بسمه تعالی

School of Computer Engineering

Advanced Computer Networks

Transport Layer and Congestion Control

Part 3

Seyed Hamed Rastegar

Fall 1401

Lecture roadmap

- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Connection-oriented transport: TCP
- Principles of congestion control
  - Motivating Discussion
  - Congestion Control and Resource Allocation
  - Queueing Disciplines
- Congestion control Techniques

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

Principles of congestion control

Flow Control vs Congestion Control

- **Flow control** involves keeping a fast sender from overrunning a slow receiver.

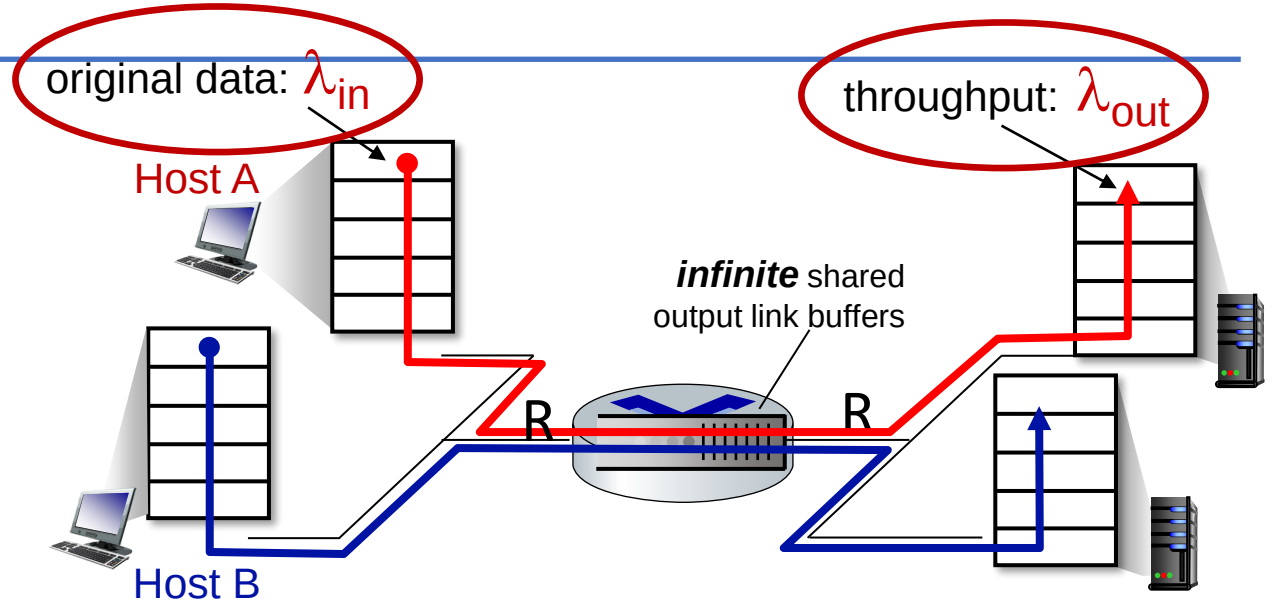
- **Congestion control**, by contrast, is intended to keep a set of senders from sending too much data into the network because of lack of resources at some point.

- These two concepts are often confused; however, they also share some mechanisms.

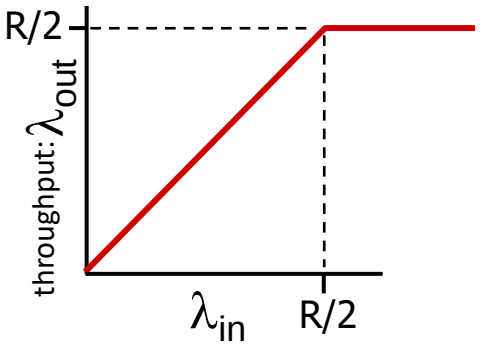
Causes/costs of congestion: scenario 1

Simplest scenario:

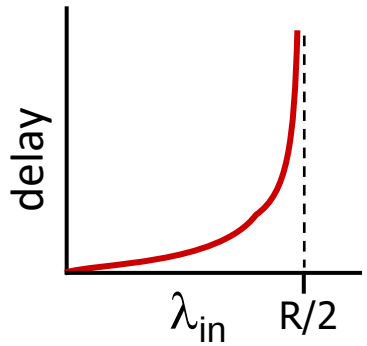
- one router, infinite buffers
- input, output link capacity: R
- two flows
- no retransmissions needed



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



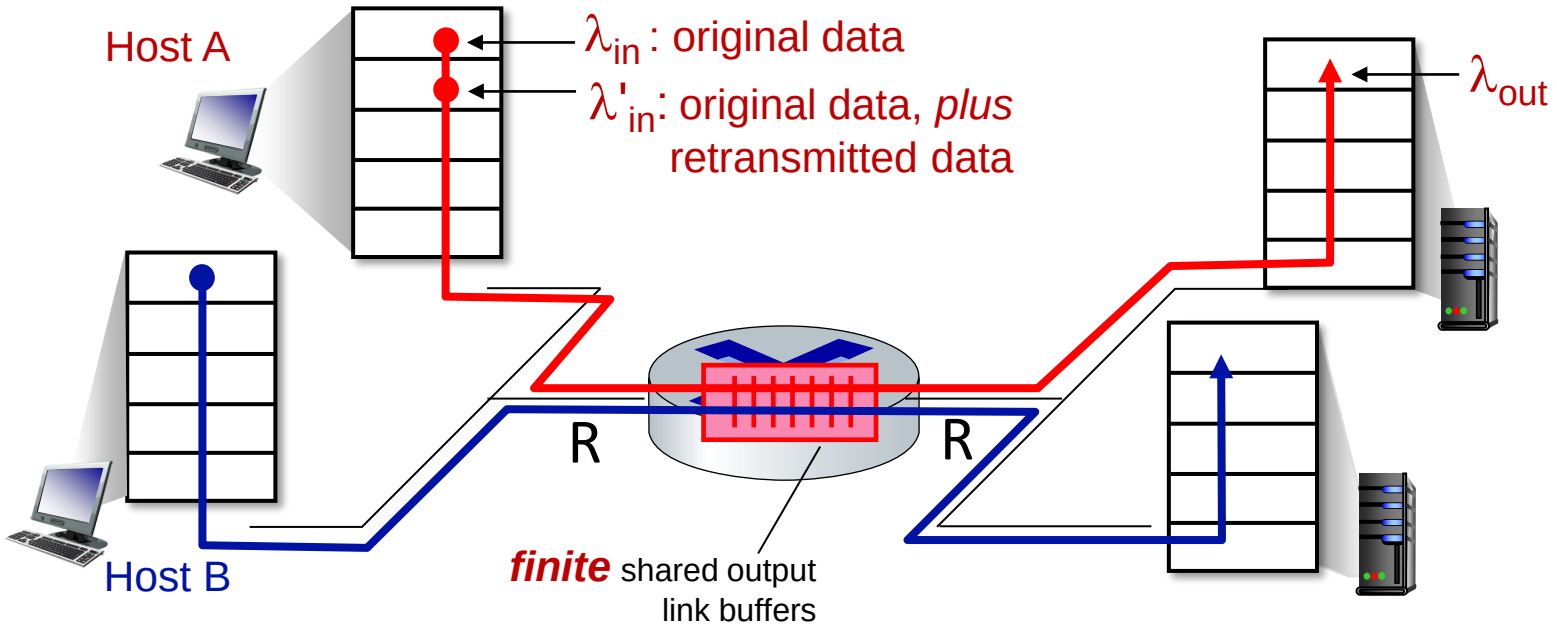
maximum per-connection throughput: R/2



large delays as arrival rate λ_{in} approaches capacity

Causes/costs of congestion: scenario 2

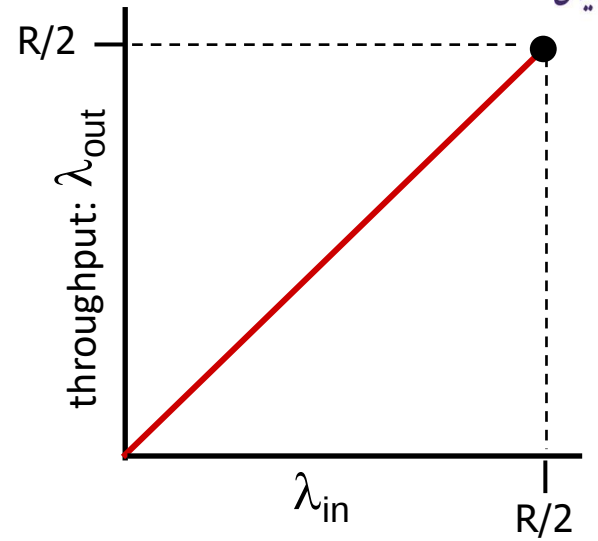
- one router, *finite* buffers
- sender retransmits lost, timed-out packet
  - application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
  - transport-layer input includes *retransmissions* : $\lambda'_{in} \geq \lambda_{in}$

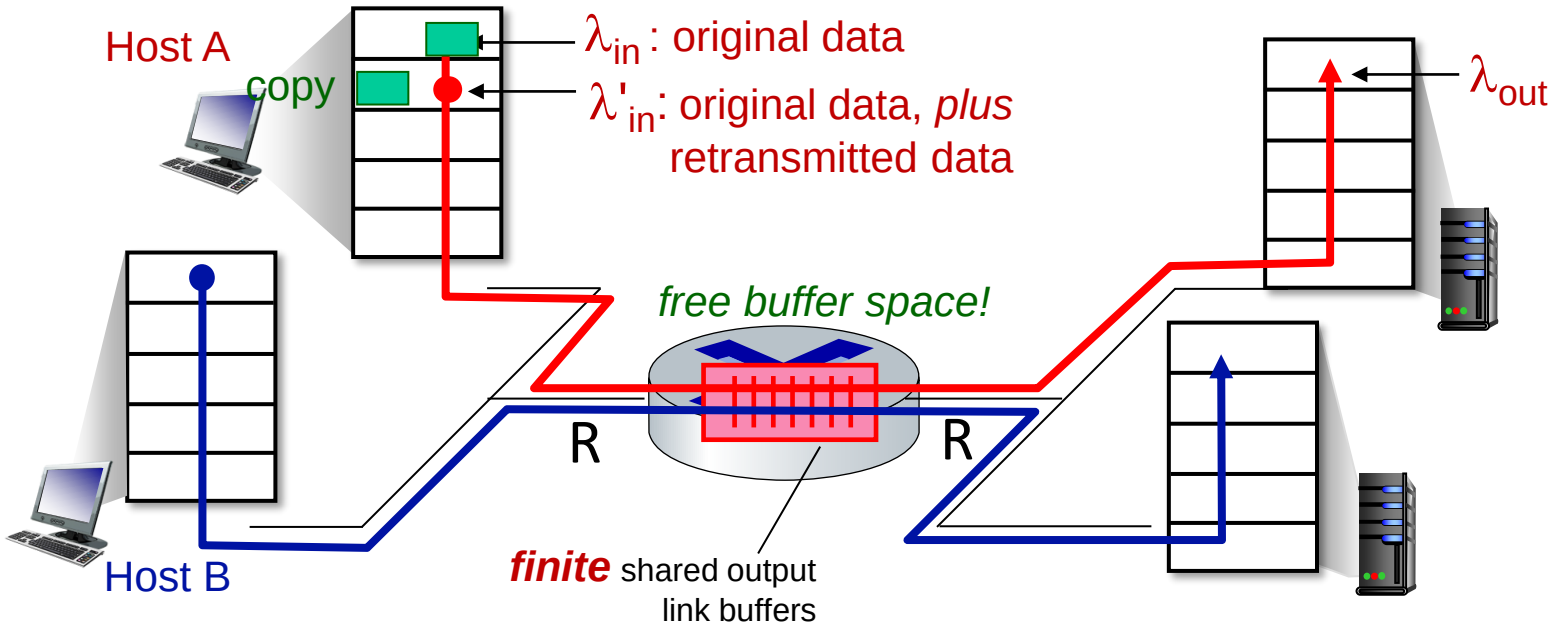


Causes/costs of congestion: scenario 2

Idealization: perfect knowledge

- sender sends only when router buffers available

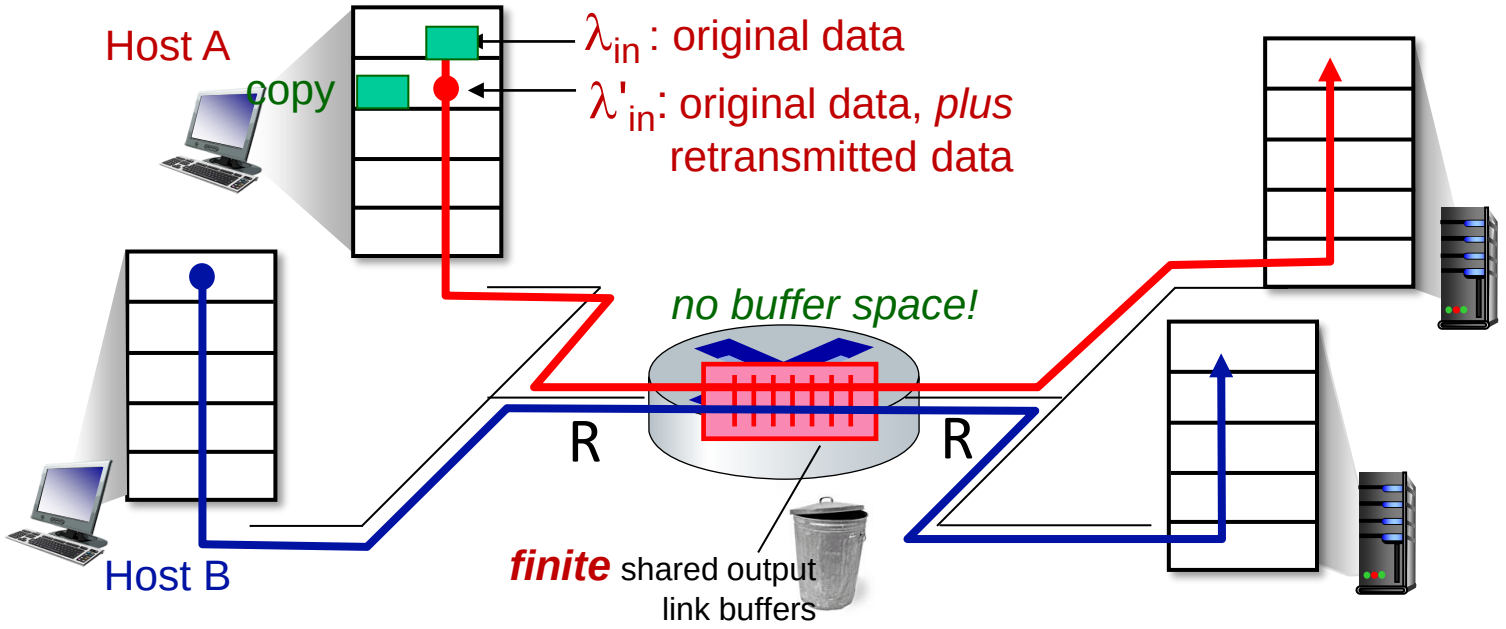




Causes/costs of congestion: scenario 2

Idealization: *some* perfect knowledge

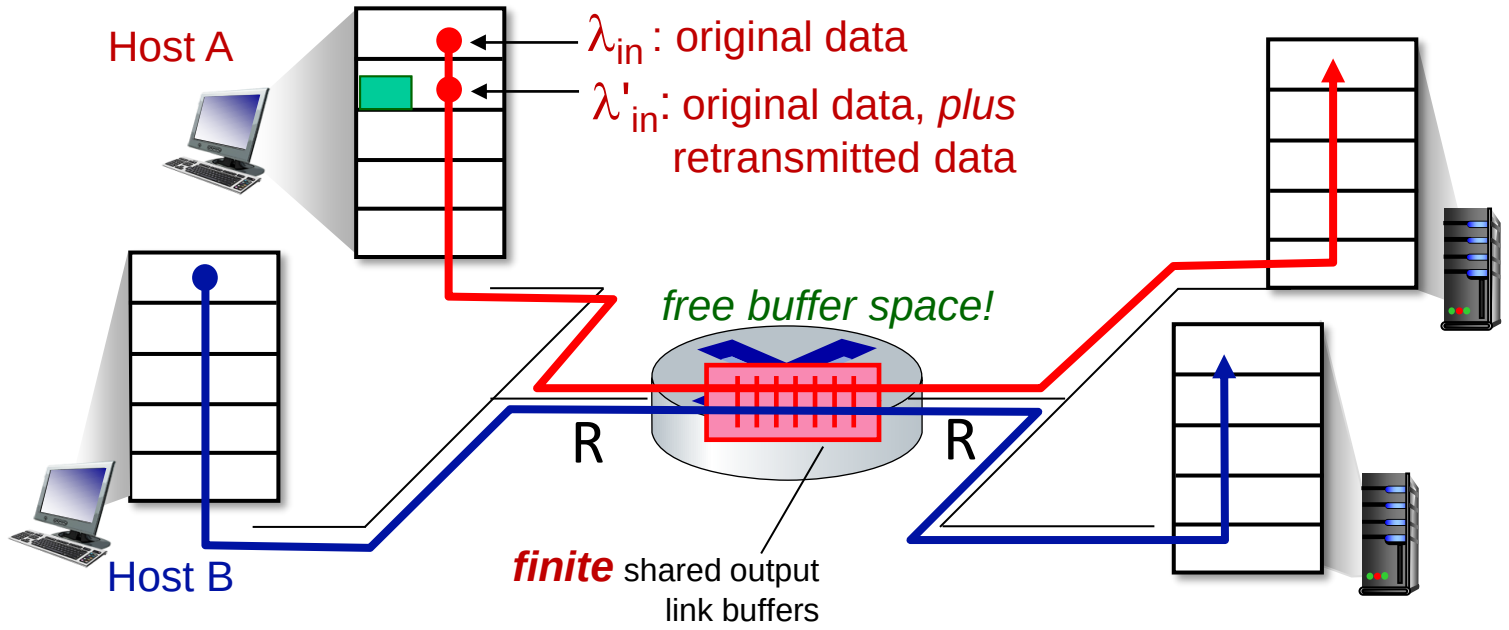
- packets can be lost (dropped at router) due to full buffers
- sender knows when packet has been dropped: only resends if packet *known* to be lost

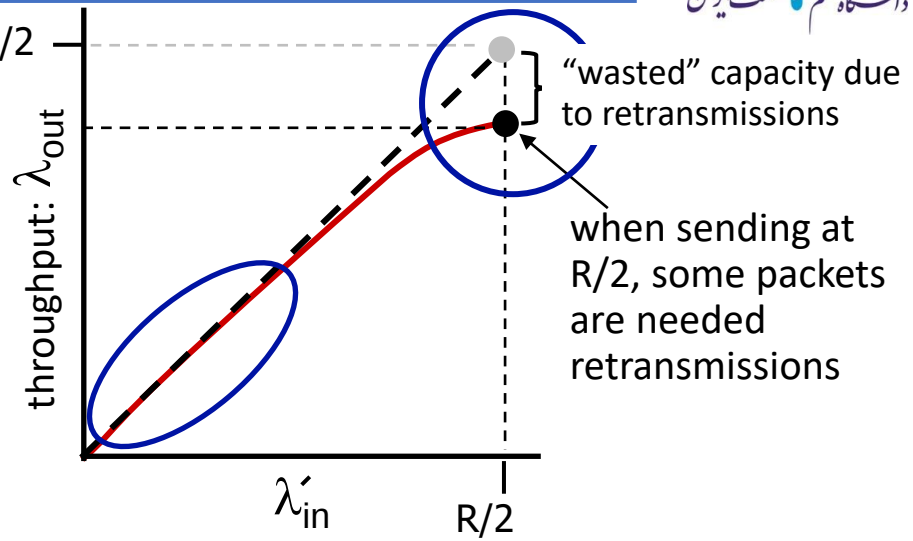


Causes/costs of congestion: scenario 2

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

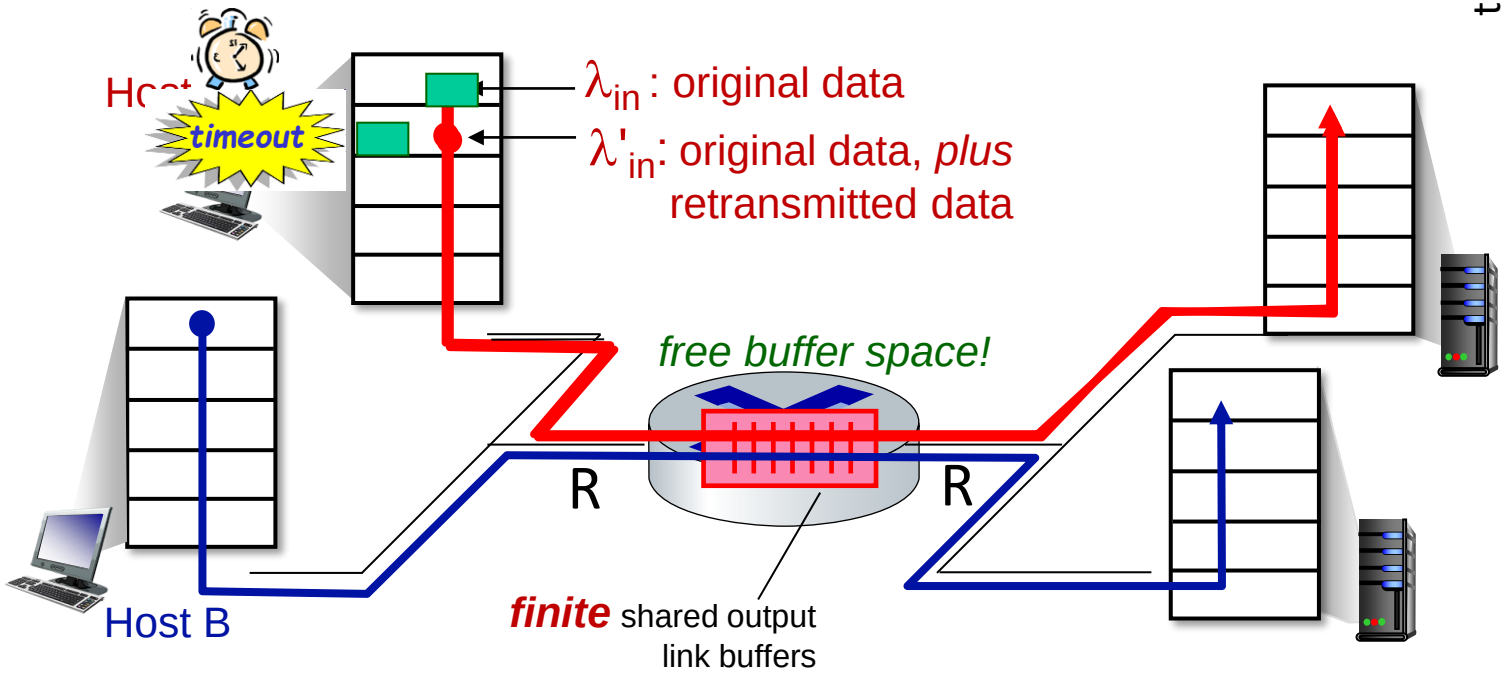


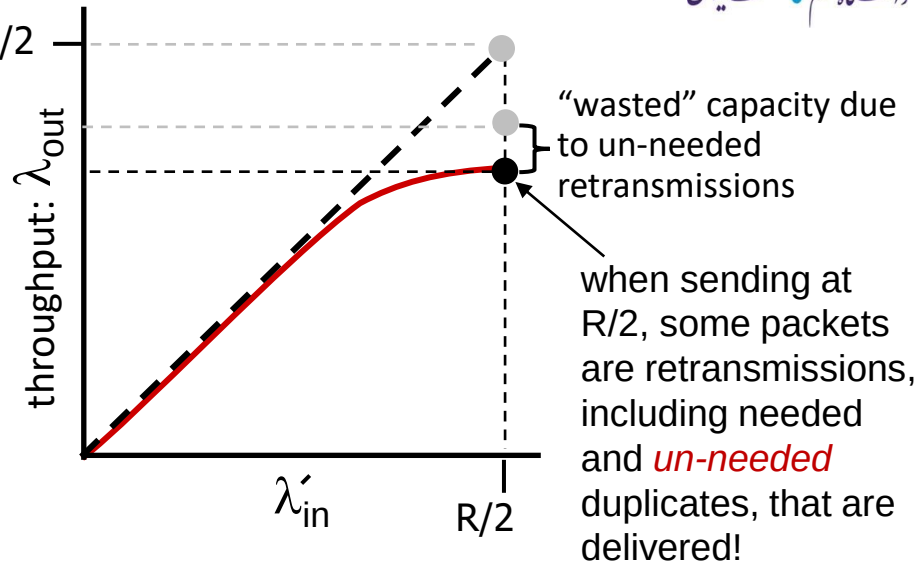


Causes/costs of congestion: scenario 2

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

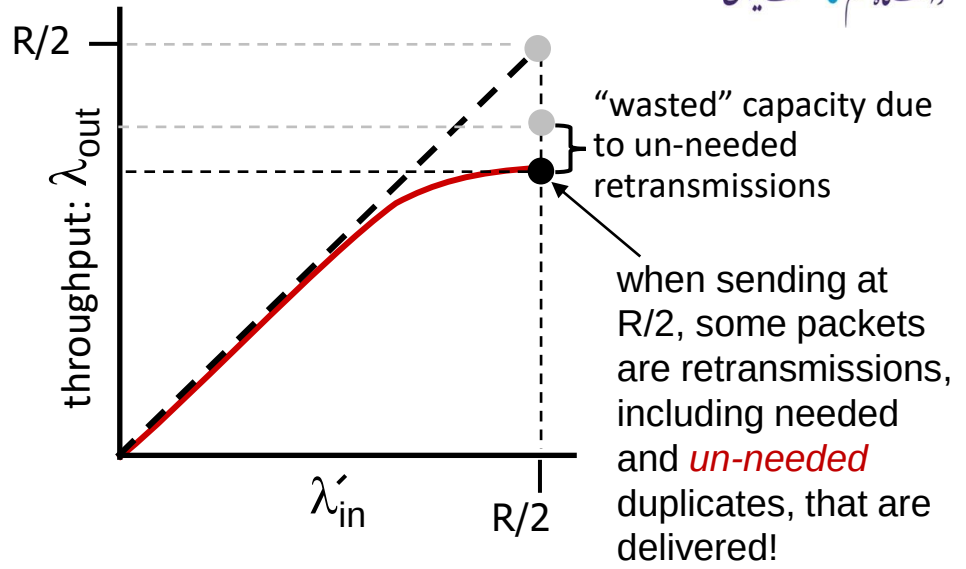




Causes/costs of congestion: scenario 2

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:

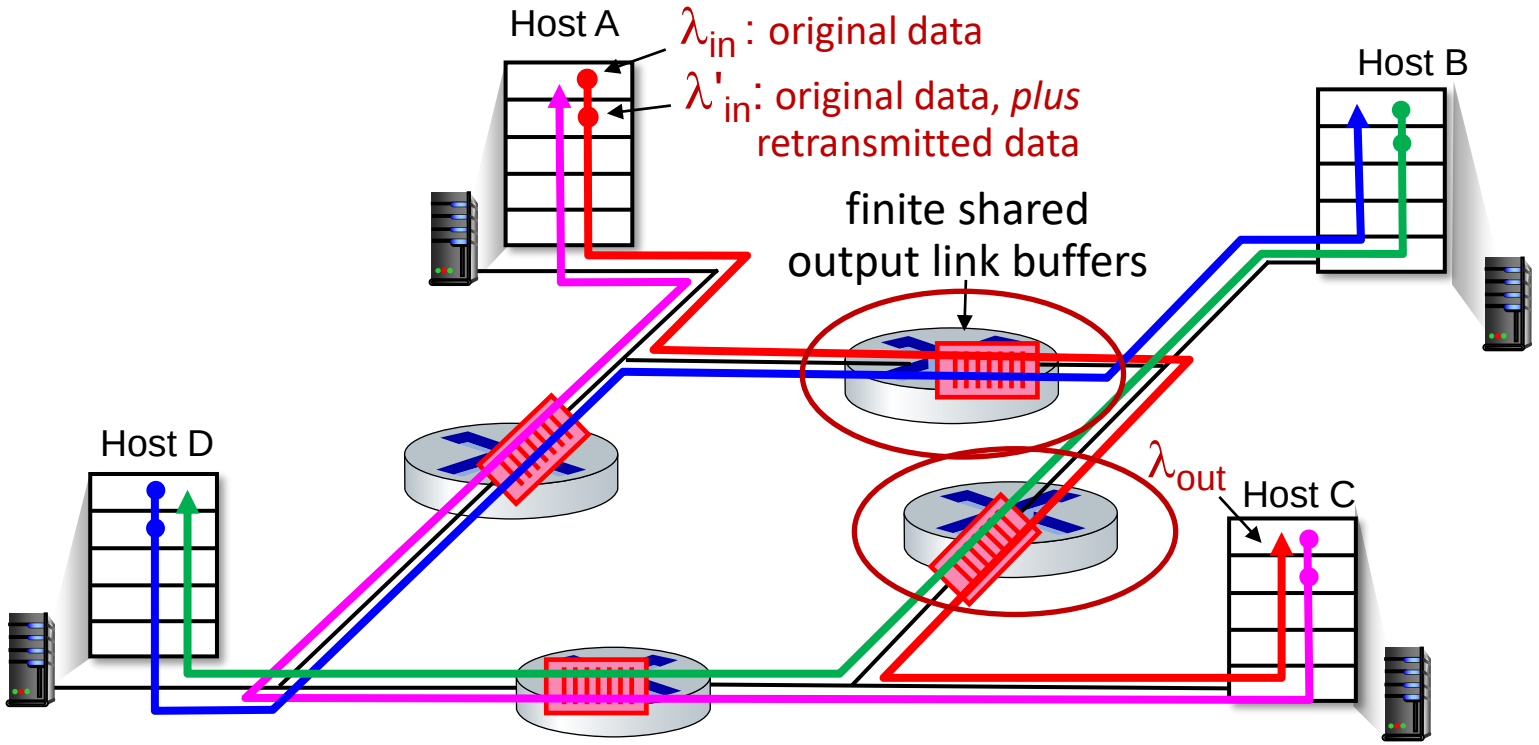
- more work (retransmission) for given receiver throughput
- unneeded retransmissions: link carries multiple copies of a packet
  - decreasing maximum achievable throughput

Causes/costs of congestion: scenario 3

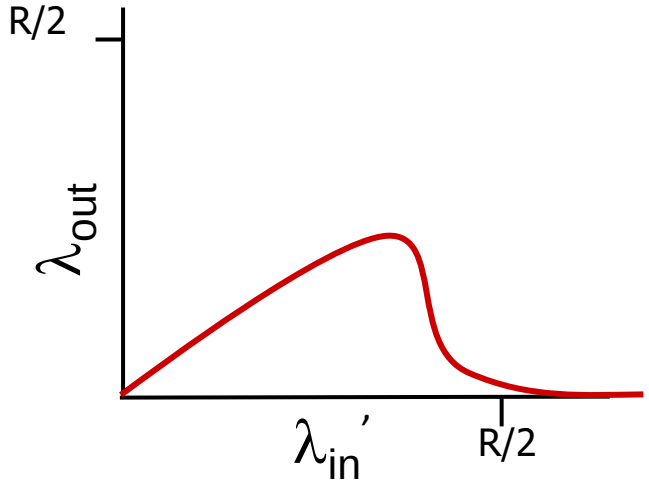
- *four* senders
- *multi-hop* paths
- timeout/retransmit

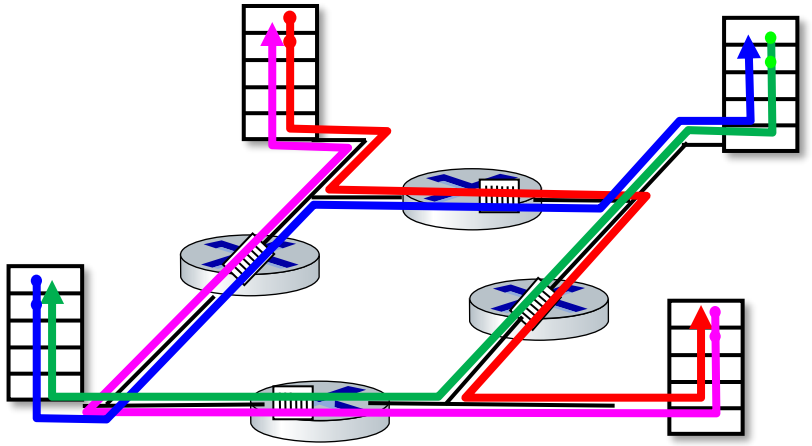
Q: what happens as λ_{in} and λ_{in}' increase ?

A: as red λ_{in}' increases, all arriving blue pkts at upper queue are dropped, blue throughput → 0



Causes/costs of congestion: scenario 3



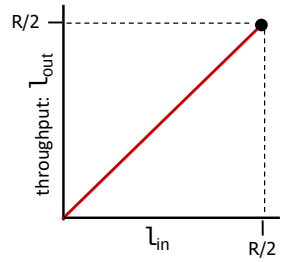


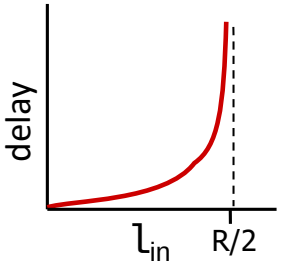
another "cost" of congestion:

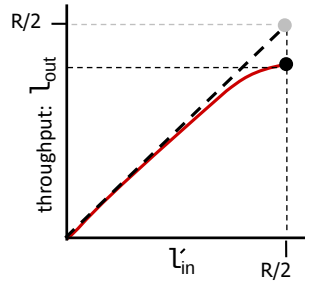
- when packet dropped, any upstream transmission capacity and buffering used for that packet was wasted!

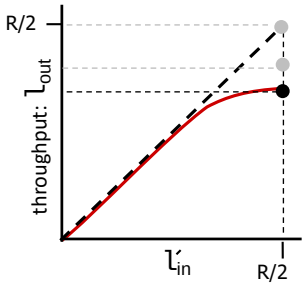
Causes/costs of congestion: insights

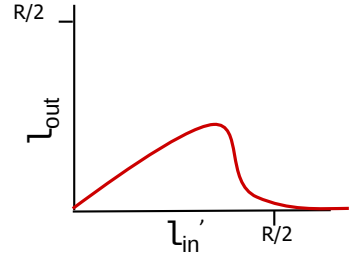
- throughput can never exceed capacity
- delay increases as capacity approached
- loss/retransmission decreases effective throughput
- un-needed duplicates further decreases effective throughput
- upstream transmission capacity / buffering wasted for packets lost downstream











What should we do to encounter congestion?

Congestion Control

Congestion control and relation to resource allocation

- Congestion control and resource allocation are **two sides of the same coin**.

- **On the one hand**, if the network takes an active role in allocating resources—for example, scheduling which virtual circuit gets to use a given physical link during a certain period of time,
  - then congestion may be avoided, thereby making congestion control unnecessary.

- **On the other hand**, you can always let packet sources send as much data as they want and then recover from congestion should it occur.
  - This is the easier approach, but it can be disruptive, because many packets may be discarded by the network before congestion can be controlled.

Congestion Control

Congestion control and relation to resource allocation

- Furthermore, it is precisely at those times when the network is congested—that is, resources have become scarce relative to demand—that the need for resource allocation among competing users is most keenly felt.

- There are also **solutions in the middle**, whereby inexact allocation decisions are made, but congestion can still occur, and hence some mechanism is still needed to recover from it.

- Whether you call such a mixed solution congestion control or resource allocation does not really matter. In some sense, it is both.

Congestion Control

Congestion control and relation to resource allocation: Where it is done

- Congestion control and resource allocation involve both **hosts** and **network elements** such as routers.
- In network elements, various **queuing disciplines** can be used to control the order in which packets get **transmitted** and which packets get **dropped**.
- The queuing discipline can also segregate traffic to keep one user's packets from unduly affecting another user's packets.
- At the **end hosts**, the congestion control mechanism paces how fast sources are allowed to send packets. This is done in an effort to keep congestion from occurring in the first place and, should it occur, to help eliminate the congestion

Congestion Control

Congestion control and relation to resource allocation: Deeper Analysis

- Resource allocation and congestion control are complex issues that have been the subject of much study ever since the first network was designed.

- One factor that makes these issues complex is that **they are not isolated to one single level of a protocol hierarchy.**

- **Resource allocation** is partially implemented in the **routers**, **switches**, and **links** inside the network and partially in the **transport protocol** running on the **end hosts**.

- **End systems** may use **signaling protocols** to convey their resource requirements to network nodes, which respond with information about resource availability.

Congestion Control

Congestion control and relation to resource allocation: Deeper Analysis

- By **resource allocation**, we mean the process by which **network elements** try to **meet** the **competing demands** that applications have for network **resources**— primarily link bandwidth and buffer space in routers or switches.

- Of course, it will often not be possible to meet all the demands, meaning that some users or applications may receive fewer network resources than they want.

- Part of the resource allocation problem is deciding when to say no and to whom. (Sometimes termed as **Admission Control**)

Congestion Control

Congestion control and relation to resource allocation: Deeper Analysis

- We use the term **congestion control** to describe the efforts made by network nodes to prevent or respond to overload conditions.

- Since congestion is generally bad for everyone, the first order of business is making congestion subside, or preventing it in the first place.

- This might be achieved simply by persuading a few hosts to stop sending, thus improving the situation for everyone else.