

Chapter 4

Network Layer: Data Plane

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.

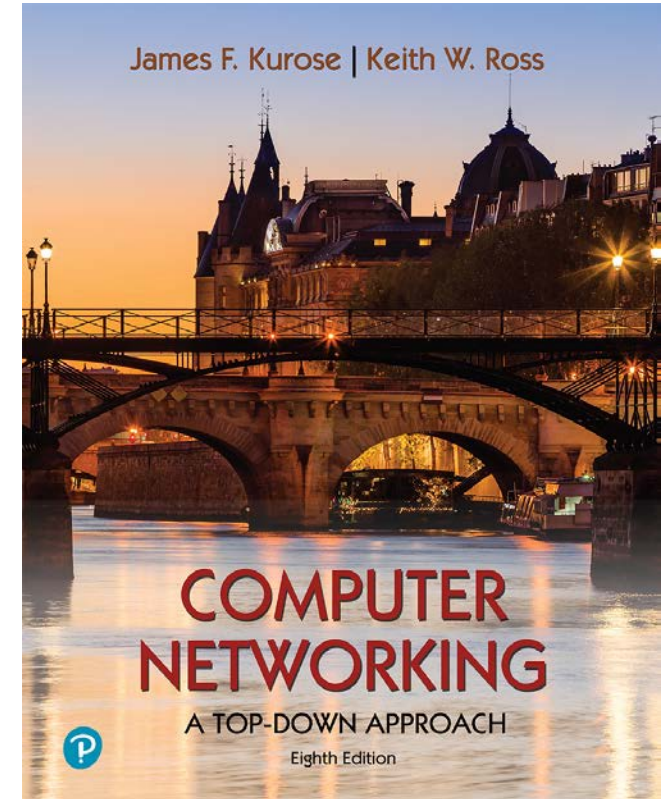
In return for use, we only ask the following:

- 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

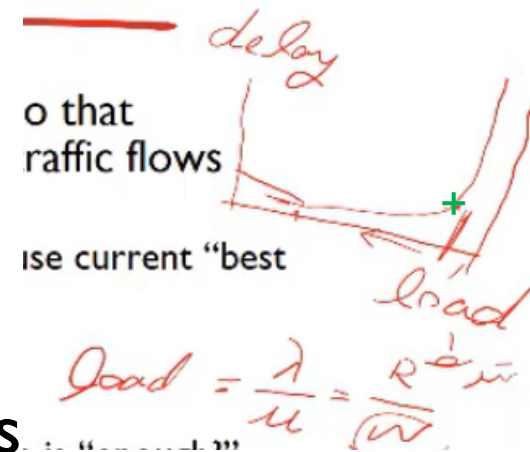
Jim Kurose, Keith Ross
Pearson, 2020

Reflections on best-effort service:

- **simplicity of mechanism** has allowed Internet to be widely deployed adopted
- sufficient **provisioning of bandwidth** allows performance of real-time applications (e.g., interactive voice, video) to be “good enough” for “most of the time”
- **replicated, application-layer distributed services** (datacenters, content distribution networks) connecting close to clients’ networks, allow services to be provided from multiple locations
- congestion control of “elastic” services helps

It's hard to argue with success of best-effort service model

Dimensioning best effort networks



- **approach:** deploy enough link capacity so that congestion doesn't occur, multimedia traffic flows without delay or loss
 - low complexity of network mechanisms (use current “best effort” network)
 - high bandwidth costs
- **challenges:**
 - **network dimensioning:** how much bandwidth is “enough?”
 - **estimating network traffic demand:** needed to determine how much bandwidth is “enough” (for that much traffic)

ادامه: این دیدگاه پیچیدگی خیلی بالایی به شبکه اضافه نمیکنه چون فقط داریم عرض باند رو افزایش میدیم و کار دیگری انجام نمیدیم ولی هزینه رو چرا خیلی افزایش میده چون افزایش عرض باند مستلزم تغییر لینک ها و جایگزین نود ها است با نودهای پرسرعت و این هزینه ارتقای شبکه رو در بر خواهد داشت چالشی که اینجا وجود داره این است که ما چقدر باید این عرض باند رو زیاد بکنیم و کی این کافی خواهد شد و چالش بعدی --> وقتی عرض باند رو زیاد می کنیم بعد از یه مدتی دوباره ریت ورودی زیاد میشه پس چقدر ما خوب می تونیم پیش بینی بکنیم که در آینده چقدر عرض باند مورد نیاز خواهد بود و براساس اون بیایم و عرض باند رو افزایش بدیم

دیدگاه اول:

این است که همین شبکه های **best effort** رو چی کار می تونیم بکنیم که وضعیت بهتری برای اپلیکیشن مالتی مدیا پیدا بکنیم

در شبکه های موجود چطوری میشه وضعیت رو بهتر کرد از نظر تاخیر یا لاس؟

در طی چند سال گذشته سرویس های مالتی مدیا تا حد زیادی روی شبکه قرار گرفته اند و مورد استفاده واقع شده اند مثل یوتیوب و اسکایپ

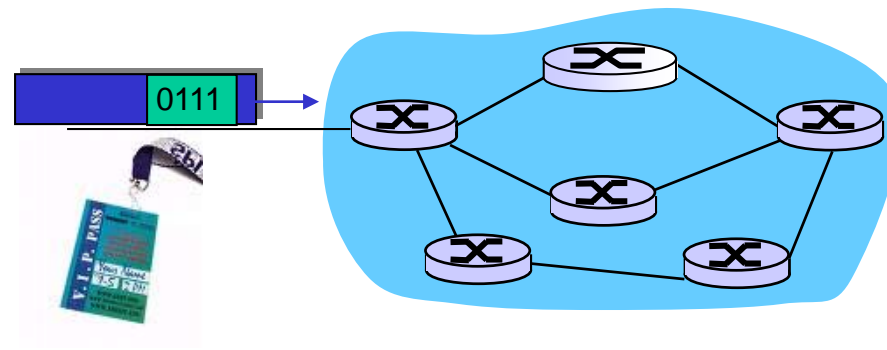
اگر عرض باند شبکه خوب باشه ما کیفیت بهتری رو تجربه میکنیم --> نکته کلیدی برای این دیدگاه است پس برای اینکه کیفیت سرویسمون خوب باشه عرض باند شبکه رو بالا می بریم و اگر ظرفیت شبکه به اندازه کافی بالا باشه تاخیرها کم میشه و پکت لاس هم کم میشه
ارتباط تاخیر شبکه با لود ترافیک روی شبکه --> عکس:

منحنی تاخیر بر حسب لود شبکه رفتارش به این ترتیب بود که در لودهای پایین تاخیر خیلی کم بود و با افزایش لود تاخیر به مقدار افزایش پیدا می کرد با یک شیب کمی و تا یک حدی و از اون حد به بالاتر رشد تاخیر خیلی بیشتر میشد با افزایش لود تا جایی که حتی با کوچکترین افزایش لود تاخیرمون تا حد خیلی زیادی افزایش پیدا میکرد --> در این منحنی لود رو نسبت نرخ ترافیک ورودی به یک صف که با لندا نشون میدیم و نرخ سرویس صف که اونو با میو نشون میدیم --> در یک صف خروجی در یک پورت خروجی در یک روتر این تقریباً معادل این است که ریت ترافیکمون چقدر است یعنی R و عرض باند لینکمون چقدر است یعنی N پس نسبت ریت ترافیک به عرض باند تعیین کننده لود میشه
نکته: اگر ریت متوسط ورودی نزدیک عرض باند بشه یا بیشتر بشه تاخیرمون به بی نهایت میل خواهد کرد و برای اینکه این اتفاق نیوفته ریت متوسطمون باید از عرض باند کمتر باشه پس ما اینجا یک نقطه عطفی داریم + که اگر از این نقطه + لودمون بیشتر باشه تاخیرمون خیلی زیاد خواهد شد و اگر از این نقطه کمتر باشه تاخیرمون کمتر خواهد شد

پس اگر نسبت ترافیک ورودی به عرض باند رو پایین نگه داریم در این صورت در نقطه پایینتری قرار خواهیم گرفت و هرچی این کمتر باشه ما نقطمون پایین تر خواهد بود پس تاخیرمون هم کمتر است با گسترش اینترنت حجم ترافیک روی اینترنت افزایش پیدا میکنه و در تمام نودی های شبکه و تمام روترها ریت ترافیکمون داره افزایش پیدا میکنه در این صورت برای اینکه تاخیر افزایش پیدا نکنه کاری که باید بکنیم این است که این عرض باند هم افزایش بدیم پس در این دیدگاه راه حل این است که ما عرض باند شبکه رو افزایش بدیم

Providing multiple classes of service

- thus far: making the best of best effort service
 - one-size fits all service model
- alternative: multiple classes of service
 - partition traffic into classes
 - network treats different classes of traffic differently (analogy: VIP service versus regular service)
- granularity: differential service among multiple classes, **not among individual connections**
- history: ToS bits

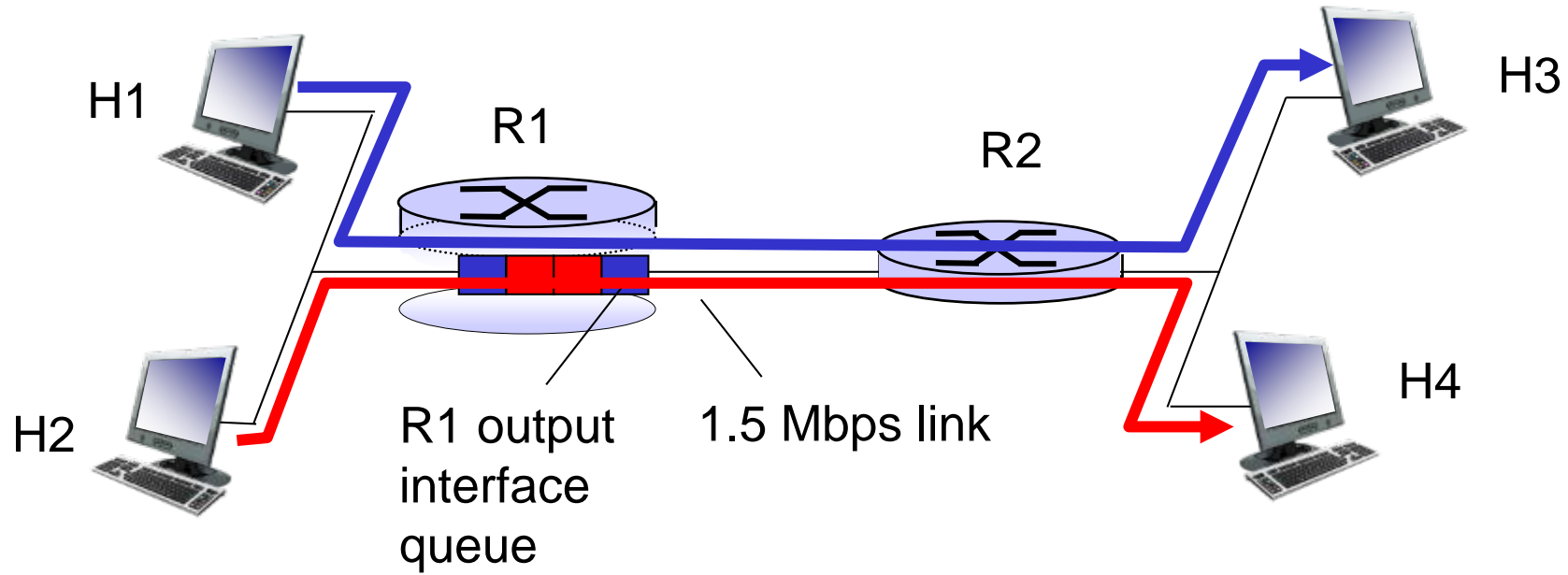


دیدگاه اولی که مطرح کردیم این بود که ما در همین شبکه **best effort** بیایم به طور عمومی وضعیت رو بهتر بکنیم و راه حل این بود که عرض باند رو افزایش بدیم و نکته ای که وجود داره این است که این راه حل، راه حلی است که برای همه ترافیک ها خوبه ینی همه ترافیک ها از این شرایط جدید بهره مند می شن و وضعیتشون بهتر میشه

دیدگاه دیگری که وجود داره این است که: بیشتر بر این است که با عرض باند موجود بدون هزینه کردن روی عرض باند ما چگونه می تونیم وضعیت بهتری رو برای ترافیک های مورد نظرمون ایجاد بکنیم؟ اینجا ترافیکمون مالتی مدیا است --> برای این منظور ما ابتدا میایم ترافیک هارو تفکیک میکنیم به ترافیک هایی که کیفیت سرویس براشون مهم است و ترافیک هایی که تاخیر و لاس براشون مهم است و ترافیک هایی که اینها براشون اهمیت نداره و بعد سعی میکنیم اون ترافیکی که کیفیت سرویس براش مهم است شرایط بهتری پیدا بکنه و این رو با برخورد متفاوتی که با این ترافیک ها در مقایسه با ترافیک های معمولی انجام میدیم این است که سعی می کنیم که وضعیتشون رو بهتر بکنیم این مشابه سرویس **VIP** است در خیلی از جاها در مقایسه با سرویس معمولی پس در فرودگاه ها مسافرهای **VIP** در صف جداگانه ای قرار میگیرن و چون معمولاً تعدادشون کم است خیلی سریع سرویس می گیرن و مسافرهای معمولی که تعدادشون زیاد است پس در صف های طولانی می ایستن تا سرویس مورد نظرشون رو بگیرن

در مورد اینکه چجوری این کار می تونه انجام بشه این مستلزم پیاده شدن یکسری مکانیزیم هایی است که این مکانیزیم ها یک مقداری پیچیدگی به شبکه اضافه میکنه ولی همچین کاری رو انجام میده که کار مفیدی است برای مثال در شبکه برای ترافیک ویس که تاخیر براش مهم است می تونیم همچین کاری بکنیم و کاری بکنیم که ترافیک ویس توی صف قرار نگیره و بیاد جلوی صف در این روش این تفکیک رو براساس نوع سرویس انجام میدیم نه؟؟24؟؟ به عبارت دیگر برای اون مثال ترافیک ماشین ها ما این تفکیک رو برای ماشین های آتش نشانی، پلیس و امبولانس و همه ماشین هایی از این نوع این ها رو یک جور باهاشون برخورد میکنیم نمیایم به ازای هر کدوم جداگانه کار جداگانه ای انجام بدیم توی شبکه هم همینطور برای همه ترافیک های ویس ما یک کار متفاوتی انجام میدیم نه اینکه برای هر کال و هر کانکشن ویسی که برقرار بشه کار جداگانه ای انجام بدیم

Multiple classes of service: scenario



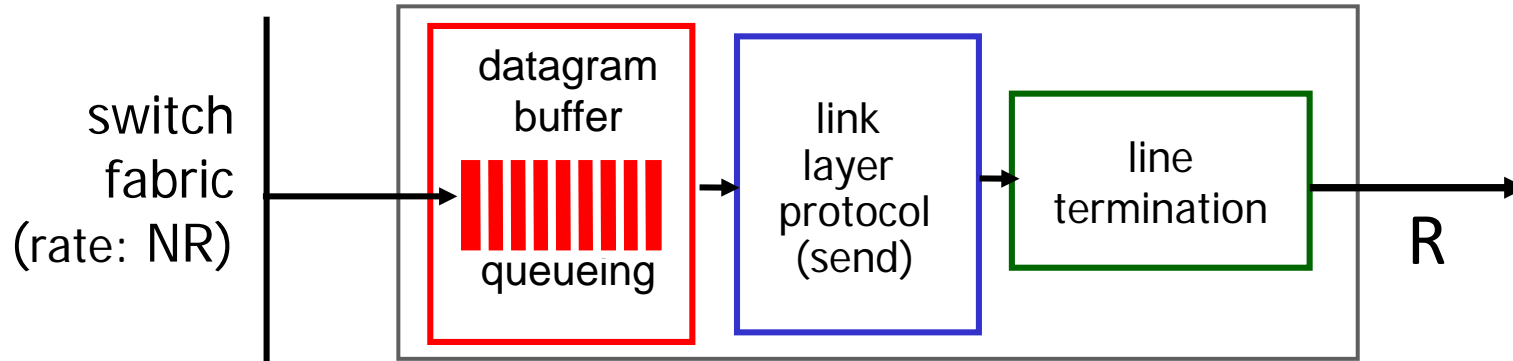
یک سناریوی ساده:

فرض میکنیم 4 تا هاست رو از طریق شبکه بهم متصل کرده ایم و در این سناریو ما دوتا ترافیک رو بین هاست 1 و هاست 3 و هاست 2 و هاست 4 از طریق روتر 1 و 2 یک مسیری توی شبکه برقرار کرده ایم

اینجا فرض میکنیم این ها دوتا ترافیک از دو نوع مختلف هستن

ادامش صفحه 13...

Output port queuing



This is a really important slide

- **Buffering** required when datagrams arrive from fabric faster than link transmission rate. **Drop policy**: which datagrams to drop if no free buffers?
- **Scheduling discipline** chooses among queued datagrams for transmission



Datagrams can be lost due to congestion, lack of buffers



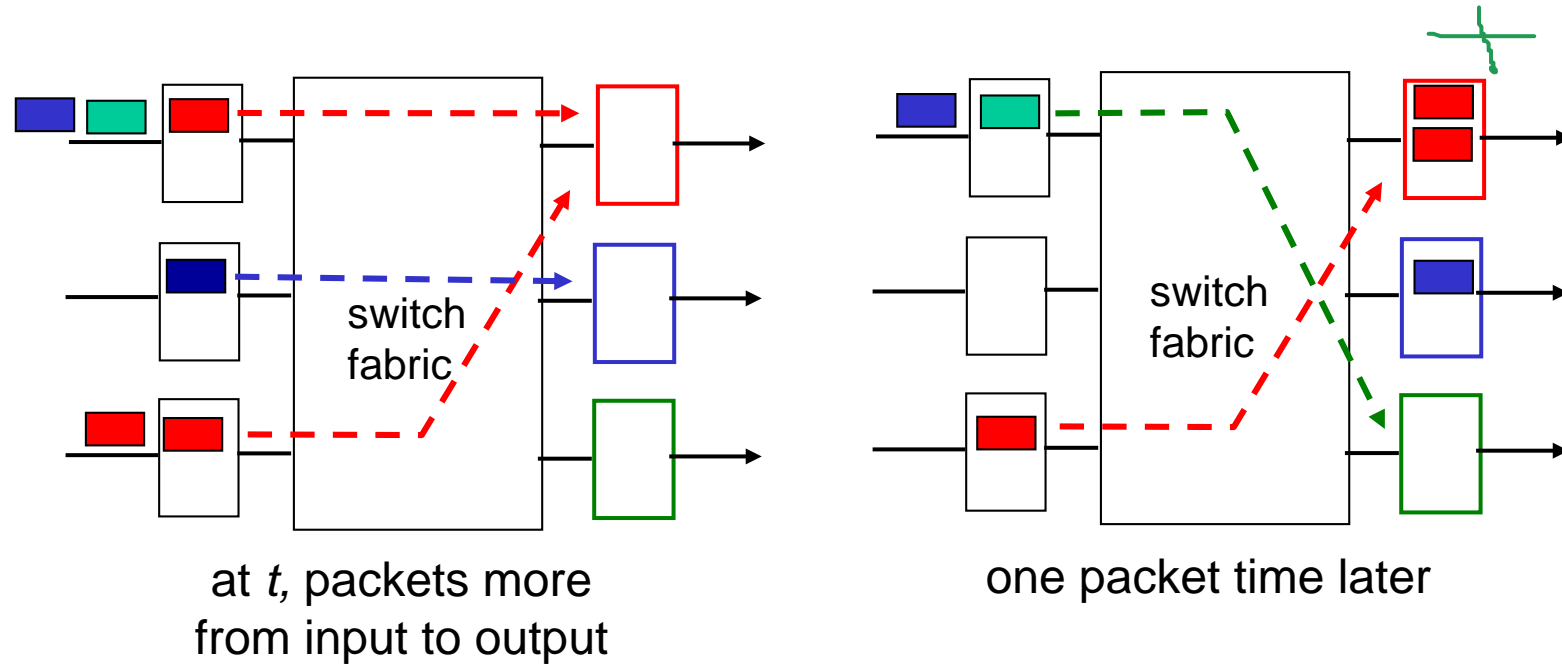
Priority scheduling – who gets best performance, network neutrality

نکته ای که الان می خواهیم بهش اشاره بکنیم این است که بافر پورت خروجی مون بسته ها توش قرار میگیرن و به صورت یک صف قرار میدیم و این صف FIFO است
این بسته هایی که توی این صف قرار میگیرن اگر صف طولانی بشه منجر به این میشه که تاخیرها افزایش پیدا میکنه و اگر صف خیلی زیاد بشه و بافر پر بشه بسته ها drop میشن و این منجر به پکت لاس میشه

برای اینکه کیفیت سرویس رو برای ترافیک مورد نظرمون تامین بکنیم میایم یک صف جداگانه براش در نظر میگیریم مثلاً برای مثال توی اسلاید 15 برای ویس یک صف جدا تشکیل می دیم و برای دیتا هم یک صف جدا دیگه --> به این ترتیب این امکان رو فراهم می کنیم که بسته های ویس بتونن قبل از بسته های دیتا سرویس داده بشن

البته این صف های جداگانه می تونه logical باشه و به صورت فیزیکی همه اینا توی بافر قرار میگیرن و از توی بافر سرویس داده میشن ولی اینجا مسئله ای که به وجود میاد این است که وقتی که آخرین بسته توی لاین ارسال شده ینی آخرین بیتش ارسال شد و عملاً لینک آزاد شد حالا ما می تونیم بسته بعدی رو بفرستیم حالا این بسته بعدی رو از کدوم صف باید انتخاب بکنیم؟
به این ترتیب ما باید بین صف ها اولویت قائل بشیم و این سوال مطرح است که اولویت رو به کی بدیم و در هر لحظه کی بتونه از لاین استفاده بکنه و این منجر به این خواهد شد که ظرفیت لاین چگونه باید بین اینها مشارکت داده بشه تا چه حدی و کدوم اولویت داشته باشن به دیگری اینجا انتظار داریم که شبکه ضمن اینکه سعی میکنه به یک سرویسی کیفیت سرویس بهتری بده ولی در عین حال نمی خواد که اون سرویس دیگه رو کنار بذاره و هیچ توجهی به اون نداشته باشه پس یه جور عدالت و بی طرفی باید اینجا رعایت بشه
حالا ایا واقعا همیشه اولویت بدیم به ویسمون نسبت به دیتا ایا این راه حل خوب است یا نه؟

Output port queuing



- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

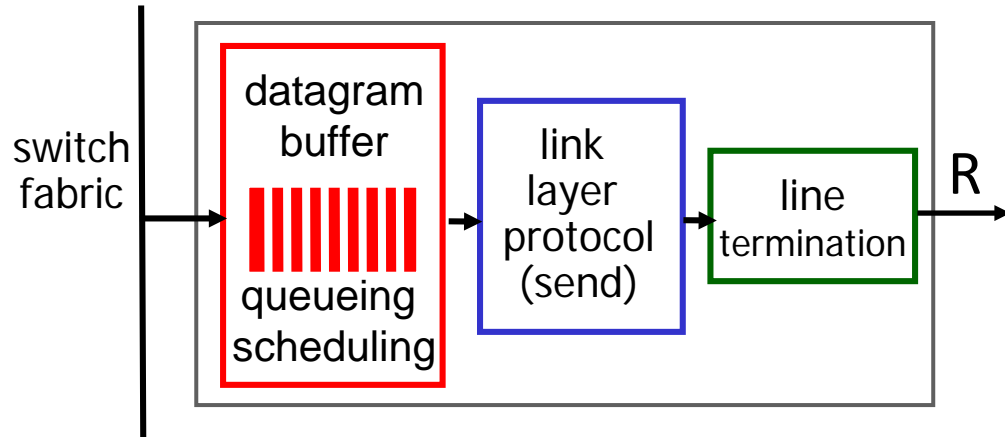
کاری که اینجا اتفاق می افتد این است که بسته هایی که از هر پورت ورودی میان مشخص میشه براساس اینکه مسیرشون چی هست به کدوم پورت خروجی باید ارسال بشن (بحث فورواردینگ) و بعد از طریق سویچینگ فبریک بسته ها به اون پورت خروجی مورد نظر داده میشن

ترافیک های مختلف از ورودی های براساس مسیرشون ممکنه منجر به این بشه که به یک پورت خروجی یکسان منتقل بشن و در نتیجه بسته به این که فبریکمون میتونه همزمان بسته های بیشتری رو به پورت خروجی برسونه یا نه و در عین حال فبریکمون یک چنین قابلیتی رو نداره و در پورت ورودی صف درست میشه و اگر این قابلیت رو داشته باشه در پورت خروجی صف درست میشه به این ترتیب صف تشکیل میشه برای اینکه از بسته هایی که همزمان به یک پورت می رسن همزمان همشون نمی تونن ارسال بشن پس صف داریم و اگر این وضعیت تداوم داشته باشه صف طولانی تر میشه و هر موقع بسته ای برای پورت قرمز نبوده صف شروع میکنه به خالی شدن این صف باعث تاخیر میشه و هرچی صف طولانی تر بشه (+) تاخیرمون بیشتر است و اگر بافرمون پر بشه بسته ها drop میشن و این منجر به لاس میشه

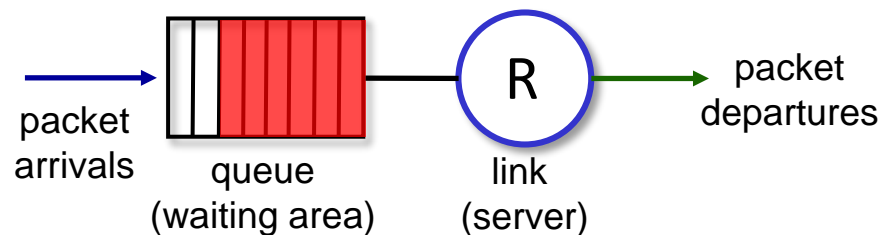
که همه اینارو قبلا گفتیم...

ادامه صفحه قبلی...

Buffer Management



Abstraction: queue



buffer management:

- **drop:** which packet to add, drop when buffers are full
 - **tail drop:** drop arriving packet
 - **priority:** drop/remove on priority basis
- **marking:** which packets to mark to signal congestion (ECN, RED)

Packet Scheduling: FCFS

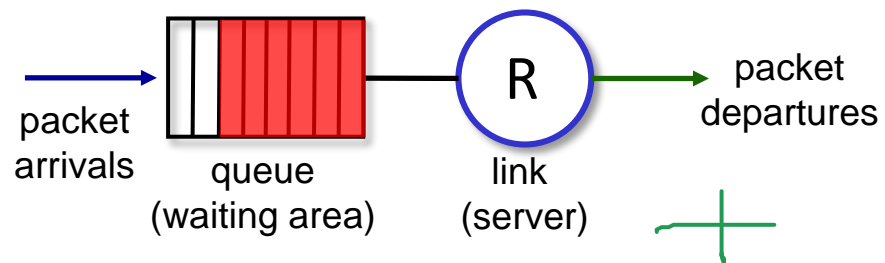
packet scheduling: deciding which packet to send next on link

- first come, first served
- priority
- round robin
- weighted fair queueing

FCFS: packets transmitted in order of arrival to output port

- also known as: First-in-first-out (FIFO)
- real world examples?

Abstraction: queue



در اینجا مدل یک پورت خروجی روتر رو می بینیم + و این مدل از یک صف تشکیل شده و یک سروری که اون پورت خروجی است که بسته ها رو به لینک خروجی می فرسته و بسته هایی که از ورودی های مختلف دریافت میشن در این صف قرار میگیرن و به ترتیب توسط سرور صف به پورت خروجی ارسال میشن و هر موقع آخرین بسته که به پورت خروجی ارسال شده ارسالش تموم شد سرور میاد بسته بعدی که باید ارسال بشه رو از صف بر می داره و می فرسته توی پورت خروجی و لینک خروجی

در حالت عادی ما یک صف داریم که این صف به صورت FIFO سرویس داده میشه ولی قبلا به این نتیجه رسیدیم که برای تامین کیفیت سرویس برای اپلیکیشن های خاص باید اونها رو توی یک صف جداگانه قرار بدیم پس توی مدلمون به جای یک صف می تونیم چند صف داشته باشیم مثلا دو تا صف و بسته های که میاد ابتدا مشخص میشه که به کدوم نوع ترافیک تعلق دارن و بعد بر اون اساس به صف مربوط به اون ترافیک ارسال میشن و از اون طرف هم سرور پورت خروجی رو داریم که بسته ها رو به لینک خروجی ارسال می کنه --> در این حالت این سرور باید انتخاب بکنه که بسته بعدی رو از کدوم صف برداره و به پورت ارسال بکنه

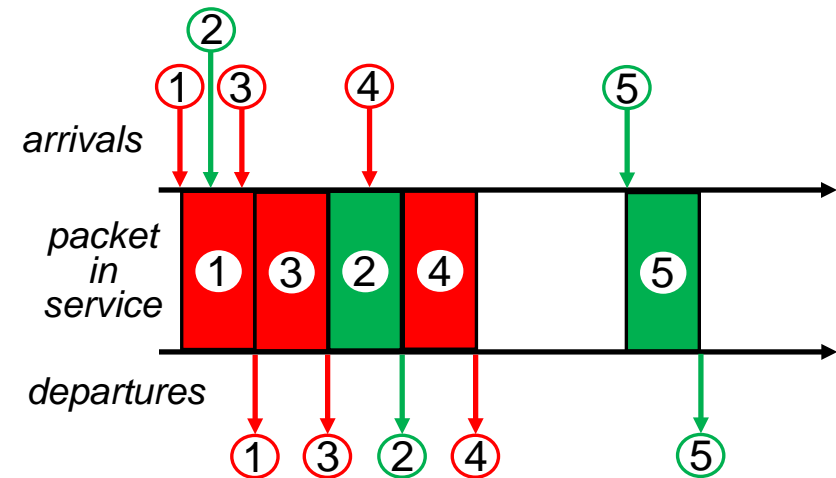
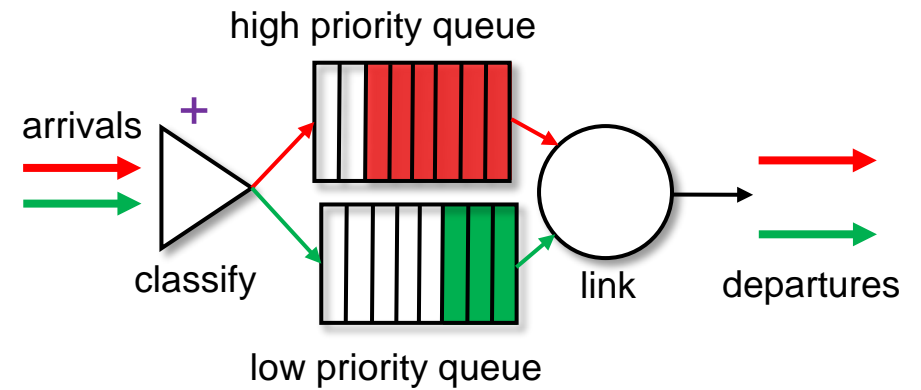
منظور از Scheduling این است که بسته بعدی که به پورت خروجی و به لینک باید ارسال بشه چجوری باید انتخاب بشه و ساده ترین روش Scheduling این است که بسته ها رو به صورت FIFO به پورت خروجی بفرستیم

توی حالت FIFO خیلی اون بحث کیفیت سرویس و تفاوت قائل شدن بین سرویس های مختلف برامون مطرح نیست و اگر اون مطرح باشه الزاما این روش بهترین روش نخواهد بود اینجا یک بحث دیگری هم داریم و اون این است که اگر یک پکت جدید بیاد و بافر پر باشه و صف جای خالی نداشته باشه در این صورت کدوم بسته باید drop بشه؟ ایا اون بسته ای که رسیده باید drop بشه یا نه؟ یا بسته ای که در جلوی صف است باید drop بشه؟ یا رندوم drop بکنیم؟

Scheduling policies: priority

Priority scheduling:

- arriving traffic classified, queued by class
 - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
 - FCFS within priority class



اگر یکسری از بسته هایی که به پورت می رسن تاخیر براشون مهم باشه و بقیه براشون مهم نباشه تاخیر در این حالت ما دوتا صف استفاده میکنیم و بسته های حساس به تاخیر رو در یک صف قرار میدیم و بقیه رو در یک صف دیگه و بعد بین این ها فرق می داریم و اونایی که حساس به تاخیر هستن رو سعی میکنیم زودتر سرویس بدیم تا تاخیرشون کمتر بشه، این کار میتونه به این صورت انجام بشه که ما اولویت بذاریم بین صف ها و به ترتیب اولویت صف ها رو سرویس بدیم --> ینی اینجا که دوتا صف داریم صف قرمز صف با اولویت بالاتر ما است و صف سبز صف با اولویت کمتر --> به این ترتیب اگر توی هر دوی این صف ها بسته هایی وجود داشته باشن وقتی که آخرین بسته ای که به لینک ارسال شده آخرین بیتش ارسال شد و لینک آماده شد برای ارسال بسته بعدی پورت خروجیمون میاد و ابتدا میره سراغ صف با اولویت بالاتر و اگر بسته ای توی این صف بود میاد این رو ارسال میکنه ولی اگر خالی بود میره سراغ صف با اولویت پایین تر و یک بسته ای رو از این صف ارسال میکنه

برای چنین راه حلی ما یک بحثی داشتیم و اون این بود که بسته هایی که می رسن به + و باید توی صف گذاشته بشن برای اینکه بتونیم بین این ها تفکیک قائل بشیم باید اینارو بتونیم شناسایی بکنیم و بفهمیم هر کدوم از بسته به کدوم نوع از ترافیک ها تعلق داره --> اینو قبلا گفتیم ممکنه یک علامتی روی بسته ها گذاشته شده باشه که براساس اون این شناسایی بشه و دسته بندی بشه و ممکنه علامت گذاری هم وجود نداشت باشه و صرفا براساس اطلاعاتی که توی هدر بسته ها به طور معمول وجود داره این دسته بندی هم همین جا انجام بشه کدوم اطلاعات در هدر بسته می تونه برای همچین کاری استفاده بشه؟ منطقی ترین اطلاعاتی که می تونه برای این مسئله استفاده بشه port number لایه tcp بسته چون این ارتباط مستقیم با نوع اپلیکیشن داره مثلا برای بسته های voip این ها از پورت نامبر خاصی استفاده میکنن و براساس اون پورت نامبر میشه اینا رو شناسایی کرد و توی صف جدا قرار دارد - در عین حال علاوه بر این ممکنه از ip ادرس مبدا یا مقصد و حتی پروتکل نامبر اینکه پکت tcp است یا udp اینا هم ممکنه استفاده کرد یا ترکیبی از این هارو میشه استفاده کرد و براساس اون ها تعیین کرد که بسته به کدوم صف باید وارد بشه

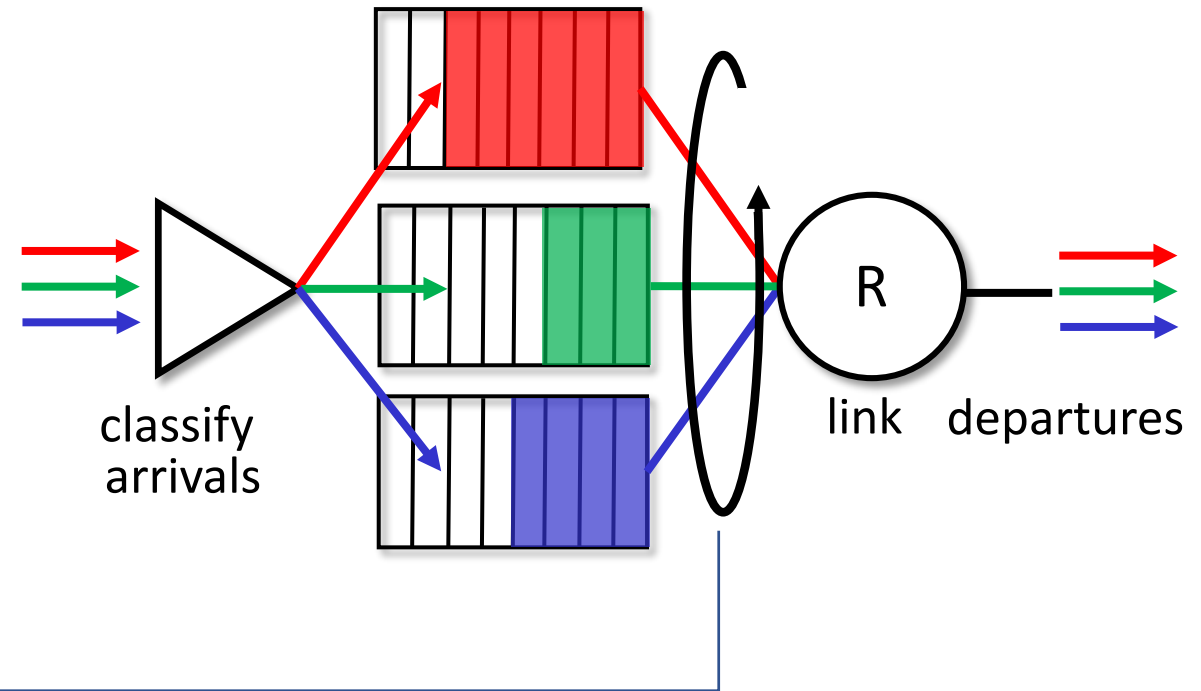
مثال: رفتار چنین صفی چگونه خواهد بود؟ فیلم 16 دقیقه 14:06 تا 16:55 دوباره ببین!!

این روش Scheduling بنظر میاد که خواسته ما رو داره برآورده میکنه ینی بسته هایی که توی صف اولویت بالاتر قرار میگیرن بسته هایی صف با اولویت پایین تر مزاحمتی برای اون اولویت بالاتری ها ایجاد نمی کنن پس تاخیر بسته ها توی صف با اولویت بالاتر تاخیرشون فقط تاخیر بسته هایی از نوع خودشون است که جلوترش قرار داره بنظر میاد این روش خیلی ایده ال باشه و اره برای بسته های از این نوع این روش خیلی ایده ال است ولی مشکل این است برای بسته هایی که حساس به تاخیر نیستن و اولویت پایین تر رو پیدا می کنن این روش می تونه خیلی خوب نباشه چون ممکنه ترافیک با اولویت بسته های بالاتر خیلی بالا بره در این صورت صف قرمز همیشه بسته خواهد داشت و اگر این صف همیشه بسته داشته باشه معنیش این است که صف سبز هیچ وقت سرویس نمی گیره

Scheduling policies: round robin

Round Robin (RR) scheduling:

- arriving traffic classified, queued by class
 - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



با در نظر گرفتن این اشکالاتی که گفته شد Scheduling بعدی که میگیریم RR است در روش RR ما برای کلاس های مختلف ترافیکی صف های جداگانه تشکیل میدیم مثلا دوتا صف و بعد بین این دوتا صف ها نوبت می داریم و بعد به نوبت سرویس می دیم مثال:

همون مثال قبلی رو داریم برخلاف روش قبلی که پکت 2 ارسال شده بود اینجا پکت 3 هست که انتخاب میشه و بعد به پورت خروجی ارسال میشه چرا؟ اینجا ما به نوبت صف ها رو سرویس میدیم <-- صف اولویت بالاتر سرویسش رو گرفته حالا نوبت صف اولویت پایین تر است پس 3 انتخاب میشه و بعد 2 انتخاب میشه و وقتی که 2 تمام شد عیار غم اینکه الان نوبت صف با اولویت کمتر است چون صف با اولویت پایین تر خالی است پس بسته بعدی از صف با اولویت بالاتر ارسال میشه این روش Scheduling چه ویژگی هایی داره؟

در این روش Scheduling باز اینجا دوتا صف فرض میکنیم و ریت لینک خروجی هم Rbps است و هر دو صف همیشه پکت برای ارسال دارن و بسته ها هم، هم اندازه هستن (اینا فرض هامون هستن) <-- یا این شرایطی که گفتیم این روش Scheduling به چه چیزی منجر میشه؟

نتیجه ای که حاصل میشه این است که این عرض باند لینک خروجی به طور مساوی بین این ها تقسیم میشه یعنی $R/2$ به صف اول می رسه و $R/2$ به صف دوم پس RR میتونیم عرض باند رو به نسبت مساوی بین دوتا صف تقسیم بکنیم و اگر تعداد صف ها سه تا بود میشد $R/3$ و... <-- حالا در شرایط واقعی بسته هامون اندازه هاشون مثل هم هستن در نتیجه چون بر حسب بسته داریم عرض باند رو تقسیم میکنیم و یک بسته تا شروع شد تا تمام نشده بسته بعدی نمی تونه ارسال بشه در نتیجه اینجا ما اگر یک درمیان اینارو بفرستیم صف با اولویت پایین تر احتمالا عرض باند بیشتری استفاده میکنه چون بسته هاش اندازه هاشون بزرگتره معمولا پس تعیین این که هر کدوم از ترافیک ها چه نسبتی از عرض باند رو استفاده بکنن به اندازه بسته هاشون هم ربط پیدا میکنه و مسئله بعدی هم این است که وقتی که یکی از این صف ها خالی باشه به اون صف سرویس داده نمیشه و از اون سهم عرض باندش استفاده نمیشه ولی ذخیره هم نمیشه پس در این حالت وقتی که بسته 2 ارسالش تموم شد عیار غم اینکه نوبت صف با اولویت پایین تر ما بسته ای با صف اولویت بالاتر رو فرستادیم چون اونجا صف خالی بود یعنی نوبت اون رو دادیم به صف اولویت بالاتر که این البته ویژگی خوبی است چو باعث میشه عرض باند تلف نشه

Scheduling policies: weighted fair queueing

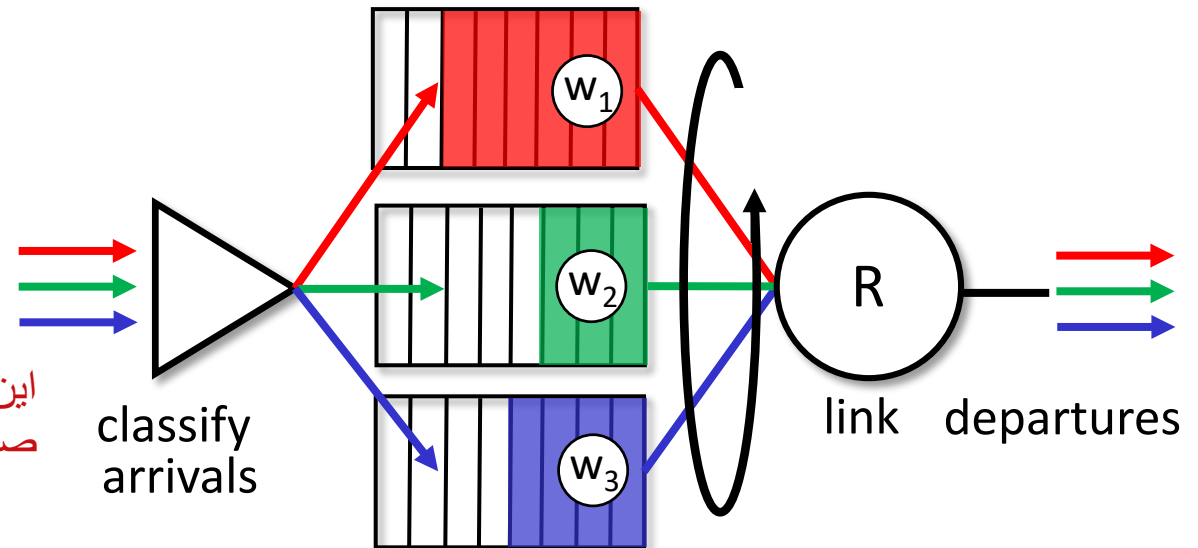
Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class, i , has weight, w_i , and gets weighted amount of service in each cycle:

$$\frac{w_i}{\sum_j w_j}$$

این فرمول در حالت کلی برای n صف است

- minimum bandwidth guarantee (per-traffic-class)



اگر بخوایم این درصدی از عرض باند رو که برای هر صف می رسه رو مشخص بکنیم و دقیقاً همون عرض باند رو بهش بدیم پس از روش WFQ به عنوان یک روش تعمیم یافته ای از RR می تونه این کارو انجام بده

در روش اسکجولینگ WFQ ما دقیقاً عرض باند لینک خروجی رو به نسبت دلخواه بین صف ها تقسیم میکنیم <-- اینجا ما سه تا کلاس سرویس داریم و بسته ها در سه صف جداگانه قرار میگیرن و ما میخوایم عرض باند لینک خروجیمون با نسبت تعیین شده ای بین این ها تقسیم بشه و هر کدام سهم مشخصی از اون عرض باند رو استفاده بکنن

روش WFQ به این صورت است که ما برای هر صف یک وزنی تعیین میکنیم مثلاً w_1 برای صف قرمز و.. و بعد به صورت RR این هارو سرویس میدیم

در روش RR گفتیم یک بسته از صف قرمز می فرستیم و بعد یک بسته از صف بعدی و همینطوری میریم جلو و به نوبت بهشون سرویس میدیم حالا در روش WFQ این بسته ها براساس وزن انتخاب میشن

برای ساده تر شدن مسئله فرض میکنیم اندازه بسته های یکسان است و صف ها همیشه پر هستن و عرض باند لینک خروجی هم Rbps است <-- در اینصورت کاری که ما انجام میدیم این است که وقتی که نوبت صف قرمز هست به تعداد w_1 بسته ارسال میکنیم و بعد که نوبت صف سبز هست به تعداد w_2 بسته و بعد برای صف ابی به تعداد w_3 بسته و بعد دوباره اینو تکرار میکنیم در این صورت هر یک از صف ها چه بخشی از عرض باند لینک خروجی رو می گیرن؟

سهم صف قرمز <-- $w_1/w_1+w_2+w_3$ در R

سهم صف سبز <-- $w_2/w_1+w_2+w_3$ در R

سهم صف ابی <-- $w_3/w_1+w_2+w_3$ در R

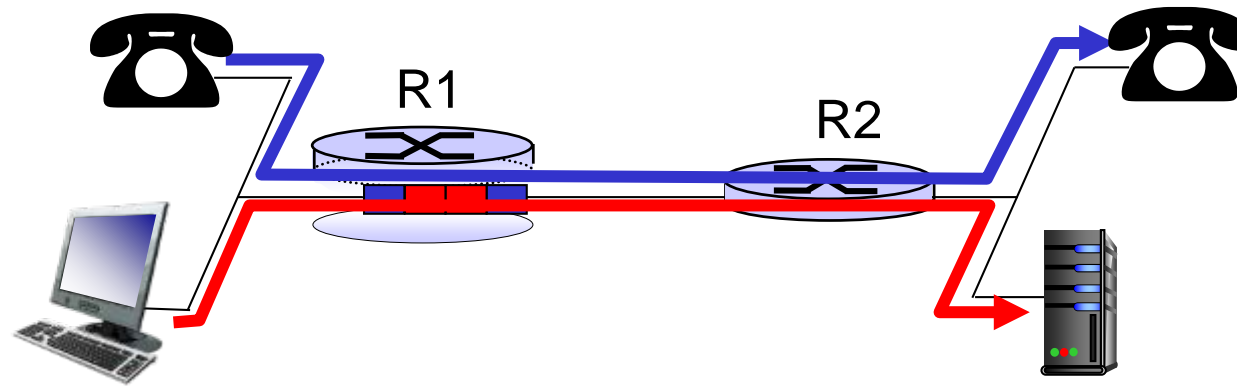
در اینجا اگر $w_1=w_2=w_3=1$ باشه این میشه همون حالت RR و به نسبت مساوی بین این ها تقسیم میشه عرض باند

برای مثال اگر w_1 یک بسته باشه و w_2 دو بسته و w_3 سه بسته در این صورت سهم صف اولمون $1/6R$ خواهد بود و $2/6R$ برای صف دوم و $3/6R$ برای صف سوم

مسئله ای که داریم این است که اندازه بسته ها در شرایط واقعی یکسان نیست چه بین کلاس های مختلف و چه داخل یک کلاس پس به این ترتیب اگر این اسکولینگ رو براساس تعداد بسته ها انجام بدیم در نهایت درصدی عرض باندی که واقعا برای هر صف می رسه متفاوت خواهد بود --> این راه حلش چی هست؟ ما در هر راند به جای اینکه بسته ها رو انتخاب بکنیم و بفرستیم متناسب با این نسبت بیت رو انتخاب میکنیم و می فرستیم به عبارت دیگر در نوبت صف قرمز ما W1 بیت می فرستیم و در صف سبز W2 بیت و ابی هم W3 بیت و نکته ای که وجود دارد این است که ما بسته ها رو به پورت خروجی می فرستیم نه بیت ها رو پس چجوری می تونه همچین اتفاقی بیوفته؟ این کار رو به صورت مجازی انجام میدیم ینی ما بعد از صف ها شمارنده بذاریم و به تعداد بیت مورد نظر در هر راند به این شمارنده اضافه بکنیم و به همین ترتیب برای صف بعدی و... و اینو چک بکنیم که هر موقع اندازه ی شمارنده مساوی یا بیشتر از اندازه تعداد بیت های بسته سر صف بود اون وقت اون بسته رو ارسال بکنیم پس در حالت کلی ما بیت به بیت با این وزن ها توزیع میکنیم ولی به صورت مجازی و وقتی که به تعداد کافی بیت برای یک صف تخصیص داده شده به صورت واقعی اون رو ارسال میکنیم

Scenario 1: mixed HTTP and VoIP

- example: 1 Mbps VoIP, HTTP share 1.5 Mbps link.
 - HTTP bursts can congest router, cause audio loss
 - want to give priority to audio over HTTP



Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

ادامش...

ترافیک اولمون فرض میکنیم ترافیک ویس است که یک ترافیک ریل تایم حساس به تاخیر است و ترافیک دوممون ترافیک وب است ینی http که ریل تایم به اون صورت نیست و تاخیر هر چند برایش مهم است ولی حساس به تاخیر در حد ویس نیست

حالا فرض میکنیم این لینک 1.5Mbps رو اینا با هم شیر می کنن

و اگر توی روتر R1 ترافیک ورودی زیاد بشه هر دوی این ترافیک ها در پورت خروجی R1 که به لینک منتهی میشه بافر می شن و صف تشکیل میشه و اینها توی صف قرار میگیرن --> پس ترافیک های مختلف اینجا پشت سر هم توی یک صف قرار می گیرن و فرقی بین اینها قائل نمیشیم در حالت عادی پس اگر صف طولانی بشه و تاخیر زیاد بشه همه این ها به یک شکل تاخیر پیدا می کنن

ضمن اینکه هر دوی این ها توی یک صف مشترک قرار می گیرن ولی شرایط ترافیکی این ها یکسان نیست مثلا ترافیک voip یک ترافیک پریودیک است با فواصل زمانی نسبتا منظم و با بسته های

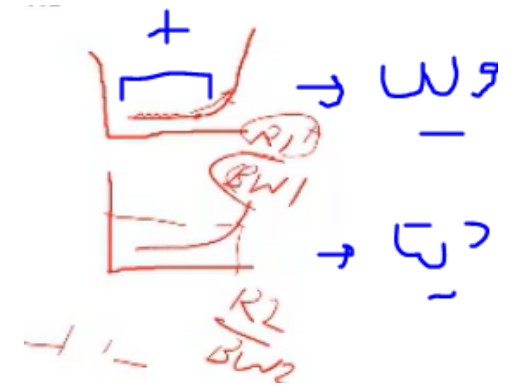
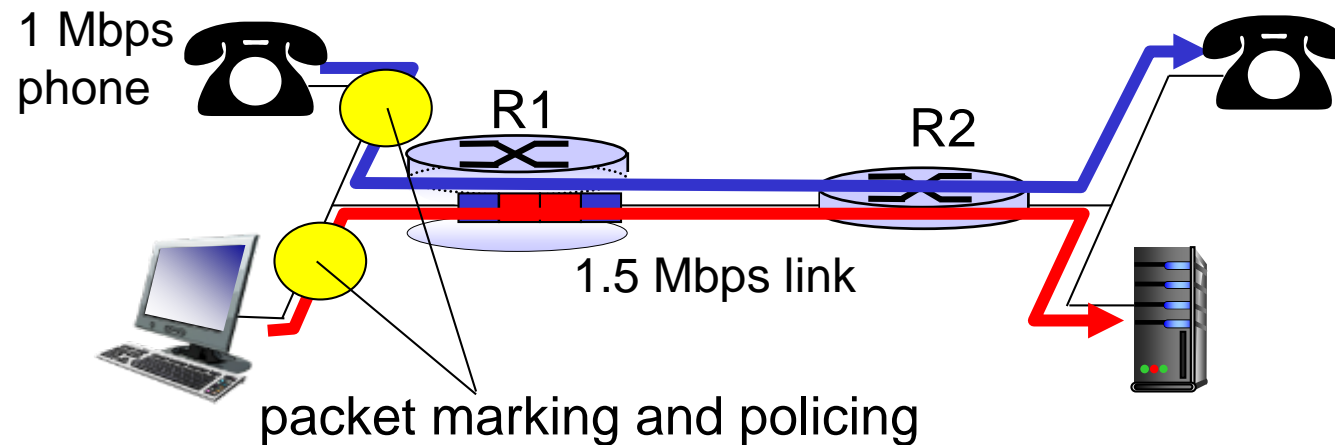
کوچیک و برعکس اون ترافیک دیتا اولاً منظم نیست و ثانياً bursty است پس ما در لابه لای پکت های منظم و کوچیک ویسمون ممکنه پکت های بزرگ و تعداد زیادی از اونها رو از دیتا داشته باشیم در نتیجه تاخیری که تحمیل میشه برای بسته های ویس به این ترتیب می تونه خیلی زیاد بشه

توی این دیدگاهی که اشاره کردیم باید بیایم این صف ها رو از هم تفکیک بکنیم ینی بسته های ویس بیان جلوی صف پس این مستلزم این است که برای اون ها یک صف جداگانه ای تشکیل بشه پس بافرینگی که انجام میدیم توش دوتا صف شکل میدیم یک صف برای ترافیک voip و یک صف برای ترافیک دیتا که به این ترتیب بتونیم بین این ها فرق قائل بشیم و بتونیم اینها رو از هم جدا بکنیم

این جدا کردن یکسری ملزوماتی هم داره و اون این است که ما به نوعی ما یک معیاری داشته باشیم که بتونیم براساس اون پکت های ترافیک voip و ترافیک وبمون رو از هم تفکیک بکنیم ینی به یک شکلی این ها باید علامت گذاری شده باشن و علامت داشته باشن که براساس اون بتونیم اینارو شناسایی بکنیم پس اولین اصلی که بهش می رسیم این است که بسته های ترافیک های کلاس های مختلف باید به نوعی علامت گذاری بشن و به نوعی شناسایی بشن و براساس این شناسایی تفکیک بشن و در صف های جداگانه قرار داده بشن

Principles for QOS guarantees (more)

- what if applications misbehave (VoIP sends higher than declared rate)
 - policing: force source adherence to bandwidth allocations
- *marking, policing* at network edge



Principle 2

provide protection (isolation) for one class from others

فرض میکنیم ما یک صف جداگانه برای ترافیک ویس ایجاد کردیم و بسته های ویس رو به این ترتیب می‌تونیم ببریم جلوی صف و زودتر از بسته های دیتا سرویس بدهیم به این ترتیب تاخیر این ها کمتر میشه در نتیجه کیفیت سرویس اپلیکیشن voip مون برآورده میشه

نتیجه ای که از این کار می‌گیریم می‌تونه نقض بشه اگر چنانچه ترافیک voip که به این ترتیب ما داریم بهش سرویس بهتری میدیم افزایش پیدا بکنه و بیشتر از اون عرض باندی که بهش اختصاص دادیم بشه --> به این ترتیب منحنی هایی که داشتیم ما اومدیم بخشی از عرض باند رو برای ترافیک voip تخصیص دادیم پس اینجا دوتا منحنی داریم یکی برای ترافیک ویس و یکی ترافیک دیتا و اینجا لودمون voip --> ترافیک voip است روی عرض باندی که برای voip در نظر گرفتیم و برای دیتا لودمون ترافیک دیتا است روی عرض باندی که برای دیتا در نظر گرفتیم

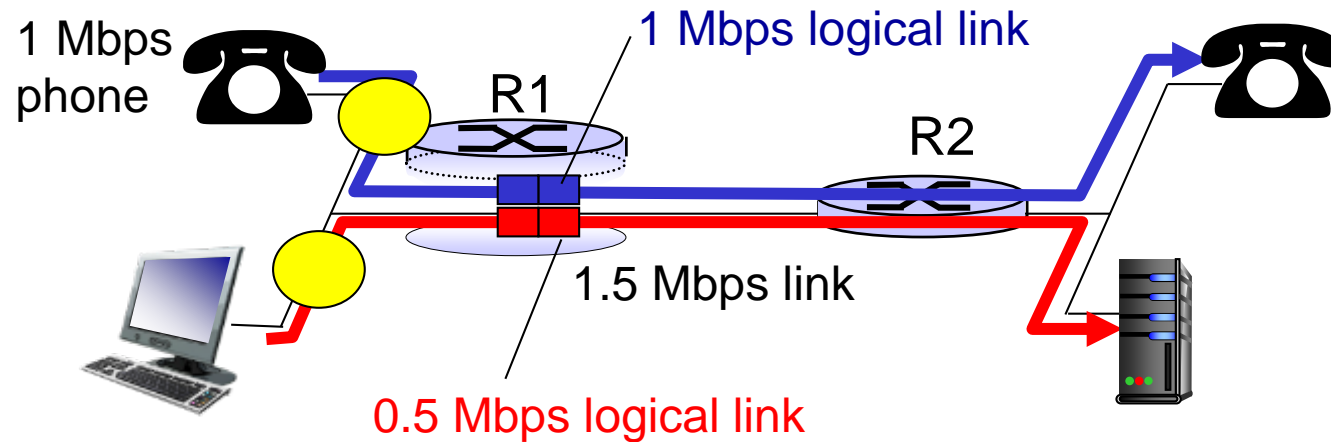
به این ترتیب اگر عرض باند رو اختصاص بدیم به هر کدام از این ها هر کدام جداگانه براساس نسبت ترافیک به عرض باندشون تاخیرشون تعیین میشه به این ترتیب می‌تونیم برای ویس ریت ویس رو نسبت به عرض باندی که برای ویس اختصاص دادیم پایین بیاریم به طوری که در یک چنین قسمت هایی + از این منحنی باشیم در شبکه و برای دیتا ممکنه ریت بیشتری رو بذاریم نسبت به عرض باندی که براش تخصیص دادیم به این ترتیب می‌تونیم ترافیک بیشتری رو منتقل بکنیم چون اینجا تاخیر خیلی مهم نیست مشکلی که پیش میاد این است که ما یک عرض باندی رو برای ویس اختصاص دادیم براساس اینکه ترافیک ورودی R1 است و به این ترتیب در این نقطه ها + قرار داریم حالا اگر این ترافیک R1 افزایش پیدا بکنه در این صورت توی منحنی میایم جلوتر و می‌ریم به نقاط بالاتر و تاخیرمون بیشتر میشه برای اینکه بتونیم این تاخیر رو تضمین بکنیم که بیشتر نخواهد شد لازمش این است که تضمین بکنیم که نرخ ترافیک ورودی voip از اون حدی که مورد انتظار است و براساس اون عرض باند اختصاص دادیم بیشتر نخواهد شد این مستلزم یک نوع کنترل روی ترافیک ورودی است و این لازمه دیگری است که ما اگر بخوایم یک چنین دیدگاهی رو توی شبکه پیاده بکنیم حتما باید این رو هم در نظر بگیریم به این میگیم policing پس علاوه بر marking که اشاره کردیم

که برای تفکیک بسته ها باید استفاده بشه ما policing رو هم باید داشته باشیم برای اینکه بتونیم ترافیک ورودی رو کنترل بکنیم و اجازه ندهیم از حد بیشتر بشه

این policing رو چگونه می‌شه انجام داد توی شبکه؟ این باز اون مکانیزیمی است که لازمه پیاده بشه برای اینکه بشه چنین کنترل هایی رو اعمال کرد که اینو بعدا می‌گیم

Principles for QOS guarantees (more)

- allocating *fixed* (non-sharable) bandwidth to flow: *inefficient* use of bandwidth if flows doesn't use its allocation



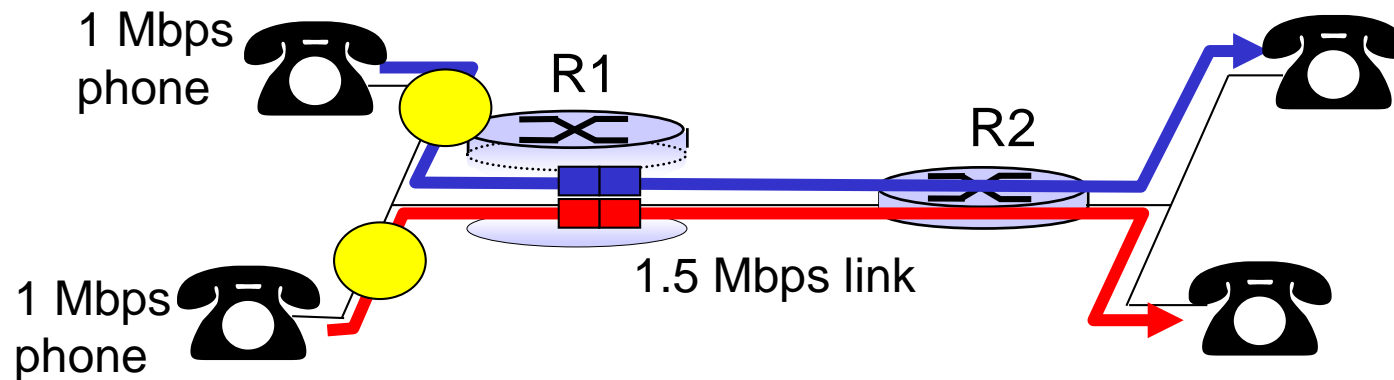
Principle 3

while providing isolation, it is desirable to use resources as efficiently as possible

برای اینکه بتوانیم همچنین تفکیکی قائل بشیم و بعد برای هرکدام جداگانه تاخیر رو مدیریت بکنیم میایم این عرض باند رو تقسیم می کنیم بین این دوتا ترافیک و برای ترافیک voip یک عرض باندی رو اختصاص میدیم و بعد براساس نسبت لود ترافیک به اون عرض باند تاخیر رو سعی میکنیم در حد پایینی نگه داریم ولی تخصیص عرض باند به صورت ثابت و دائمی برای یک نوع ترافیک می تونه مشکل ساز باشه کی این دچار مشکل میشه؟ وقتی که ما ترافیک voip نداشته باشیم <-- ما اومدیم یک بخشی از عرض باند رو اختصاص دادیم به ترافیک voip ولی حالا لحظه ای است که ترافیک voip نداریم کلا و نتیجه اش این میشه که اون قسمت از عرض باندمون عملاً تلف میشه و استفاده مفید نمیشه در حالی که ترافیک دیگرمون هر چند که تاخیر براش مهم نیست ولی به هر حال می تونه از عرض باند استفاده بکنه و عبور بکنه حداقل اینه که میتونه ریت بیشتری می تونه به دست بیاره حتی اگر تاخیری براش مهم نیست میتونه Throughput بیشتری رو به دست بیاره پس به یک اصل سوم هم می رسیم و اون این است که ما این عرض باندی که تفکیک میکنیم باید جوری اینو جدا بکنیم در صورتی که ترافیک نداریم بتونیم اینو استفاده بکنیم برای ترافیک های دیگه <-- این هم باز مستلزم مکانیزیم هایی است

Per-connection QOS guarantees

- *basic fact of life*: can not support traffic demands beyond link capacity



Principle 4

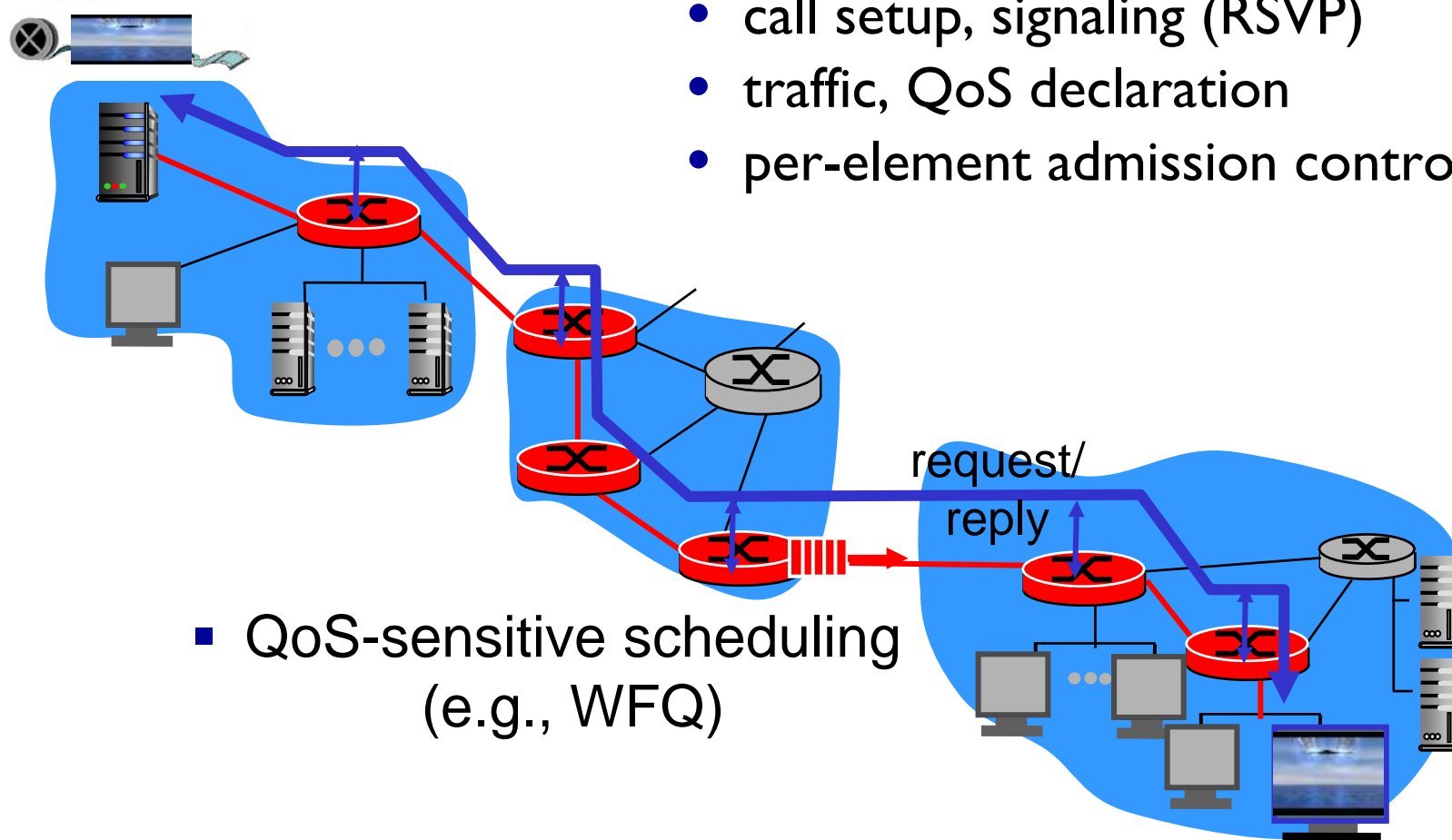
call admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

اینجا ما یک عرض باندی داریم که از provider داریم سرویس میگیریم و تعداد کال های همزمانمون ترافیکی که ایجاد می کنن از این عرض باندی که در نظر گرفته شده بیشتر است حالا اگر ما اینجا این ترافیک اضافی رو بفرستیم توی شبکه providerمون توی اسکجولینگش مثلا 6.5Mbps برای ما تخصیص داده مثلا برای اون 100 تا مکالمه همزمان و اگر تعداد مکالمه همزمان بیشتر باشه و ترافیکش هم ارسال بشه به ISP اینجا این ظرفیت لازم براش فراهم نخواهد بود و provider با یک سیاستی این اضافات رو حذف خواهد کرد و حتی اگر حذف هم نکنه عملا توی صف ترافیک ورودی ریتش می ره بالاتر نسبت به اون ریتی که داره سرویس میده و این تاخیرها رو می بره بالا و کیفیت سرویس نقض میشه واسه همین است که داریم این مکانیزیم هارو داریم پیاده میکنیم که این اتفاق نیوفته --> این مستلزم این است که روی کال ها مدیریت انجام بدیم و اجازه ندیم که کال بعدی اضافه بشه به عبارت دیگر ما باید ترافیکی که وارد میشه رو روش مدیریتی داشته باشیم که این مدیریت Per-connection می تونه اعمال بشه در نتیجه ما کال های اضافه رو اینجا رد خواهیم کرد و اجازه نمی دیم برقرار بشه که به این ها اصولا میگیم call admission ینی ما اینجا یک call admission ضرورت داره که انجام بدیم

call admission: ترافیک جدید رو صرفا با call admission قبول میکنیم و این به این شکل نیست که هر ترافیکی بیاد و اضافه بشه به ترافیک ورودی شبکه و توی مرحله call admission ما بررسی می کنیم که اگر در لینک هامون ظرفیت کافی برای ترافیک جدید نباشه اینو رد میکنیم و صرفا اگر باشه اینو قبول میکنیم و در call admission ما علاوه بر این چکی که انجام میدیم که ایا می تونیم این ترافیک رو قبول بکنیم یا نه ما بقیه توافق هامون هم بین provider و customer انجام میدیم و این ها توافق هایی هستن که لازم است برای اینکه بشه اون گارانتی که برای کیفیت سرویس گفتیم رو شبکه بتونه مسئولیتش رو به عهده بگیره

- *resource reservation*

- call setup, signaling (RSVP)
- traffic, QoS declaration
- per-element admission control



- QoS-sensitive scheduling (e.g., WFQ)

برای یک کانکشن end to end برای یک سرویسی که حساس به تاخیر است مثلا streaming مالتی مدیا ما میخوایم به صورت end to end ینی از مبدا تا مقصد تاخیرمون از یک حد بیشتری نشه

رافیکمون از مبدا تا مقصد از یکسری as رد میشه و هر as یک isp داره اینو مدیریت میکنه پس یکسری isp به عنوان سرویس دهنده داریم که این ها از همدیگر سرویس میگیرن تا در نهایت به مقصد که isp اخری است و ???