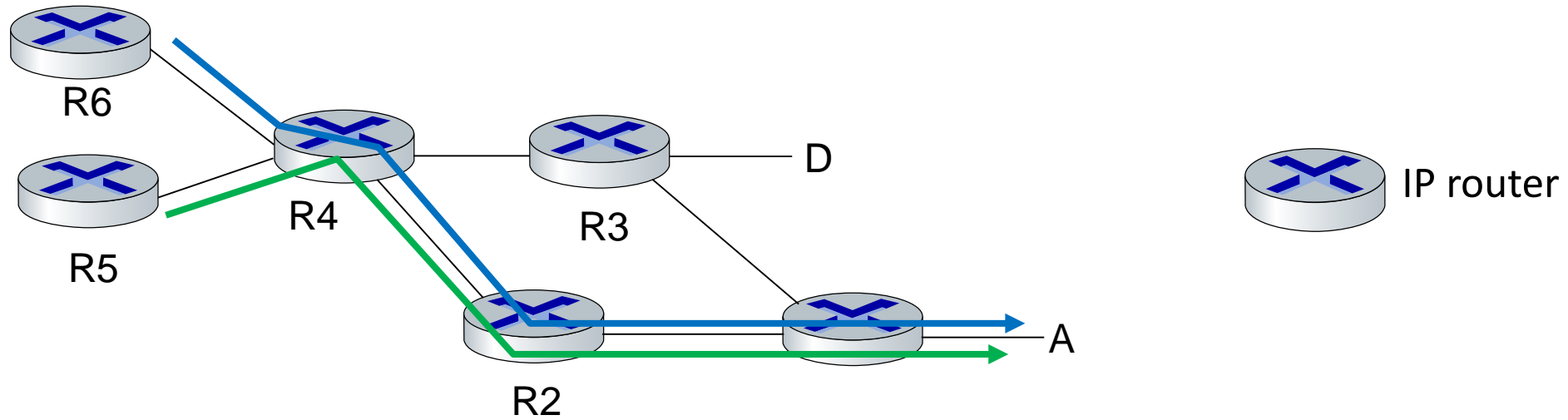
- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*don't inspect IP address*)
 - MPLS forwarding table distinct from IP forwarding tables
- *flexibility*: MPLS forwarding decisions can *differ* from those of IP
 - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
 - re-route flows quickly if link fails: pre-computed backup paths

-
روترهایی که قابلیت MPLS رو دارن <-- این روترها علاوه فورواردینگ مبتنی بر ip فورواردینگ مبتنی بر MPLS هم میتونن انجام بدن که به این روترها label-switched router یا Lsr میگن

Lsr ها پکت ها رو در مسیر تعریف شده ای که کانکشن براش برقرار شده در این مسیر فوروارد میشن براساس همون هدری که لیبله که توی هدر است و کاری به ip ادرس مقصد در اینجا نداریم و فورواردینگ هم براساس lookup روی تیبل های فورواردینگ MPLS است و این تیبل ها مستقل از فورواردینگ تیبل عادی ip مون هستند و جدا از اونها هستن و برای هر کانکشنی که ستاپ میشه توی این تیبل اون entry که گفتیم ایجاد میشه که مبنای فورواردینگ بسته های اون کانکشن قرار خواهند گرفت

flexibility: در MPLS مسیری که بسته ها باید طی بکنن براساس هر معیار دلخواهی ایجاد میشه و بعد این مسیر ستاپ میشه و توی اون تیبل ها اون اطلاعات وارد میشن و بسته ها توی اون مسیر فوروارد میشن ولی در ip معمولی مسیر صرفا براساس ادرس مقصد است
نکته: اگر کانکشن مشکل پیدا کرد خیلی سریع اطلاعات منتقل میشه به لینک کانکشن backup

MPLS versus IP paths



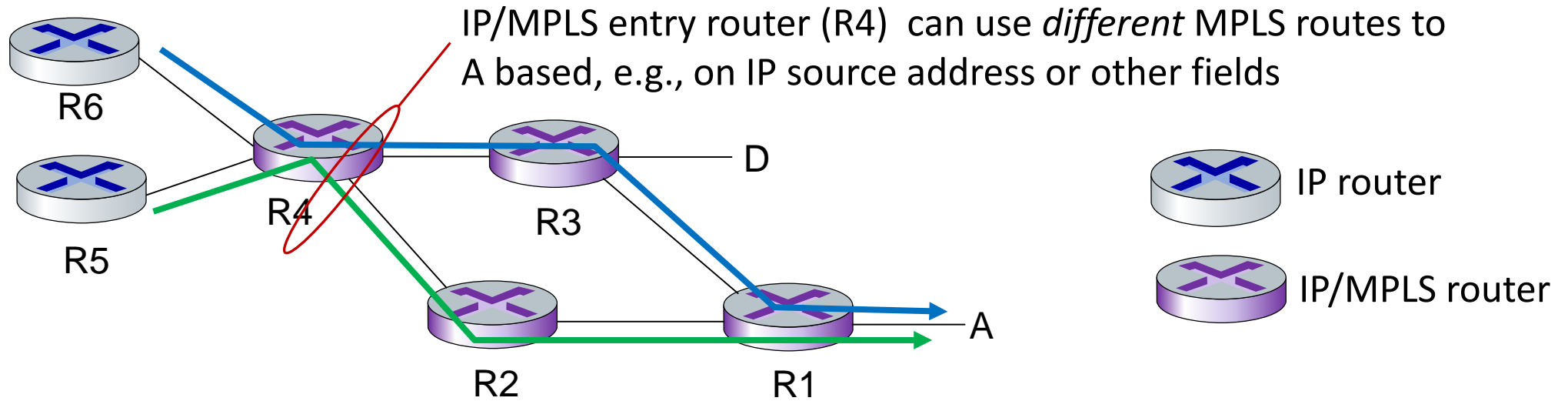
- **IP routing:** path to destination determined by destination address alone

MPLS در مقایسه با ip:

این محدودیتی که ip داره که مثلا اگر دو ارتباط وجود داشته باشه بین مبدهای مختلف و یک مقصدی و میان توی روتر R4 و براساس جدول فورواردینگ تیبل همشون ناچارن از مسیری که بهشون گفته برن پس بسته های کانکشن های مختلف با شرایط مختلف چون مقصدشون یکی است توی یک مسیر قرار میگیرند

در MPLS این محدودیت رو نداریم --> صفحه بعدی..

MPLS versus IP paths



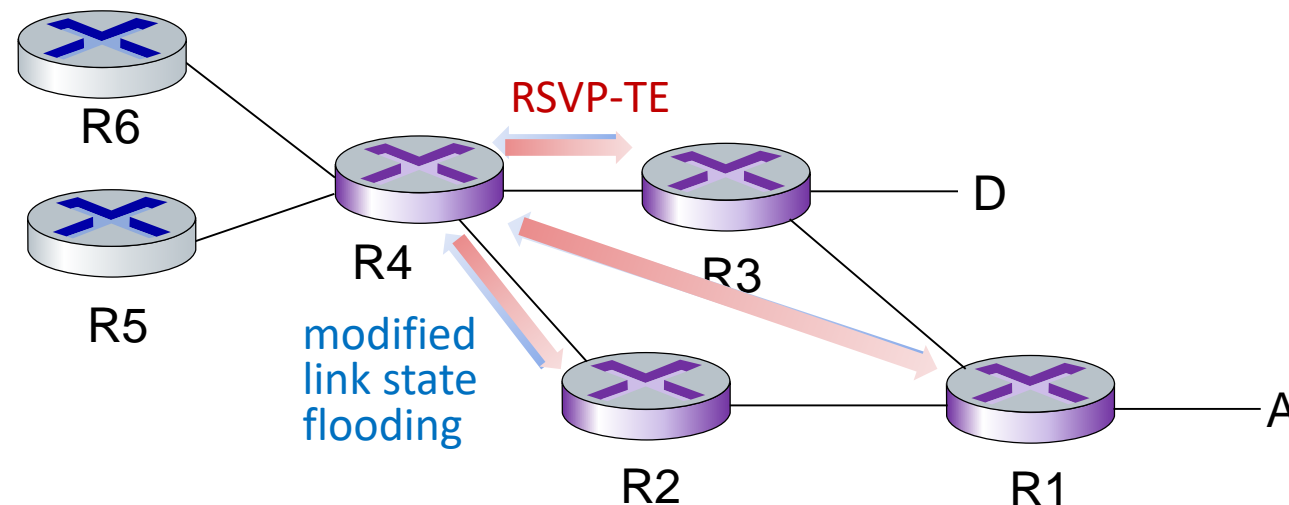
- **IP routing:** path to destination determined by destination address alone
- **MPLS routing:** path to destination can be based on source *and* destination address
 - flavor of generalized forwarding (MPLS 10 years earlier)
 - *fast reroute*: precompute backup routes in case of link failure

توی MPLS به صورت انعطاف پذیر بسته هامون حتی می تونن برای مقصد یکسان از مسیرهای مختلف عبور کنند این مستلزم این است که در مرحله **call setup** براساس ویژگی های هر کانکشنی مسیر مناسبی برای اون ستاپ بشه مستقل از بقیه کانکشن ها و اینکه مقصد چی هست که این می تونه براساس مبدا و مقصد باشه یا براساس نوع اپلیکیشن کانکشن باشه یا ...

امکان **fast reroute** وجود داره ینی در همین مرحله علاوه بر این که مسیر اصلی رو ستاپ میکنیم یک مسیر جایگزین که اشتراک لینک و نود نداشته باشه برقرار میشه که در همون مرحله که کانکشن ستاپ انجام میگیره یک کانکشن بک آپ هم ایجاد میشه و ستاپ میشه که در صورتی که شکستی رخ داد این بتونه جایگزین بشه

MPLS signaling

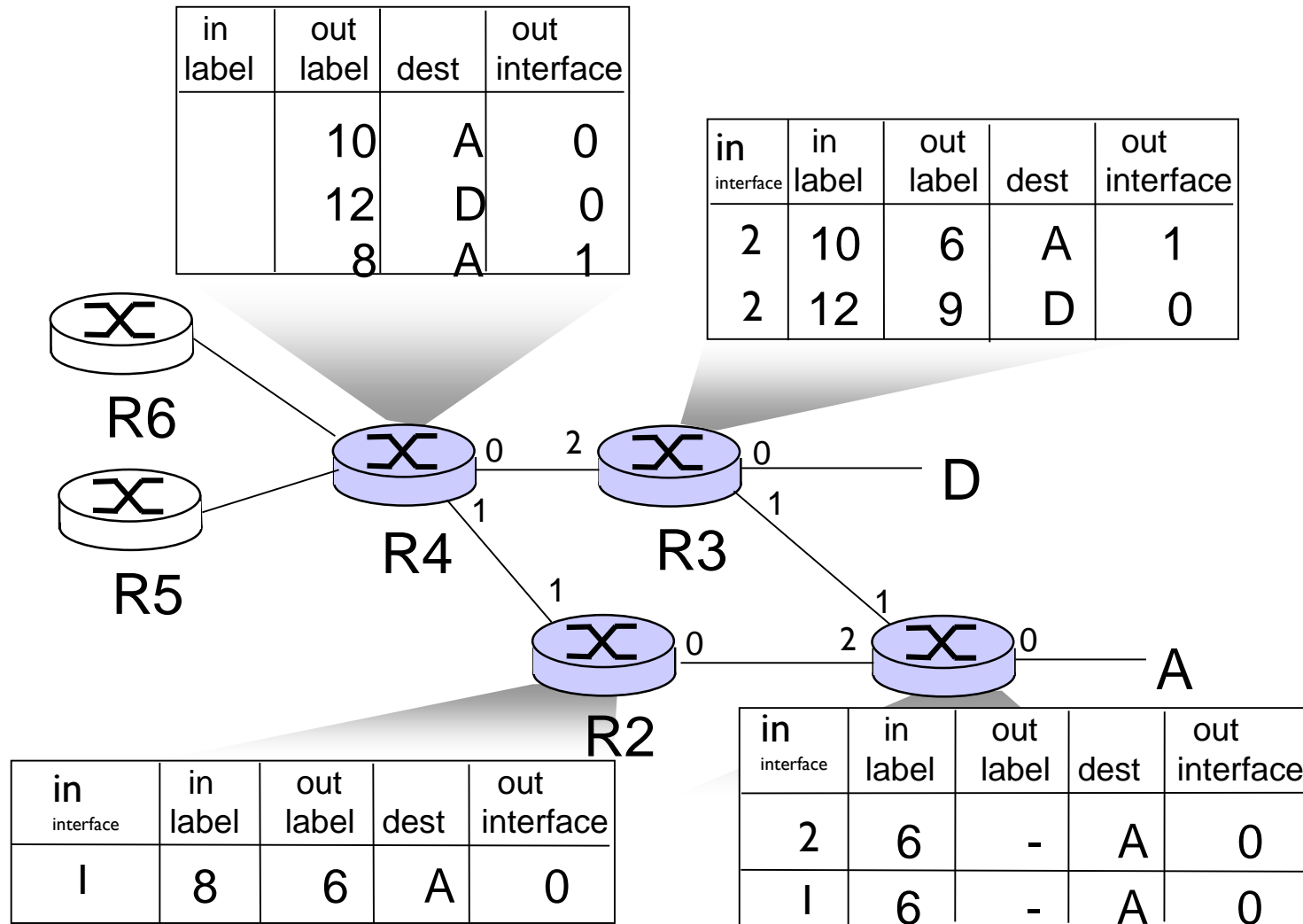
- modify OSPF, IS-IS link-state flooding protocols to carry info used by MPLS routing:
 - e.g., link bandwidth, amount of “reserved” link bandwidth
- entry MPLS router uses RSVP-TE signaling protocol to set up MPLS forwarding at downstream routers



-
MPLS یک پروتکل مبتنی بر virtual circuit switching و در شبکه های مبتنی بر virtual circuit switching باید از سیگنالینگ استفاده بکنیم

در MPLS چون روی ip است سیگنالینگ پروتکل می تونه از امکانات لایه ip برای به دست آوردن اطلاعات عرض باند و مسیر در سطح شبکه استفاده بکنه ولی در نهایت تصمیم گیری روی اون مسیرهایی که می تونه مناسب باشه برای یک کانکشنی با استفاده از پروتکل RSVP و یا ورژن جدیدتر اون RSVP-TE این انتخاب می تونه انجام بگیره --> در این پروتکل یک درخواستی روی مسیرهایی که مناسب تشخیص داده میشه به سمت مقصد ارسال میشه و این درخواست ها توی مسیر جلو می رن و رزرویشن منابع رو انجام میدن و هر کدوم که منابع نداشت ممکنه ریجکت بشه و اونی ک منابع کافی داشت به مقصد می رسه و در نهایت مقصد بین اونهایی که به مقصد رسیدن یکی رو انتخاب می کنه با هر معیاری و رزرویشن رو برمیگردونه توی همون مسیری که با این کار کانکشن ستاپ رو تکمیل میکنه

MPLS forwarding tables



مثال:

و ته این پروسه فورواردینگ تیبل های MPLS در بین نودهای بین راه ستاپ میشن و برای هر کانکشن جدید یک entry جدید توی تیبل اضافه میشه

و یکی از کارهایی که در این پروسه اتفاق می افته تخصیص لیبل است --> که لیبل به ازای هرلینک می تونه متفاوت انتخاب بشه و انتخاب لیبل برای کانکشن می تونه خیلی پروسه ساده تری بشه در مقایسه با حالتی که اگر این می خواست برای کل مسیر ینی از مبدا تا مقصد یک لیبل ثابتی داشته باشه

به این ترتیب بسته ورودی به شبکه MPLS در اولین نود به یک مسیری اختصاص داده میشه و لیبل اون روی پکت ها گذاشته میشه و بعد مشخص میشه مسیر از چه پورته عبور میکنه و بعد در نود بعدی صرفا مشخص میشه که این از چه پورته میاد و با چه لیبلی پس به این ترتیب بسته شناسایی میشه و اینکه به چه پورته باید ارسال بشه و با چه لیبلی

در نهایت این بسته مسیرشو ادامه میده تا به مقصد برسه و بعد توی مقصد براساس این لیبل دوباره شناسایی میشه و بعد اینجا لیبل حذف میشه و بعد منتقل میشه به لایه IP و براساس IP به لینکی که دو طرف مقصدش است ارسال میشه

پس در edge router ابتدایی شبکه یک هدر جدید به نام هدر MPLS اضافه میشه به بستمون و در روترهای بعدی براساس همین هدر و لیبلی که توی اون گذاشته شده بسته هدایت میشه به مقصد و در edge router انتهایی این هدر حذف میشه و ادامه مسیر براساس ip ادامه پیدا میکنه