

Chapter 3

Transport Layer

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.

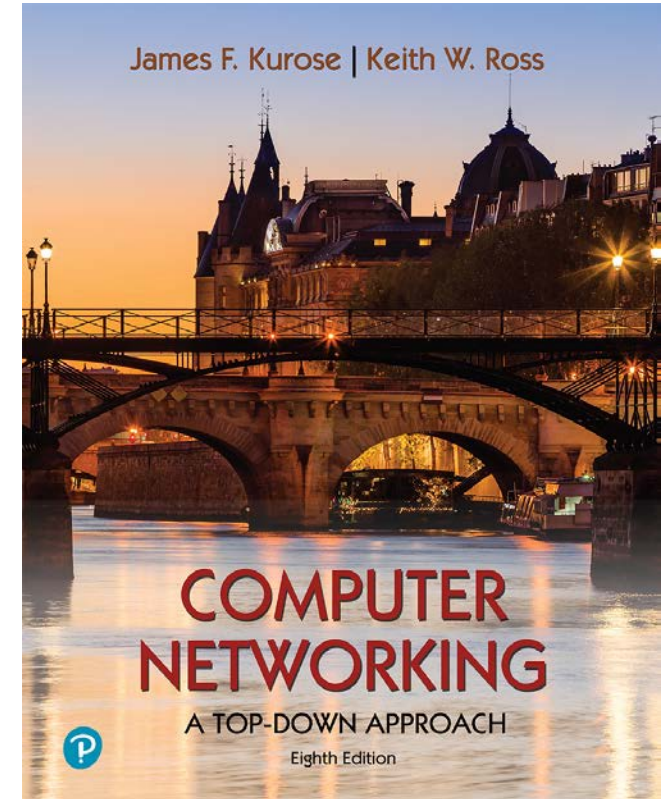
In return for use, we only ask the following:

- 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

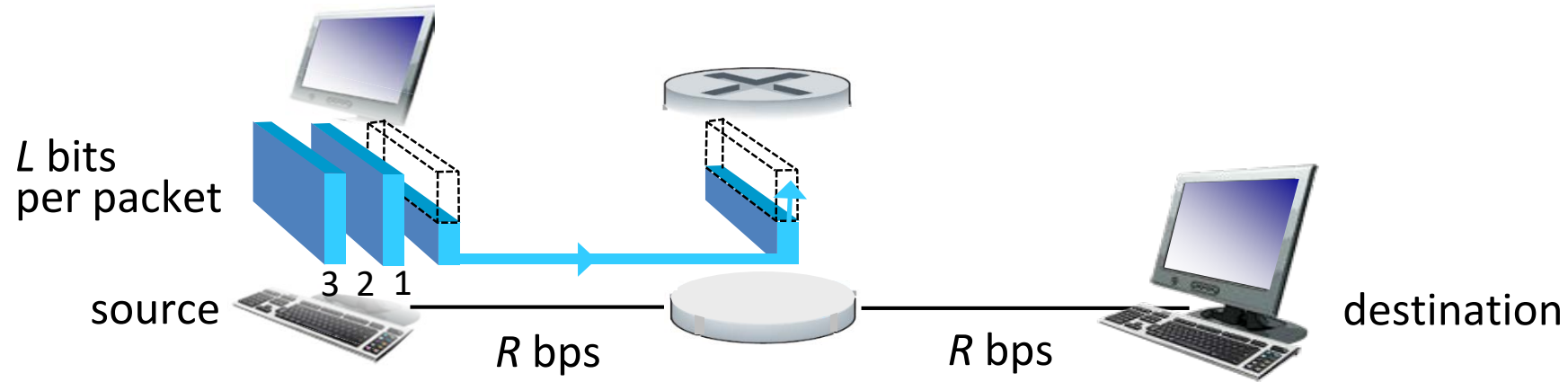


Computer Networking: A Top-Down Approach

8th edition

Jim Kurose, Keith Ross
Pearson, 2020

Packet-switching: store-and-forward



- **packet transmission delay:** takes L/R seconds to transmit (push out) L -bit packet into link at R bps

One-hop numerical example:

- $L = 10$ Kbits
- $R = 100$ Mbps
- one-hop transmission delay = 0.1 msec

انتقال بسته ها در سطح شبکه به صورت store-and-forward انجام میشه ینی در هر نود میانی کل بسته به طور کامل دریافت و ذخیره میشه و بعد به نود بعدی ارسال میشه

این روش ارسال بسته ها باعث ایجاد تاخیر میشه

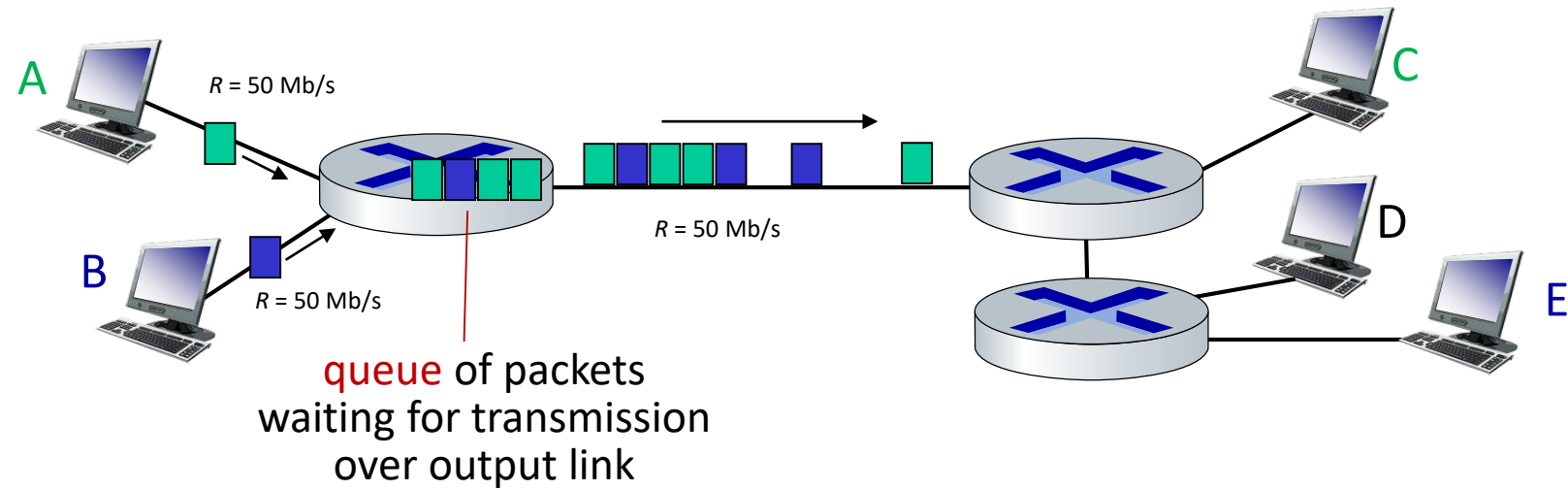
ارسال هر بسته از نود به لینک خروجی براساس طول بسته و طول ریت نرخ خروجی مدت زمانی طول می کشه

برای مثال اگر طول بسته L باشه و سرعت لینک خروجی R باشه L/R ثانیه طول میکشه که بسته به لینک خروجی ارسال بشه

مثلا توی مثالی که زده 5 ثانیه طول می کشه تا از اولین بیت تا آخرین بیت بسته به لینک خروجی ارسال بشه

حالا اگر در طول مسیر ما تعداد نود بیشتری داشته باشیم برای مثال ما اینجا دوتا نود داریم یک نود ابتدایی و یک نود میانی پس در کل میشه $L/R + L/R$ -- پس این میشه تاخیر ارسال بسته به لینک هست

Packet-switching: queueing



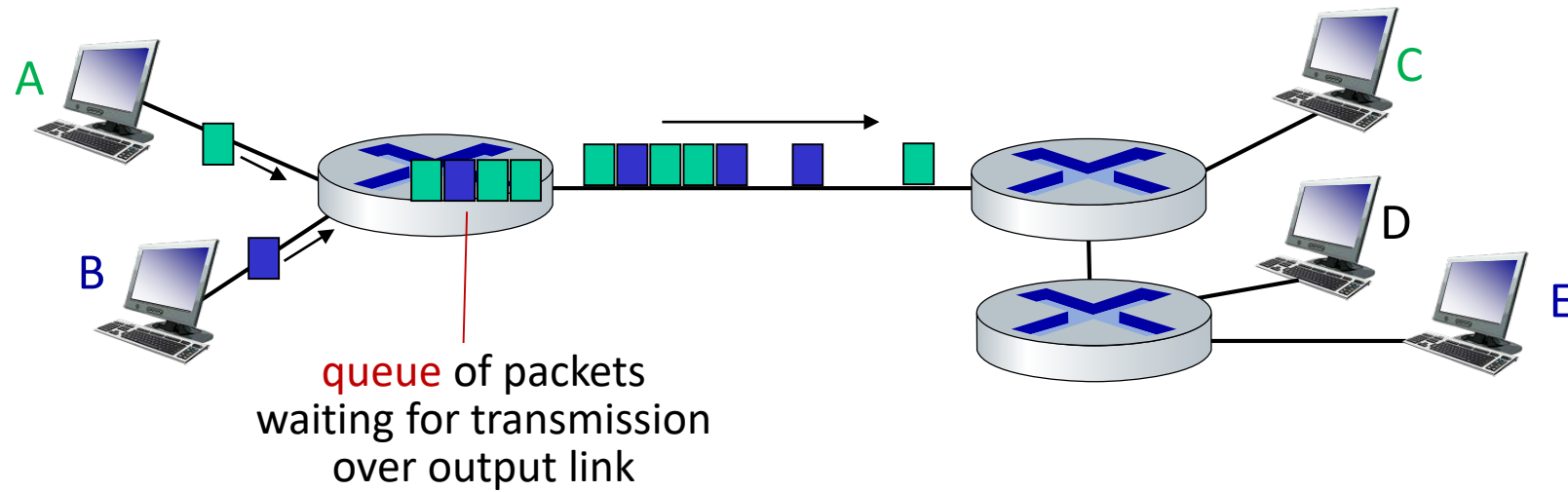
Queueing occurs when work arrives faster than it can be serviced:



-

تاخیر صف: این برخلاف تاخیرهای دیگر می تونه مقدارش خیلی زیاد باشه
در یک نود بین راه بسته ها می رسن و براساس جدول فوروارد به لینک خروجی ارسال هدایت
میشن
اگر ظرفیت بافر پر بشه بسته هایی که میان drop میشن

Packet-switching: queueing



Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

تاخیر و پکت لاس چجوری اتفاق می افتد؟

بسته هایی که از پورت های مختلف وارد میشن و مقصدشون طوری است که به یک لینک مشترک ارسال میشن تعداد این ها اگر زیاد بشه توی بافر صف می بندن

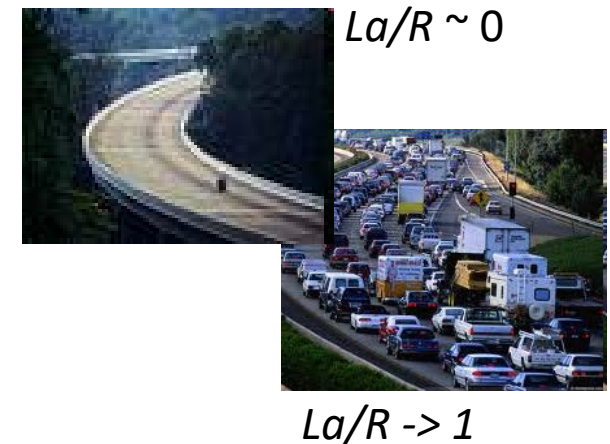
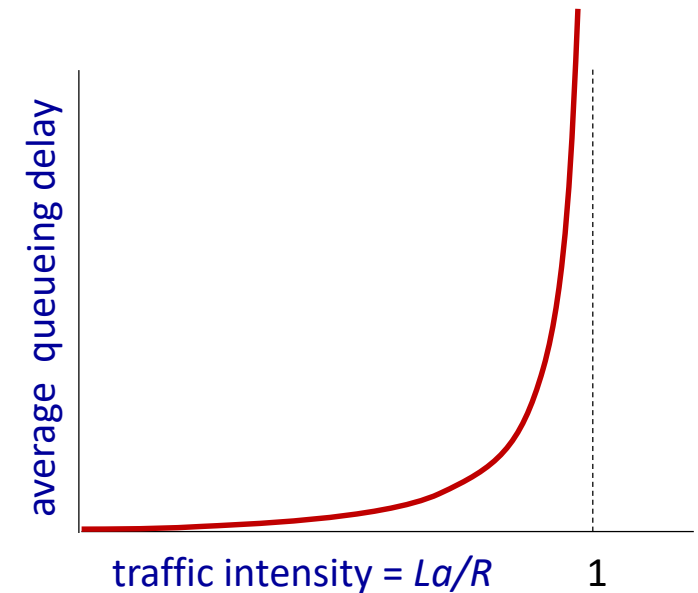
اگر بافر پر باشه پکت لاس اتفاق می افتد این پکت لاس رو TCP روی مکانیزیم reliable data transfer تشخیص میده و مجددا این پکت رو ارسال میکنه ولی به هر حال این وضعیت مطلوبی نیست برای اینکه این ارسال مجدد یک زمان transmtion از مبدا تا مقصد مجدد رو طول می کشه تا ارسال بشه و این خودش تاخیر زیادی رو ایجاد میکنه و در عین حال ظرفیت لینک های توی شبکه رو اشغال میکنه

Packet queueing delay (revisited)

- a : average packet arrival rate
- L : packet length (bits)
- R : link bandwidth (bit transmission rate)

$$\frac{L \cdot a}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}} \quad \text{“traffic intensity”}$$

- $La/R \sim 0$: avg. queueing delay small
- $La/R \rightarrow 1$: avg. queueing delay large
- $La/R > 1$: more “work” arriving is more than can be serviced - average delay infinite!



Principles of congestion control

Congestion:

- informally: “too many sources sending too much data too fast for *network* to handle”
- manifestations:
 - long delays (queueing in router buffers)
 - packet loss (buffer overflow at routers)
- different from flow control!
- a top-10 problem!



congestion control:

too many senders,
sending too fast



flow control: one sender
too fast for one receiver

- Congestion چی هست؟

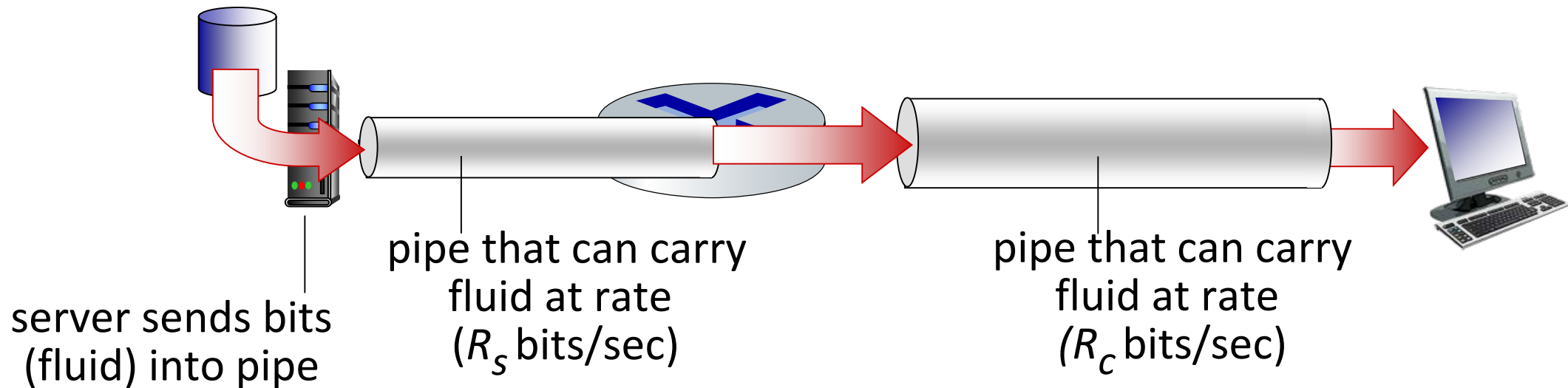
Congestion وقتی اتفاق می افتد که تعداد زیادی از منابع توی شبکه حجم زیادی از اطلاعات رو همزمان با سرعت بالا توی شبکه می فرستن و در شرایطی ممکنه این حجم اطلاعات در یک زمان بیشتر از حجم اطلاعاتی باشه که شبکه می تونه این رو عبور بده پس در این صورت در نودهای بین راه صف تشکیل میشه و این صف طولانی میشه و حتی بافر پر میشه و بسته های جدید ممکنه drop بشن و پکت لاس اتفاق می افتد و بسته هایی هم که توی صف قرار می گیرن با تاخیر روبه رو میشن چون هر بسته ای که توی صف قرار میگیره تا بیاد به سر صف برسه و ارسال بشه به اندازه زمان ارسال تمام بسته های جلوی خودش باید منتظر بمونه و این تاخیر می تونه خیلی زیاد باشه

ما این حالت رو می گیم Congestion یا ازدحام یا شلوغی

و این یکی از 10 مسئله اول شبکه بوده و مدت هایی طولانی روش کارش شده و هنوز هم روش داره کار میشه

Throughput

- *throughput*: rate (bits/time unit) at which bits are being sent from sender to receiver
 - *instantaneous*: rate at given point in time
 - *average*: rate over longer period of time

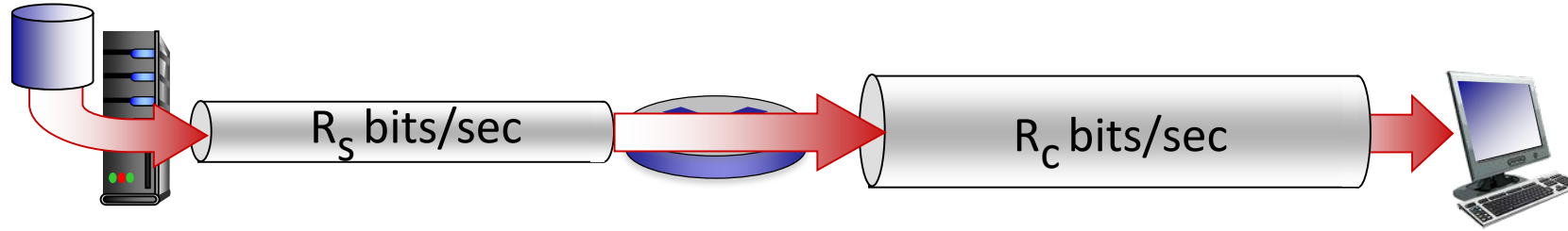


throughput: ریت انتقال اطلاعات بین مبدا و مقصد است معمولاً برحسب **bits/sec** است و هم می‌تونیم ریت لحظه‌ای رو در نظر بگیریم یعنی در یک لحظه تعداد بیت در ثانیه که از یک مقطعی از شبکه عبور میکنه و هم مقدار متوسط اون رو در طول یک بازه زمانی طولانی تر

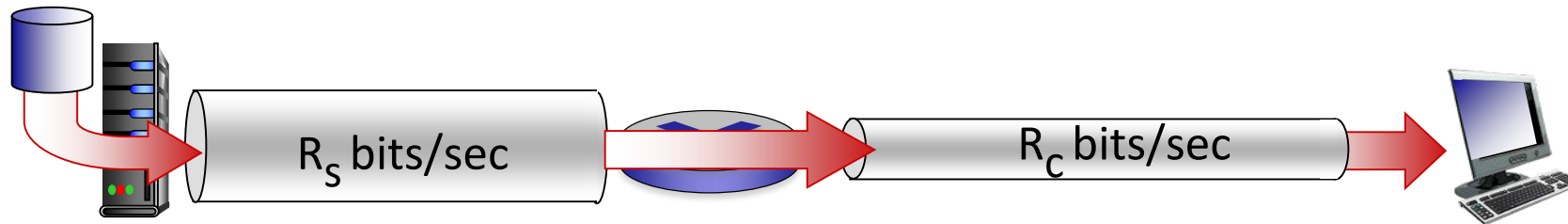
مثلاً توی این مثال یک نود بین مبدا و مقصد باشه و یک لینک از مبدا تا این نود و یک لینک هم از این نود تا مقصد و اگر فرض کنیم عرض باند لینک اولمون **Rs** بیت بر ثانیه باشه و عرض باند لینک دوم **Rc** بیت بر ثانیه در اینجا سوال مطرح میشه که **throughput** بین مبدا و مقصد چه مقداری خواهد بود

Throughput

$R_s < R_c$ What is average end-end throughput?



$R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput

در اینجا چه چیزی throughput انتقال اطلاعات بین مبدا و مقصد رو تعیین خواهد کرد؟ به عبارت دیگر چه عاملی محدودکننده این throughput خواهد بود؟

اون لینک هایی که تعداد بیت کمتری در هر ثانیه می تونه عبور بده به صورت bottleneck محدود می کنن throughput

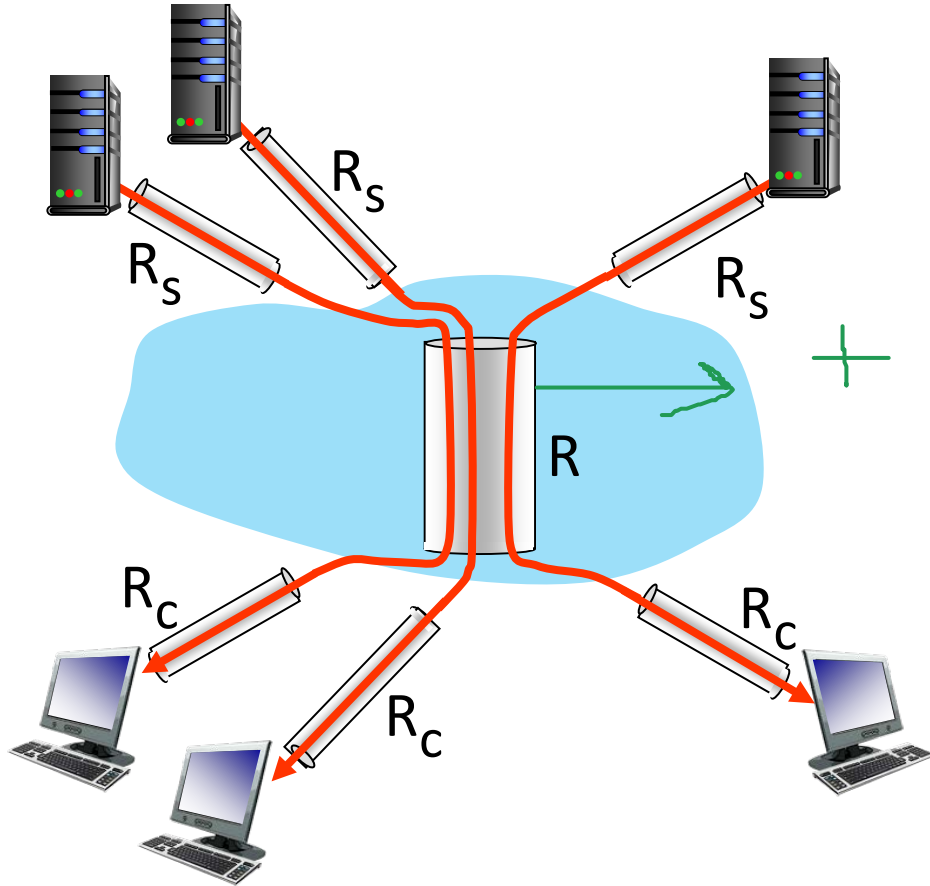
در این سناریو اگر ریت لینک اول R_s کمتر از R_c باشه این لینک R_s به عنوان bottleneck تعیین کننده خواهد بود

در سناریوی بعدی:

اگر لینک دوم R_c کمتر باشد این لینک به عنوان bottleneck تعیین کننده ظرفیت خواهد بود

پس لینک bottleneck است که $\text{throughput} \rightarrow \text{end to end}$ رو تعیین میکنه

Throughput: network scenario



10 connections (fairly) share
backbone bottleneck link R bits/sec

- per-connection end-end throughput:
 $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/

در یک سناریوی واقعی تر ما بین مبدا و مقصد لینک های متعددی رو خواهیم داشت
اگر فرض کنیم این + یکی از لینک های مسیر است در سطح شبکه و فرض کنیم این لینک
bottleneck است ینی بین همه لینک هایی که در سطح شبکه داریم این + کمترین نرخ رو داره
برای این کانکشن

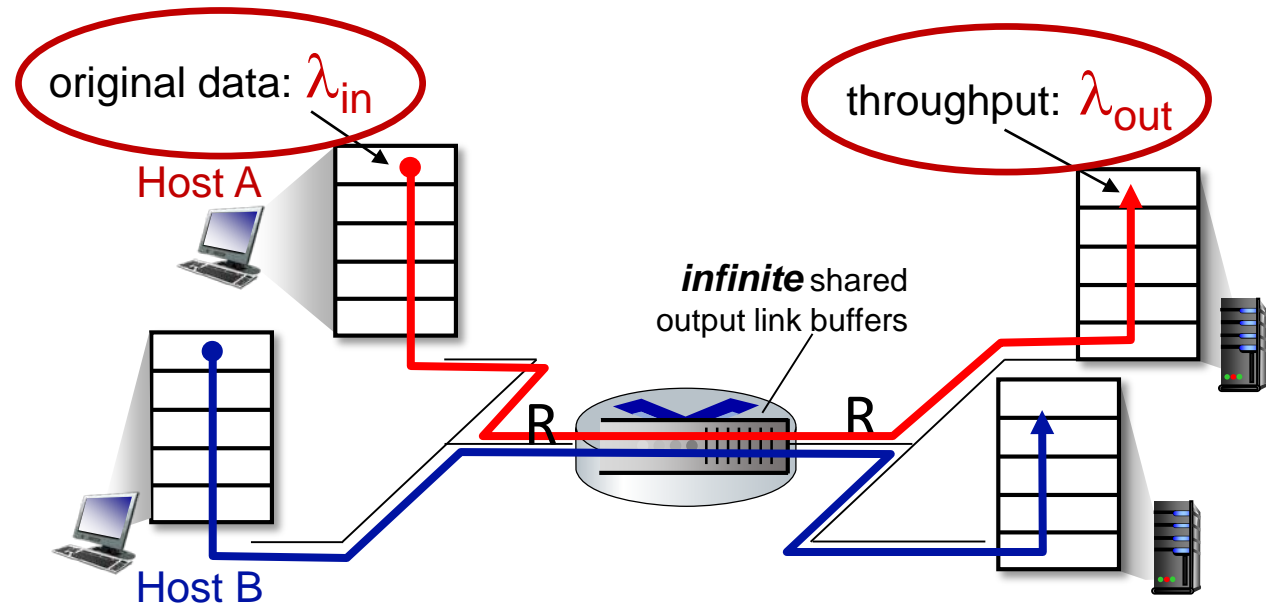
چرا میگیریم برای این کانکشن؟ چون لینک های داخل شبکه بین کانکشن های مختلف از مبدهای
مختلف به مقصدهای مختلف شیر میشن و در یک لحظه ممکنه کانکشن های مختلفی از طریق این
لینک برقرار باشه --> حالا اگر فرض کنیم حالت عادلانه داریم ظرفیت این لینک + بین کانکشن
های همزمان تقسیم میشه مثلا اگر 10 تا کانکشن روی این لینک برقرار باشه به هر کدوم از اونها
1/10 ظرفیت لینک می رسه که اگر ظرفیت لینک r باشه میشه $r/10$

پس برای یک کانکشن مشخص ما لینک access مبدا رو داریم و لینک bottleneck شبکه رو
داریم و لینک access مقصد رو ینی R_s , $r/10$, R_c حالا کدوم از این ها تعیین کننده
throughput خواهد بود؟ هر کدوم که bottleneck باشه مثلا اگر $r/10$ کمتر باشه این هست که
محدود کننده خواهد شد ولی البته معمولا لینک های access هستن که محدود کننده هستن و
معمولا لینک های داخل شبکه و به خصوص core شبکه ظرفیت های بالاتری دارن

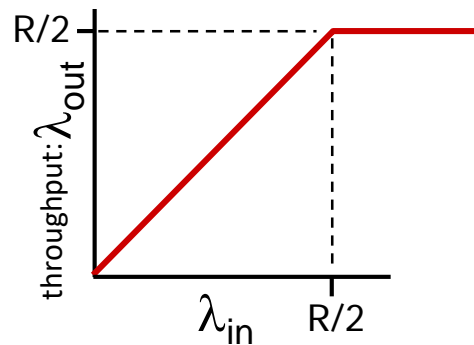
Causes/costs of congestion: scenario 1

Simplest scenario:

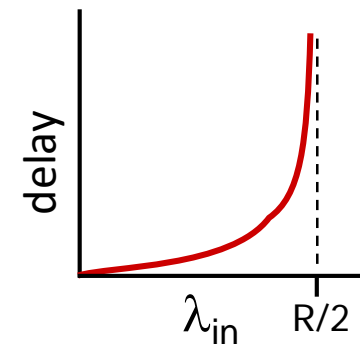
- one router, infinite buffers
- input, output link capacity: R
- two flows
- no retransmissions needed



Q: What happens as arrival rate λ_{in} approaches $R/2$?



maximum per-connection throughput: $R/2$



large delays as arrival rate λ_{in} approaches capacity

این نمودار رفتار این صف رو نشون میده

رفتار و تاثیرات Congestion:

فرض میکنیم که دوتا هاست مبدا داریم و دوتا هاست مقصد که بین هر جفتی از اونها یک ترافیکی در جریان است و فرض می کنیم یک نود بینشون وجود داره و این ترافیک از لینک نود مبدا تا نود میانی و از نود میانی تا نود مقصد عبور میکنه
در سناریوی اول فرض می کنیم که بافر نودمون محدودیت نداره و ظرفیت اون بی نهایت می تونه فرض بشه

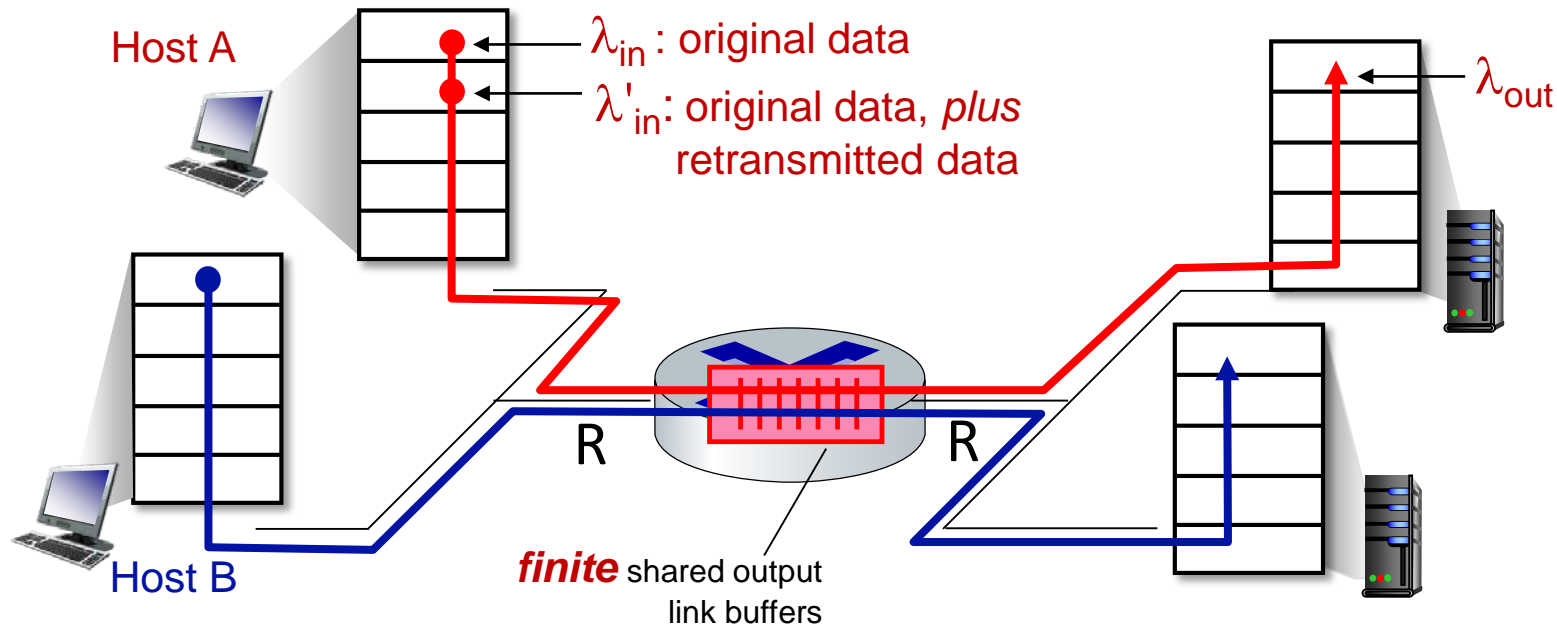
هاست A ترافیکی با ریت لاندا λ به شبکه می رسه و ترافیکی که مقصدمون از شبکه هم می گیره با لاندا λ نشون میدیم

اگر ریت لینک خروجی مشترک روتر برای این دوتا ترافیک R بیت بر ثانیه باشه در حالت ایده ال ما می تونیم تصور بکنیم که این ظرفیت بین این دوتا کانکشن + به طور مساوی تقسیم میشه و هر کدوم از اون ها نصف این ظرفیت رو به خودشون اختصاص میدن پس سهم کانکشن اولمون از ظرفیت این لینک میشه $R/2$ بیت بر ثانیه

فرض می کنیم هاست A شروع می کنه به ارسال اطلاعات با ریت پایین و به تدریج اینو افزایش میده ینی لاندا λ رو افزایش میده --> با این شرایطی که گفتیم همه ترافیکی که از این مبدا به نود میانی می رسه به خروجی ارسال میشه در نتیجه لاندا λ برابر با لاندا λ میشه تا زمانی که لاندا λ برسه به $R/2$ ظرفیت سهم کانکشن اولمون است و اگر لاندا λ از $R/2$ بیشتر بشه لاندا λ outمون روی $R/2$ محدود خواهد شد چون ظرفیت بیشتری رو لینک خروجی نداره برای ارسال اطلاعات بیشتر در نتیجه برای بسته های اضافی تر چه اتفاقی می افته؟ بسته های اضافی تر اینجا توی بافر قرار میگیره و صف تشکیل میشه و اگر ارسال اطلاعات با ریت بزرگتر از $R/2$ از مبدا ادامه پیدا کنه چون به هر حال ریت خروجی بیشتر از $R/2$ نمی تونه باشه این صف شروع میکنه با اضافه تر شدن به این ترتیب لاندا λ مون هرچی افزایش پیدا بکنه صفمون طولانی تر خواهد شد و در نتیجه بسته ای که ته صف قرار میگیره تا بیاد به سر صف برسه تاخیر خیلی بیشتری رو تحمل خواهد کرد

Causes/costs of congestion: scenario 2

- one router, *finite* buffers
- sender retransmits lost, timed-out packet
 - application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
 - transport-layer input includes *retransmissions* : $\lambda'_{in} \geq \lambda_{in}$



سناریوی دوم:

بافر مون محدود است

فرض میکنیم مکانیزیم retransmission برقرار است و بسته هایی که Acknowledge نمیشن و timeout میشه مبدامون اینارو مجددا می فرسته

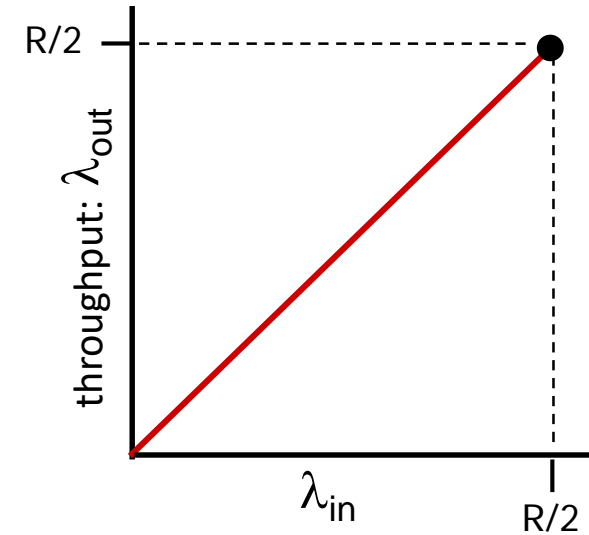
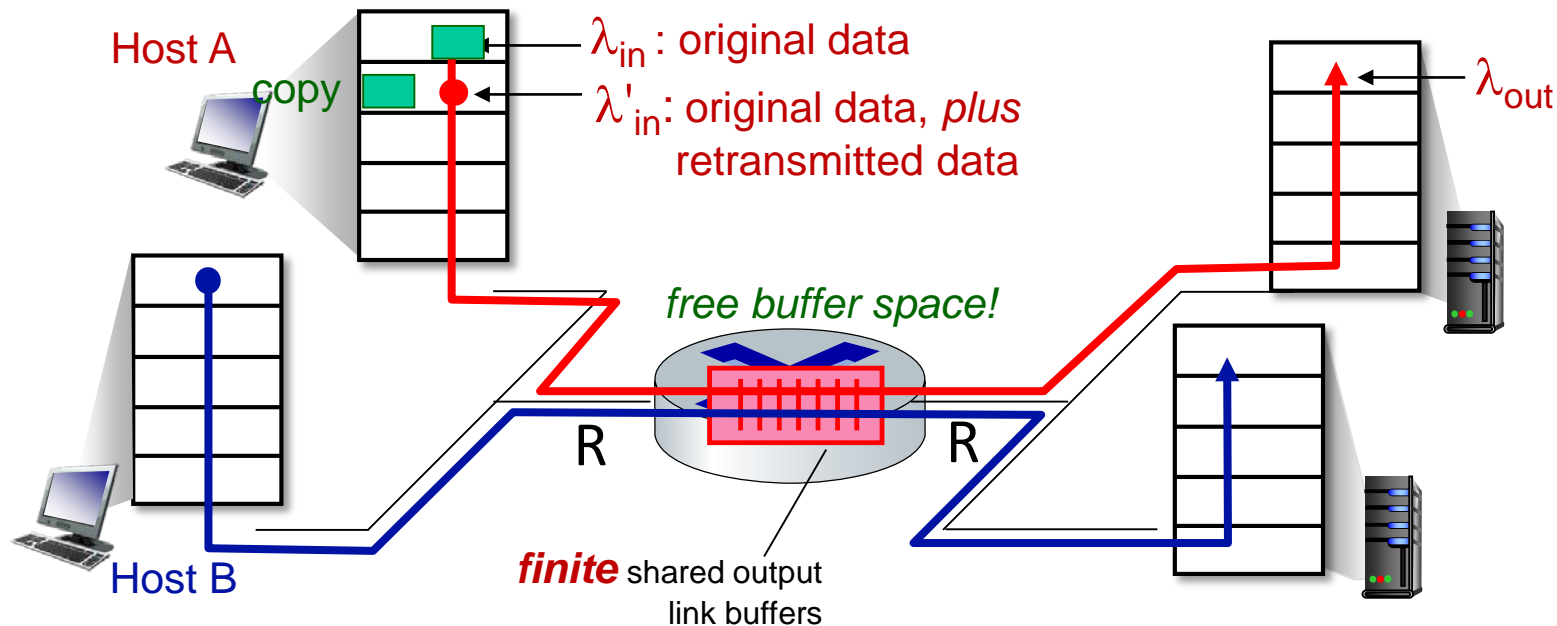
در این حالت اگر اپلیکیشن ما دیتایی با ریت لاندا in به لایه transport میده که ارسال بشه از طریق شبکه در نهایت اپلیکیشن ما در مقصد ینی لاندا out ترافیکی که دریافت میکنه با اون برابر خواهد بود ینی لاندا in و لاندا out در سطح لایه اپلیکیشن برابر خواهد بود و اگر لاندا پرین in ترافیکی باشه که لایه transport توی شبکه می فرسته لاندا پرین in از لاندا in بیشتر خواهد بود چرا؟ بخاطر اینکه لاندا پرین in بسته هایی رو که ارسال مجدد می شن رو هم شامل میشه ینی عرض باند بین مبدامون تا نود میانی ترافیکی که از این عبور میکنه بخشیش بسته هایی هستن که مجددا دارن ارسال میشن که اون ها قبلا جز لاندا in حساب شدن پس لاندا پرین in ما بیشتر از لاندا in خواهد بود

ابتدا یک حالت ایده ال رو در نظر میگیریم که اون این هست که فرض میکنیم مبدا از وضعیت بافر نود مطلع است و تا حدی بسته ها رو می فرسته که بافر جا داشته باشه پس بسته های زیادی ممکن بیان و صف تشکیل بشه و تاخیر صف می تونه اتفاق بیوفته ولی تا حدی نخواهد بود که صف پر بشه و بسته های drop بشن پس یک بسته ارسال میشه و یک کپییش در لایه transport نگهداری میشه و احتمالا میاد توی صف قرار میگیره و وقتی نوبتش شد به مقصد ارسال میشه

Causes/costs of congestion: scenario 2

Idealization: perfect knowledge

- sender sends only when router buffers available



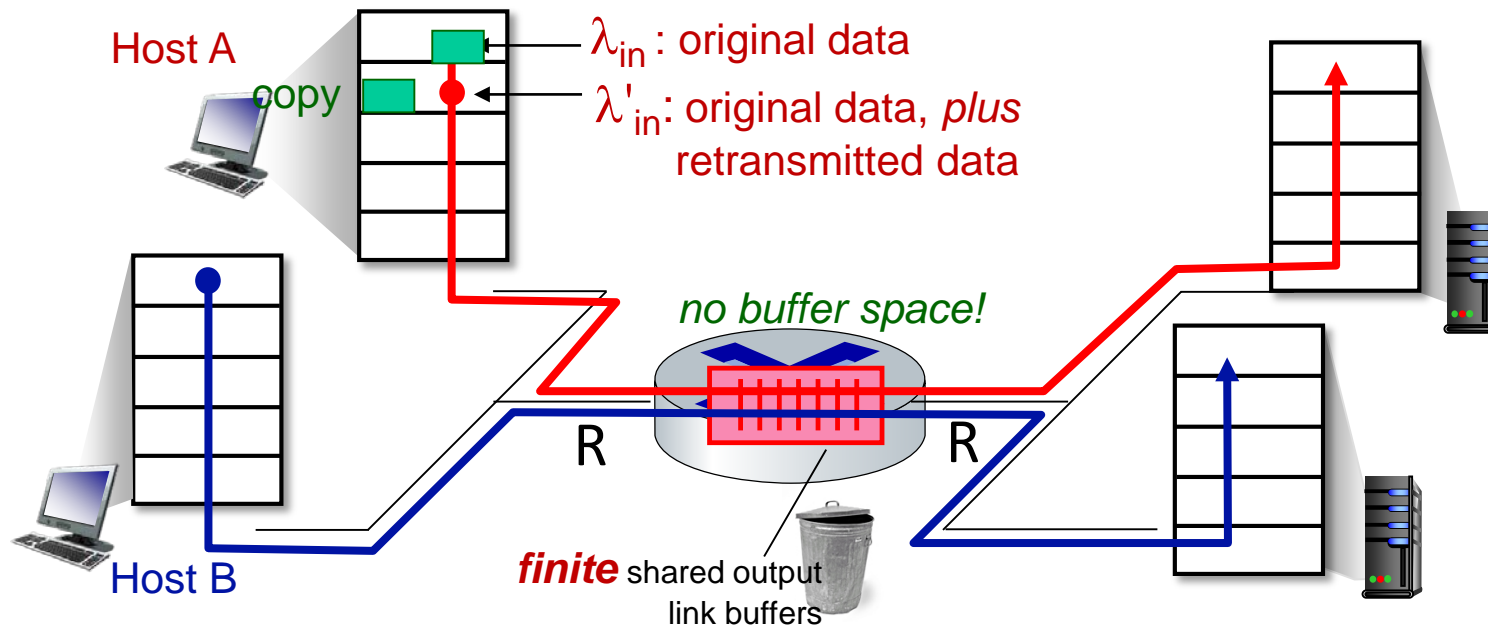
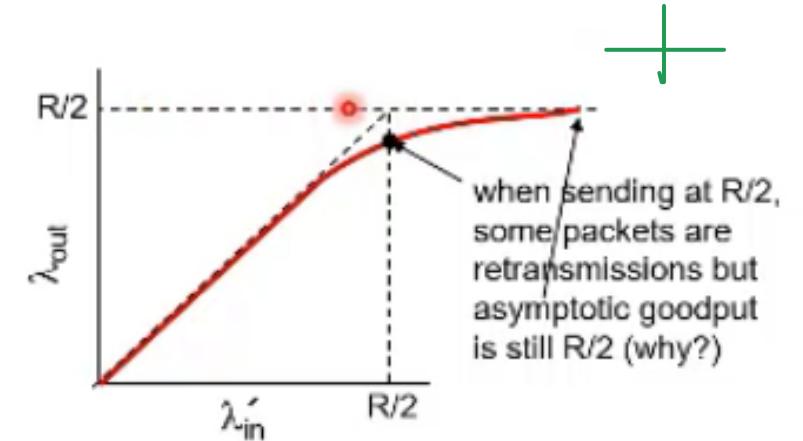
-

در این حالت تا زمانی که متوسط ترافیک ورودی از $R/2$ کمتر است ترافیکی که به مقصد می رسه هم با اون برابر خواهد بود تا $R/2$

Causes/costs of congestion: scenario 2

Idealization: *some* perfect knowledge

- packets can be lost (dropped at router) due to full buffers
- sender knows when packet has been dropped: only resends if packet *known* to be lost



حالت بعدی: (فکر کنم این میشه)

حالا فرض میکنیم مبدا می تونه بسته های بیشتری بفرسته به طوری که می تونه loss اتفاق بیوفته در اینصورت بسته ها retransmit باید بشن ولی بازم یک فرض ایده الی می کنیم و اون این است که مبدا فقط بسته های رو retransmit می کنه که می دونه که گم شدن پس یک بسته ارسال میشه و کپیش توی نگهداری میشه و این ممکنه توی صف جا نباشه و drop بشه در این صورت کپی مجددا ارسال میشه تا به مقصد برسه

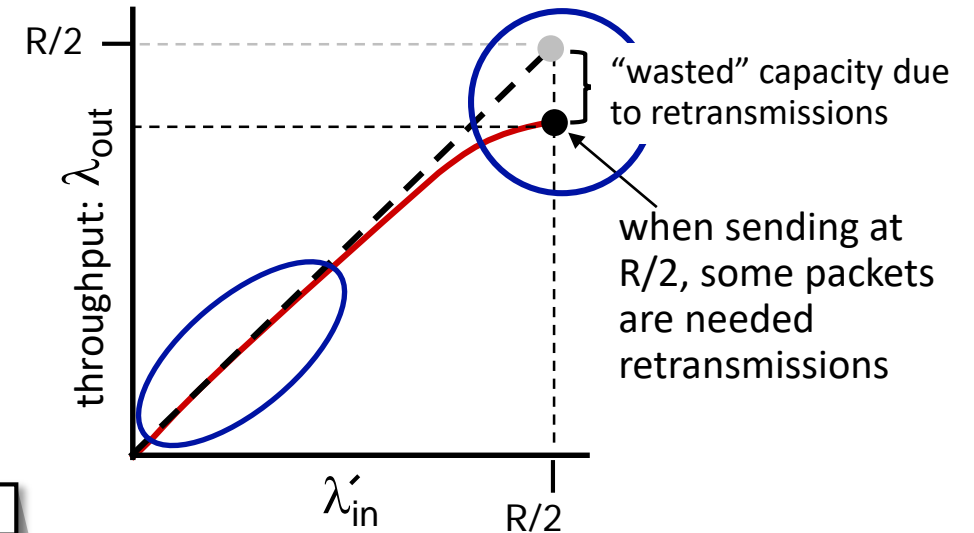
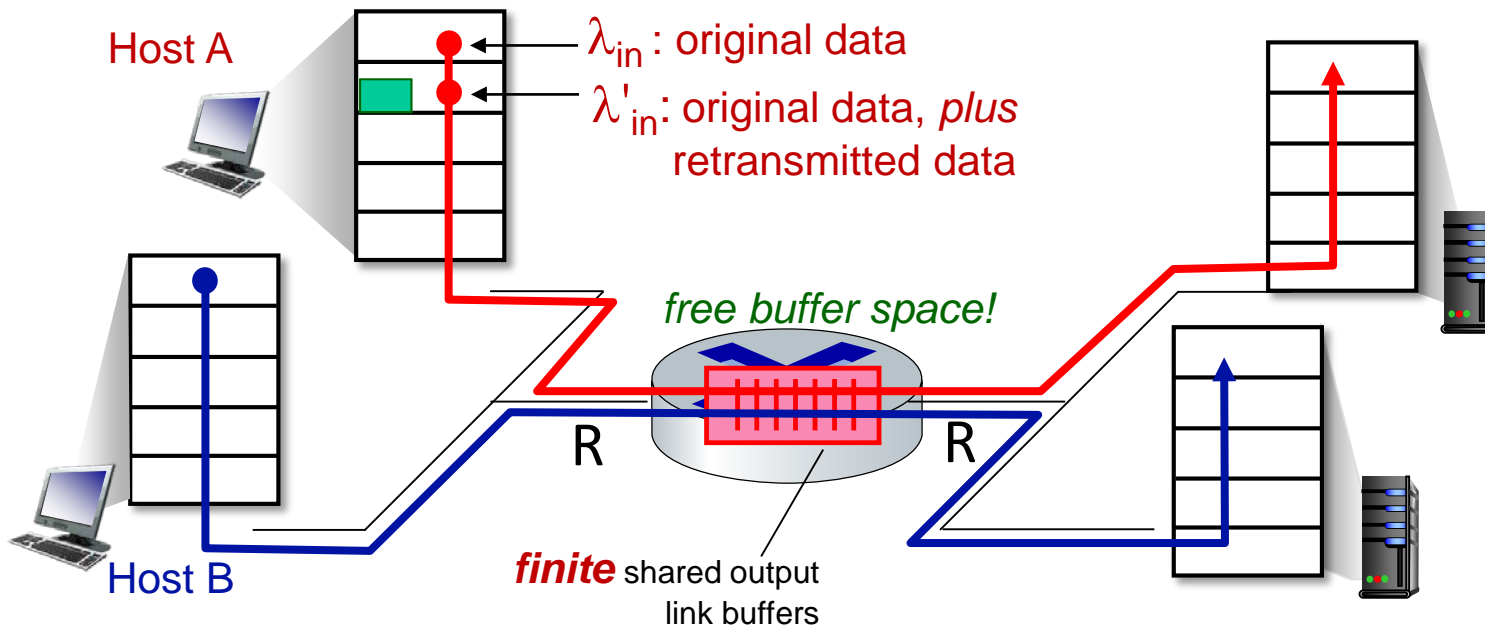
در این حالت وضعیت ترافیک به چه صورت خواهد بود؟

لاندا پرین in ترافیک واقعی است که از مبدا به لینک شبکه ارسال میشه --> این هرچی افزایش پیدا بکنه لاندا out افزایش پیدا می کنه تا به جایی نزدیک $R/2$ که در اینجا حالتی است که صف تشکیل میشه و طولانی میشه و در نتیجه loss میتونه اتفاق بیوفته و retransmission مطابق اون و این retransmission ها باعث میشن لاندا out ما از لاندا پرین in کمتر باشه همین طور اگر از ظرفیت لینک بیشتر استفاده بکنیم در نهایت ممکنه ترافیکی که مقصد دریافت میکنه به $R/2$ برسه و در این صورت این حجم اضافیه که از $R/2$ تا اون ترافیک $x + R/2$ این x اش مال retransmission ها خواهد ینی در نهایت می تونیم تا جایی پیش بریم که مقصد $R/2$ مون ترافیک رو دریافت بکنه --> شکل + که خودم اضافه اش کردم
ادامش صفحه بعدی...

Causes/costs of congestion: scenario 2

Idealization: *some* perfect knowledge

- packets can be lost (dropped at router) due to full buffers
- sender knows when packet has been dropped: only resends if packet *known* to be lost



ادامه...

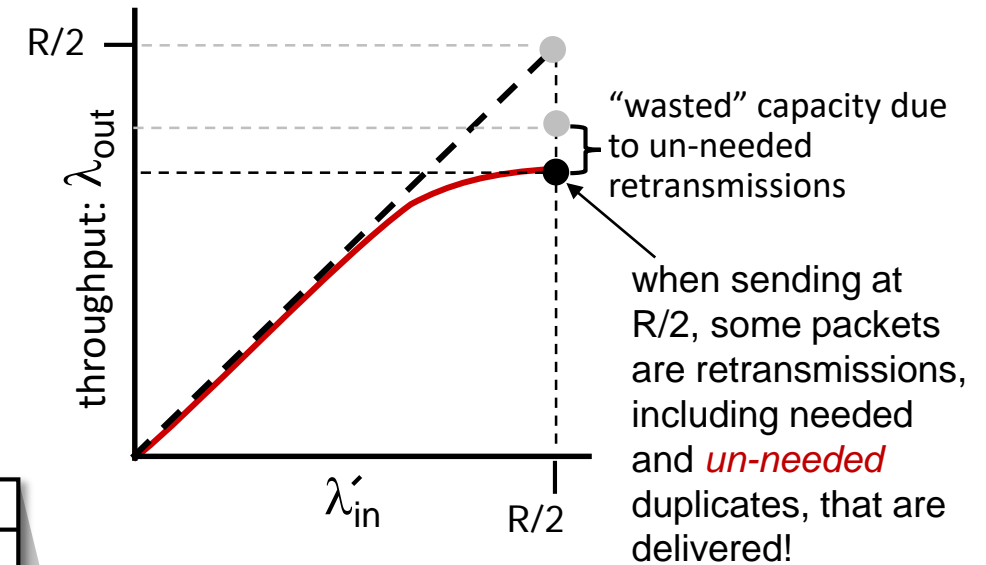
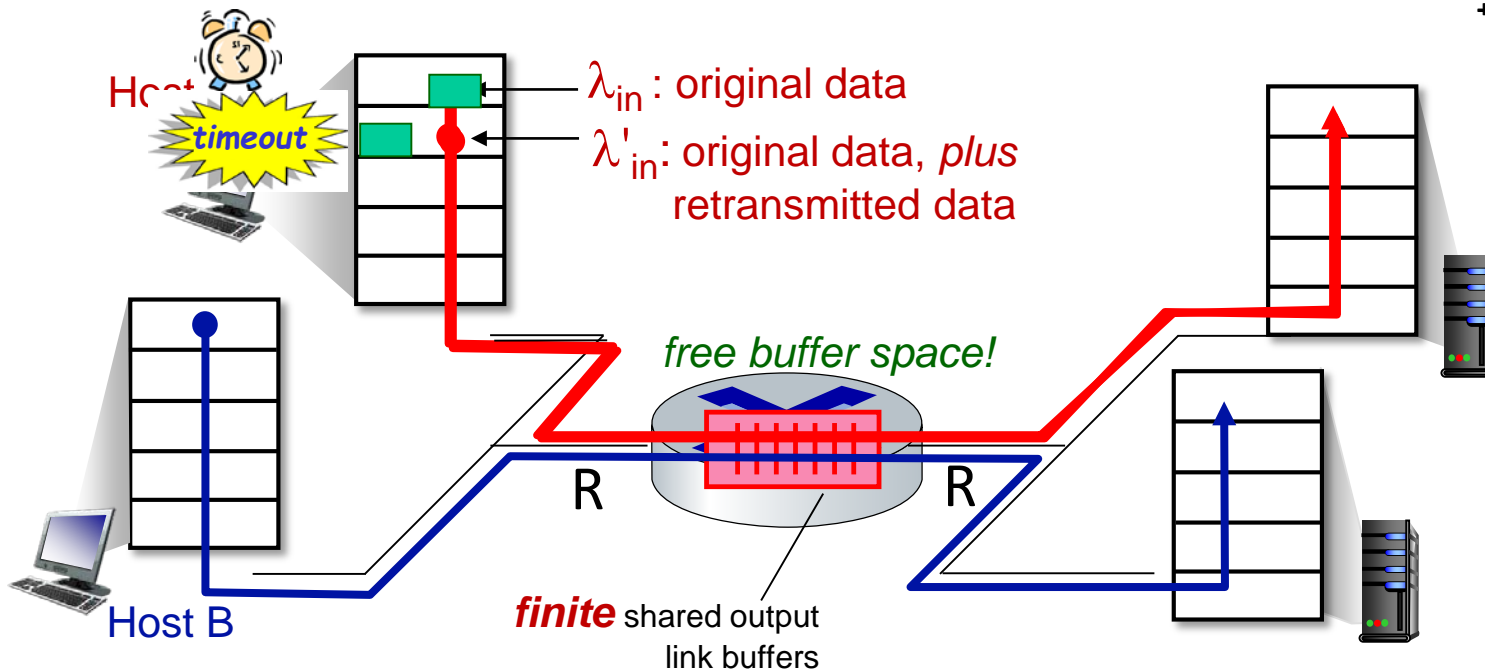
زمانی که پکت لاس اتفاق می افتد باید دوباره پکت هایی که لاس شدن ارسال بشن و retransmission یا ارسال مجدد اتفاق بیوفته و این retransmission بخشی از اون $R/2$ رو اشغال میکنه ینی در شرایطی که تا $R/2$ ارسال میکنیم <-- ینی لاند پرین in تا $R/2$ باشه اون چیزی که دریافت میکنیم در مقصد کمتر از $R/2$ خواهد بود چون بخشی از $R/2$ که داریم پر میکنیم با retransmission ها داریم پر میکنیم ینی retransmission پکت هایی که قبلا نرسیدن ینی $R/2$ داریم می فرستیم ولی کمتر از $R/2$ داره دریافت میشه پس یه مقداری از عرض باند رو داریم از دست می دیم پس و گزردهی کمتر از اون $R/2$ میشه که قرار بود بشه <-- عوارض congestion است

اینجا سناریو ایده ال است ینی فرض شده فرستنده از وضعیت بافر اطلاع داره و می فهمه چه پکت هایی لاس شدن و اونارو retransmission میکنه ولی هنوز تایم اوت نداریم

Causes/costs of congestion: scenario 2

Realistic scenario: *un-needed duplicates*

- packets can be lost, dropped at router due to full buffers – requiring retransmissions
- but sender times can time out prematurely, sending *two* copies, *both* of which are delivered



ولی اگر توی سناریوی قبلی تایم اوت هم در نظر بگیریم وضعیت اینجا بدتر هم میشه -->

اگر تایم اوت مقدارش کمتر باشه این باعث میشه وقتی تاخیر صف از یه حدی بیشتر شد منجر به

تایم اوت بشه قبل از اینکه پکت لاس اتفاق بیوفته ینی فقط تاخیر زیاد شده و بافر پر نشده و پکت

لاس هم اتفاق نیوفتاده پس به مقصد خواهد رسید این پکت ها در حال کلی ولی چون تایم اوت کم

ست شده پکت دوباره ارسال میشه و پکتی که قبلا به مسیر رسیده دوباره ارسال میشه و توی مقصد

چون **#seq** تکراری مقصد اینو دور می ریزه --> توی این حالت ظرفیت لینک خروجی هم اشغال

میشه چون تا مقصد می رن و اونجا **drop** میشن چون تکراری است پس اگر لاندا پرین **in** تا **R/2**

برسه لاندا **out** از اون حد قبلی هم کمتر میشه چون یه مقداری هم بسته هایی هستن که می رسن

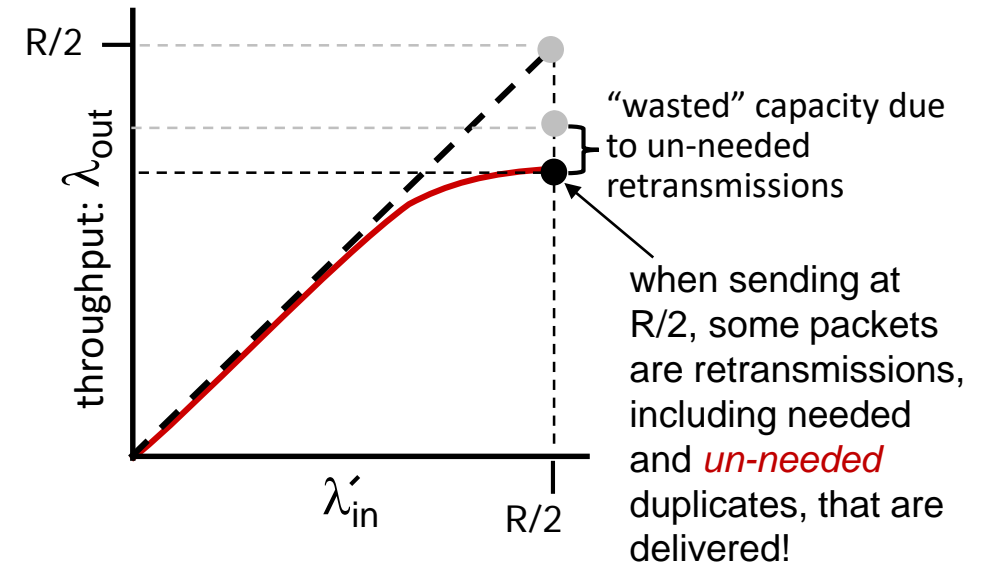
ولی تکراری است و از حالت قبلی هم این لاندا **out** کمتر میشه پس عرض باند تلف شده به این

ترتیب خیلی بیشتر میشه

Causes/costs of congestion: scenario 2

Realistic scenario: *un-needed duplicates*

- packets can be lost, dropped at router due to full buffers – requiring retransmissions
- but sender times can time out prematurely, sending *two* copies, *both* of which are delivered



“costs” of congestion:

- more work (retransmission) for given receiver throughput
- unneeded retransmissions: link carries multiple copies of a packet
 - decreasing maximum achievable throughput

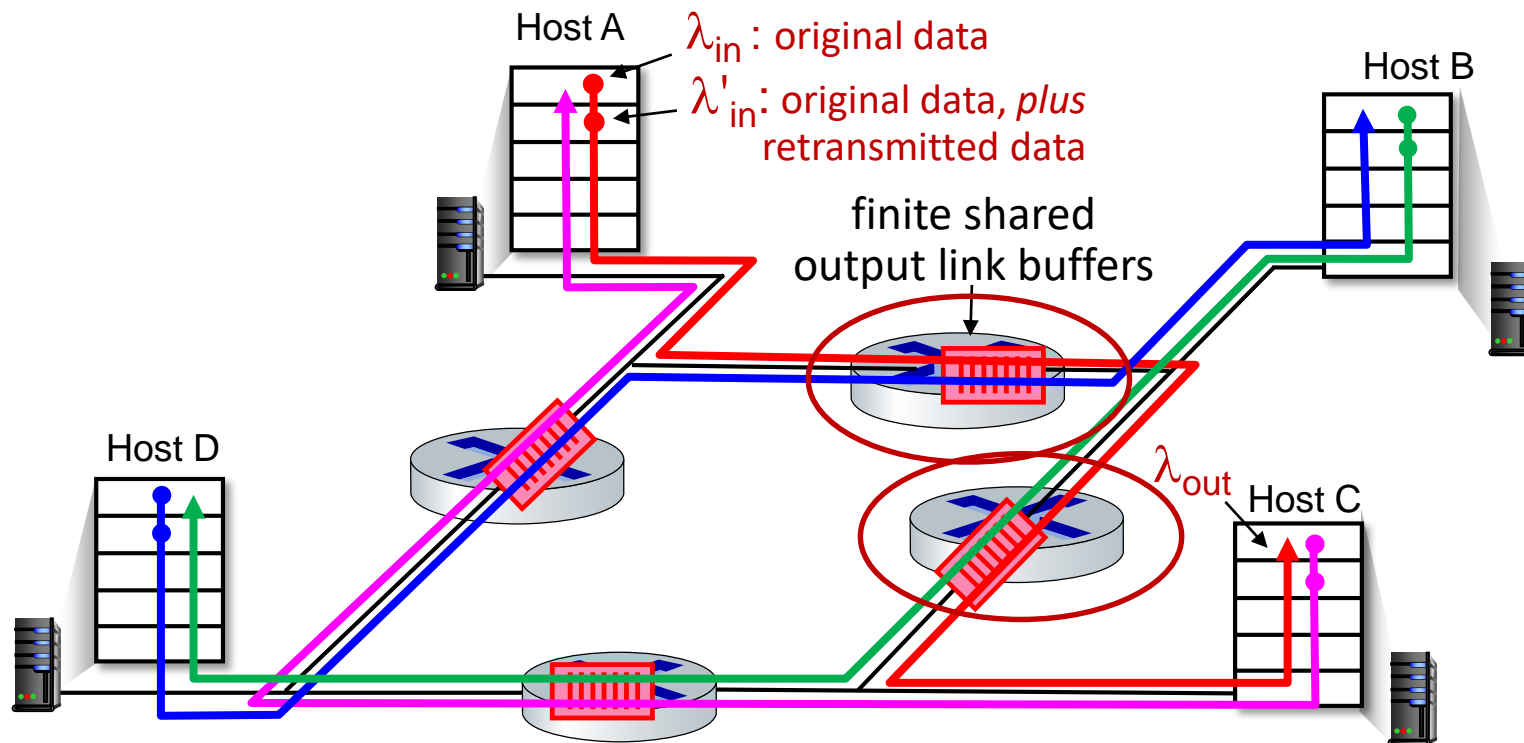
پس در شرایط واقعی retransmission بخاطر پکت های لاس شده و retransmission
خاطر پکت هایی که تاخیرشون بیشتر از تایم اوت شده و عملا به مقداری از عرض باند مفید رو از
دست میدیم <-- یکی از نتایج congestion است

Causes/costs of congestion: scenario 3

- *four* senders
- *multi-hop* paths
- timeout/retransmit

Q: what happens as λ_{in} and λ'_{in} increase ?

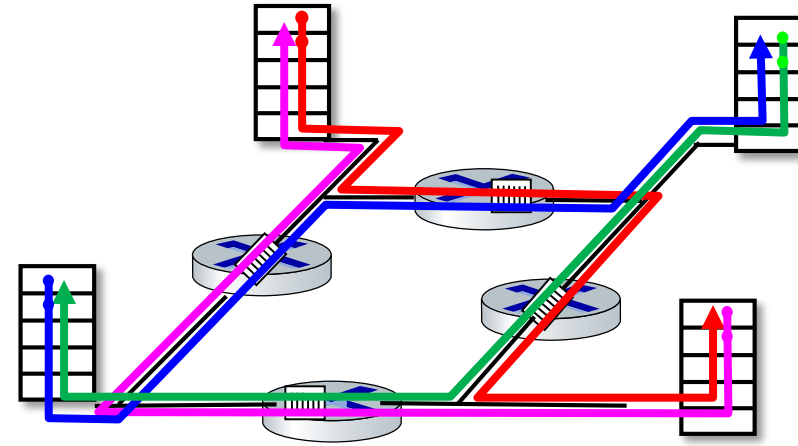
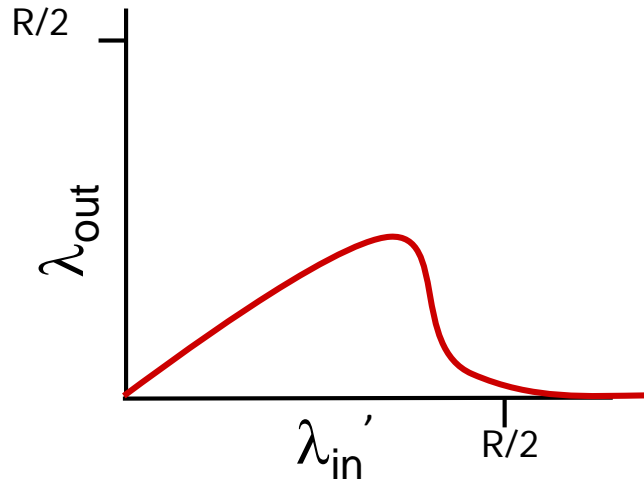
A: as red λ'_{in} increases, all arriving blue pkts at upper queue are dropped, blue throughput $\rightarrow 0$



سناریو: یک شبکه به این صورت است

اینجا ترافیک قرمز ما از A تا C از روتر بالایی عبور پیدا میکنه و ترافیک ابی هم از D تا B از روتر بالایی عبور پیدا میکنه --> دوتا ترافیکی که لینک این روتر بالایی رو شیر می کنن و ارزش عبور می کنن دوباره ابی و قرمز است منتها شرایطشون مثل هم نیست
این شرایطی که مثل هم نیست منجر به وضعیت جدیدی میشه و اون حالتی که گفتیم اینا هر کدوم می تونن نصف عرض باند رو استفاده بکنن این شرایط بدتر از این میشه
صفحه بعدی...

Causes/costs of congestion: scenario 3



another “cost” of congestion:

- when packet dropped, any upstream transmission capacity and buffering used for that packet was wasted!

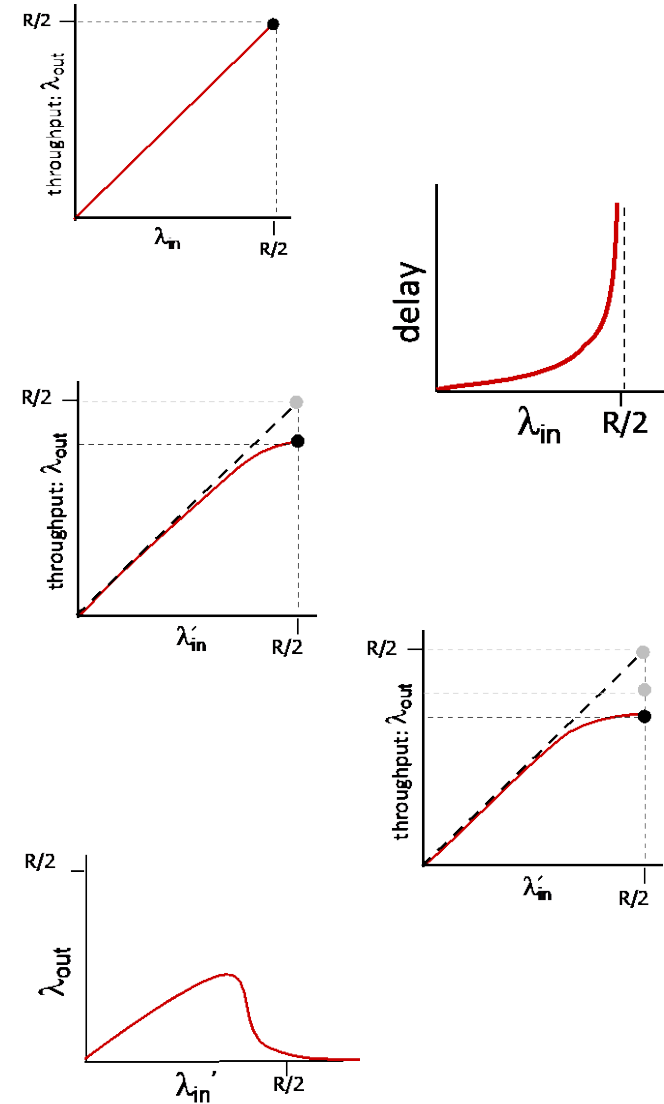
ادامه..

به این صورت که ترافیک قرمز ما اگر افزایش پیدا بکند اینجا باعث میشه که صف پر بشه و صف که پرشد باعث میشه drop اتفاق بیوفته و این Drop برای هر دو ترافیک رخ میده چون صف مشترک است اینجا ترافیک ابی چون از یک لینک قبل تر و با تاخیر بیشتری داره به این روتر مشترک می رسه این drop هایی ک اتفاق می افته در پروسه tcp منجر به این خواهد شد که هاست D ارسالش رو کمتر بکند پس ترافیک قرمز سهم بیشتری از این لینک رو به دست میاره و سهم ابی رو هم کمتر میکنه و این می تونه منجر به این بشه که ترافیک ابی عرض باندش به صفر میل بکند --< منحنی مال ترافیک ابی است --< ترافیک ابی لاندا پرین in اش افزایش پیدا میکنه تا یه جایی و تا نزدیک $R/2$ که داره میشه و congestion داره اتفاق می افته لاندا out اش شروع میکنه به کم شدن و این می تونه به صفر میل بکند پس گذردهی ترافیک ابی به دلیل congestion در اون روتر مشترک به صفر حتی میل می کنه

بخشی از ظرفیت شبکه رو به این ترتیب از دست می دیم بخاطر بسته هایی که بعدا در نودهای بالایی drop خواهند شد مثل همین D که توی روتر مشترک drop میشه بسته هاش ولی یه سری لینک قبلش بوده که داشته استفاده می کرده --< پس عرض باند مفید شبکه به این ترتیب کمتر است

Causes/costs of congestion: insights

- throughput can never exceed capacity
- delay increases as capacity approached
- loss/retransmission decreases effective throughput
- un-needed duplicates further decreases effective throughput
- upstream transmission capacity / buffering wasted for packets lost downstream



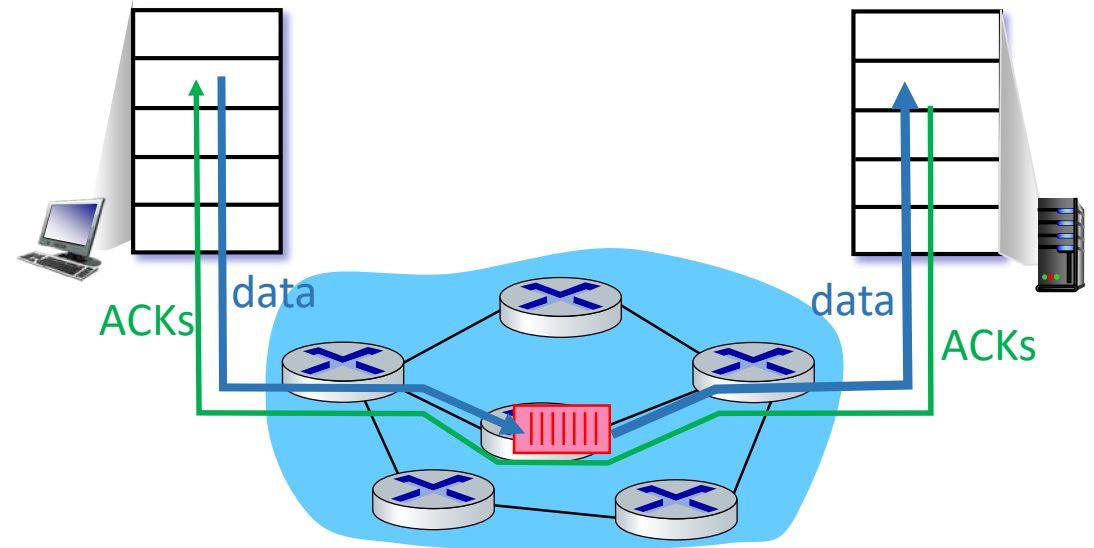
جمع بندی:

در شرایط واقعی گذردهی کمتر از گذردهی ظرفیت اسمی لینک ها خواهد بود در این منحنی ها اگر بخوایم اون نقطه ای رو پیدا بکنیم که واقعا $R/2$ ترافیک به مقصد می رسد یعنی لاندا out ما $R/2$ باشد کی اتفاق می افته؟ وقتی که لاندا in خیلی بیشتر از $R/2$ است چه قدر بیشتر؟ اونقدری که retransmission ها و transmission تکراری رو اگر هم حذف بکنیم تازه بعدش می رسیم به $R/2$ ولی این معنیش این خواهد بود اون ترافیک دیگری که از همین لینک عبور میکنه اون کمتر از $R/2$ بشه مثل سناریوی اخر که گفتیم -- پس throughput کل شبکمون کمتر از اون ظرفیت عرض باندی هست که می تونه استفاده بشه

Approaches towards congestion control

End-end congestion control:

- no explicit feedback from network
- congestion *inferred* from observed loss, delay
- approach taken by TCP



پس congestion می تونه اتفاق بیوفته و congestion اگر اتفاق بیوفته اثرات مستقیمش افزایش تاخیر- پکت لاس - کاهش throughput یک ترافیک بین مبدا و مقصد و ضایع شدن عرض باند مفید شبکمون است

برای مدیریت congestion ما دوتا رویکرد داریم:

یک بحث اجتناب از congestion است و یک بحث مدیریت congestion

اجتناب از congestion این است که ما نداریم وضعیت وخیم بشه ینی نداریم به اینجا بکشه که صف ها طولانی بشه و بافرها پر بشه و

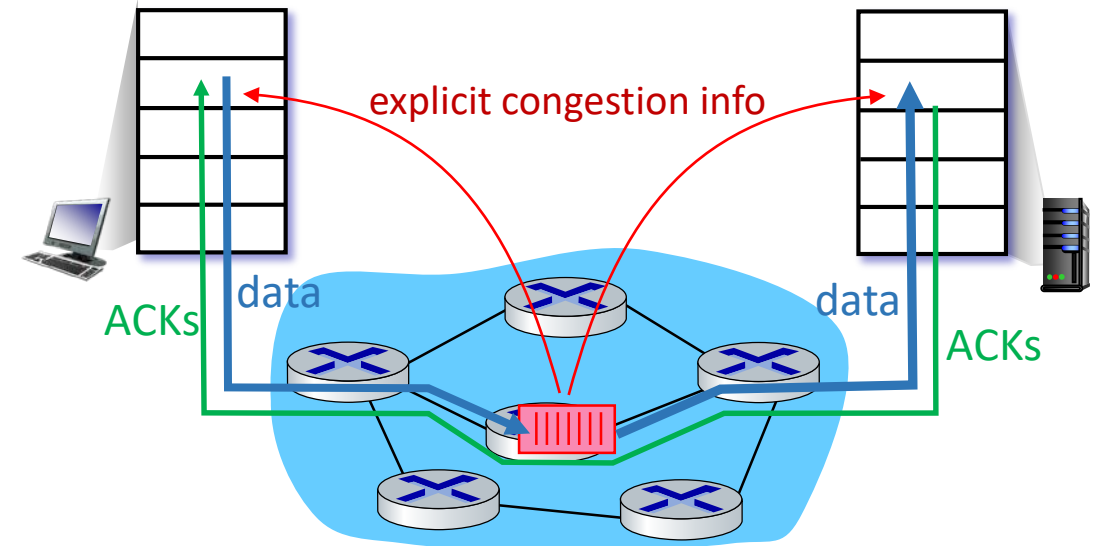
مدیریت congestion ینی این اتفاق افتاده حالا چی کار کنیم وضعیت به حالت مناسب برگرده دوتا دیدگاه برای هر دوی این ها وجود داره:
دیدگاه اول:

End-end congestion control: بین مبدا و مقصد سعی میشه که یه کاری اتفاق بیوفته تا این وضعیت کنترل بشه که این فراتر از خود شبکه است ینی congestion یک وضعیتی است که توی شبکه اتفاق می افته ولی اینجا ما بین مبدا و مقصد سعی میکنیم وضعیت رو مانیتور بکنیم تا متوجه بشیم congestion اتفاق افتاده سعی کنیم باهاش مقابله بکنیم
این مانیتورینگ بر چه اساسی می تونه نتیجه بگیره که congestion اتفاق افتاده؟ از اثرات و عوارض congestion --> اثراتش: مثلا تاخیر اضافه میشه و تا حدی که اضافه شد پکت لاس شروع میشه به اتفاق افتادن --> هم تاخیر و هم پکت لاس رو اگر با هم هماهنگ باشن می تونن تشخیص بدن پس ما اینجا اصلا کاری به خود روتر نداریم بلکه ترافیکی که عبور میکنه و به مقصد می رسه تاخیر این رو اگر بتونیم مانیتور بکنیم و دنبال بکنیم و هر موقع دیدیم تاخیرها داره زیاد میشه می فهمیم congestion داره اتفاق می افته یا هر موقع پکت لاس اتفاق افتاد

Approaches towards congestion control

Network-assisted congestion control:

- routers provide *direct* feedback to sending/receiving hosts with flows passing through congested router
- may indicate congestion level or explicitly set sending rate
- TCP ECN, ATM, DECbit protocols



اینجا ما یک بحثی رو پذیرفتیم و اون این است که اگر پکت لاس توی شبکه اتفاق افتاده این برای congestion است و congestion باعث پکت لاس شده پس اگر پکت لاس اتفاق بیوفته ما ترافیک ورودی رو کم میکنیم تا congestion کم بشه --> ایا این فرض درستی است؟ بله درسته در بیشتر شبکه ها به جز یه سری شبکه ها مثل شبکه های بی سیم که داخلشون drop بسته ها بخاطر نویز است و اونجا وقتی که پکت لاس میشه الزاما به معنای congestion است و می تونه بخاطر وضعیت کانال باشه

دیدگاه دوم:

Network-assisted congestion control: با تکیه بر خود شبکه و روترهای شبکه است
ینی روترهای شبکه هم درگیر میشن و کمک می کنن --> چجوری اتفاق می افته؟ در دیدگاه اول داریم به صورت غیر مستقیم میگیریم که توی شبکه congestion است
پس روتری که توش congestion اتفاق افتاده می تونه به طور مستقیم اینو بفهمه و روتری که اینو تشخیص داد روتر اینو اطلاع میده این اطلاع می تونه به صورت فیدبک باشه ینی بیاد به مبدا همون ترافیک خبر بده که این مستلزم یک مکانیزم مستقلی است که علاوه بر اون ارتباط tcp که دیتا داره توش منتقل میشه یک مکانیزم هایی طراحی شده که پروتکل جداگانه ای برای این لازم نداره ینی میاد در همون کانکشن و توسط همون بسته هایی که عبور پیدا میکنند این رو اطلاع رسانی میکنه پس در دیدگاه دوم روتر توی شبکه درگیر است و خودش اینو تشخیص میده و به مبدا یا از طریق مقصد اینو به مبدا منتقل میکنه و مبدا هم ترافیک کمتری توی شبکه می فرسته