UMASS AMHERST

ECE 671 Introduction to Computer Networks

Week 3 Lesson 5

Transport Layer & Congestion Control

Rationale

- Transport layer is an important part of Internet protocol stack
- Understanding complete protocol stack is important
- Transport layer services are used by application layer
  - Even if someone does not care about networking details, understanding which transport layer protocol to use (and why) is really important
- Congestion control in transport layer shows how resource sharing is achieved without central control

Objectives

- Explain the process of connection setup and teardown in TCP
- Evaluate causes of congestion and possible solutions in a transport layer
- Analyze TCP approaches to mitigate congestion

Prior Knowledge

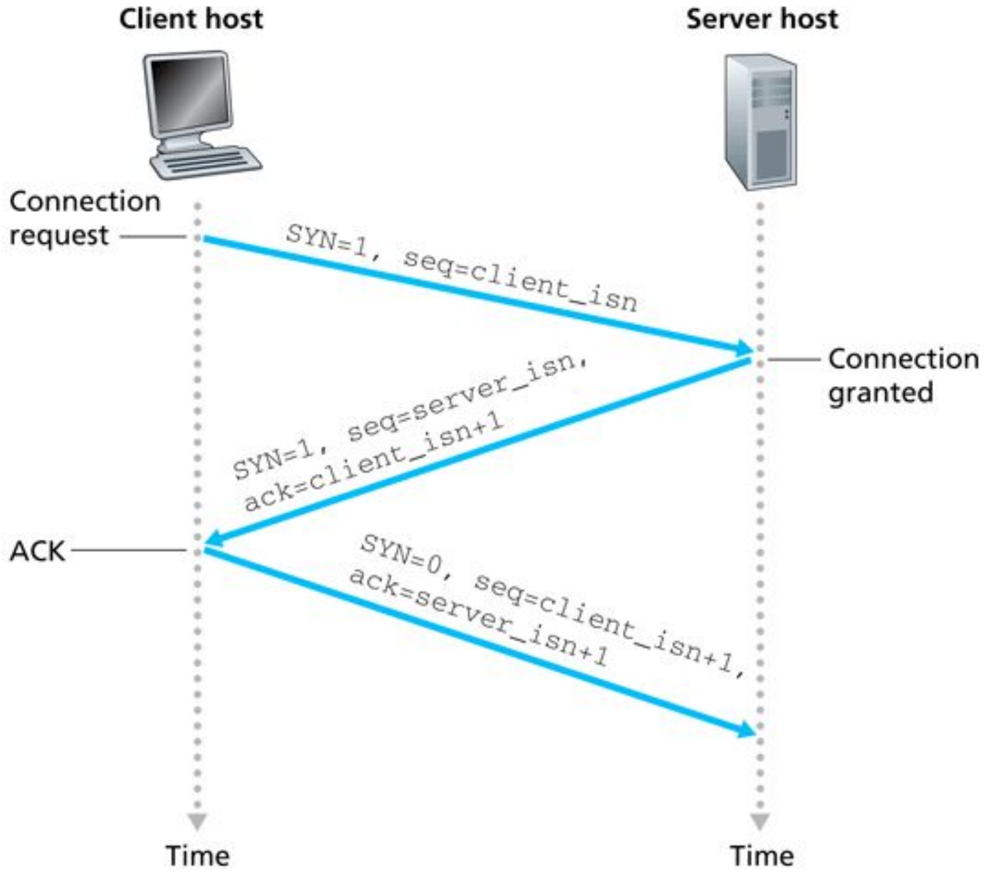
- Transport layer connect application processes
  - Different transport layer protocols depending on requirements
- Transport layer features in last lesson
  - De-/multiplexing
  - Reliability

Orchestrated Discussion (Hand Raise): Lesson Reflection Feedback

- **Discuss questions and comments on Lesson Reflection from prior lesson**

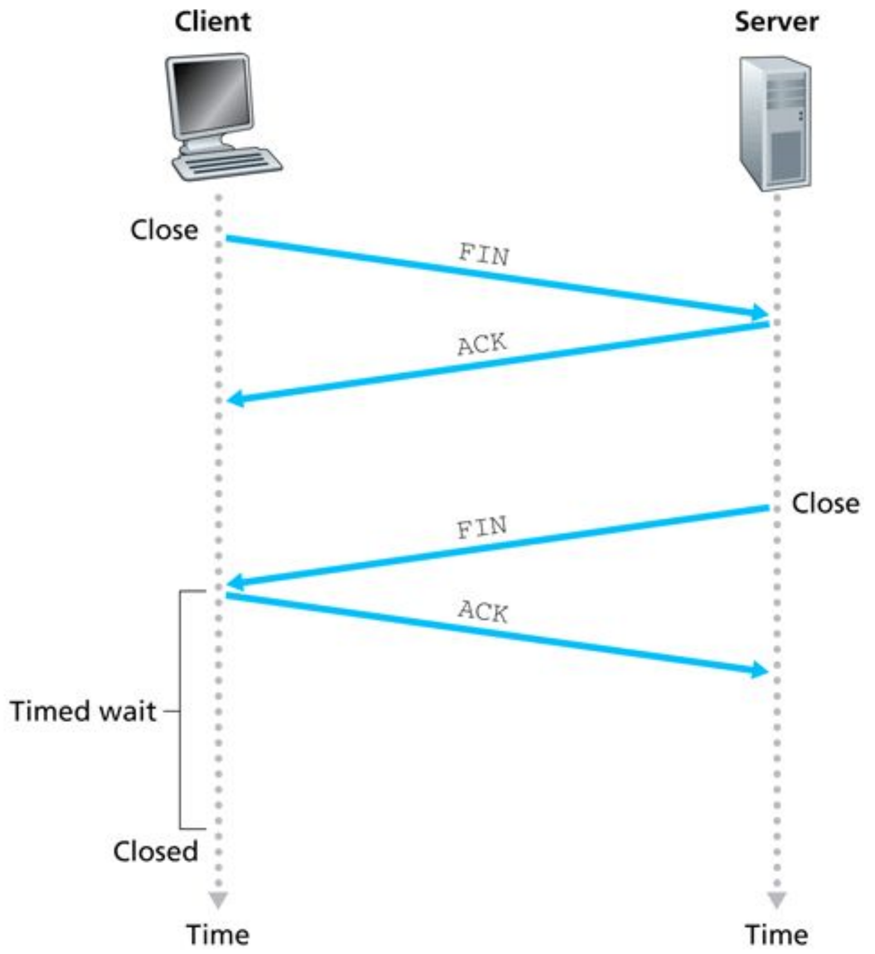
TCP Connection Management

- Connection setup
  - Three-way handshake
    - SYN
    - SYN/ACK
    - ACK
  - SYN counts as one byte
  - ACK may carry data already
  - Flag in header identifies SYN
    - Used in network systems to identify new connections



TCP Connection Management

- Connection teardown
  - Each side closes when transmission is complete
    - FIN
    - ACK
  - Final FIN or ACK may get lost
    - Need to be able to retransmit
  - Connection cleanup after timed wait



Document Cam: Performance Expectations

- **Read "For Impatient Web Users, an Eye Blink Is Just Too Long to Wait" (by Steve Lohr, New York Times, 2/29/2012)**

Group Discussion and Report Back (Short Answer): Performance Expectations

- **Answer the following questions:**
  - **What are users' expectations for Internet interactions?**

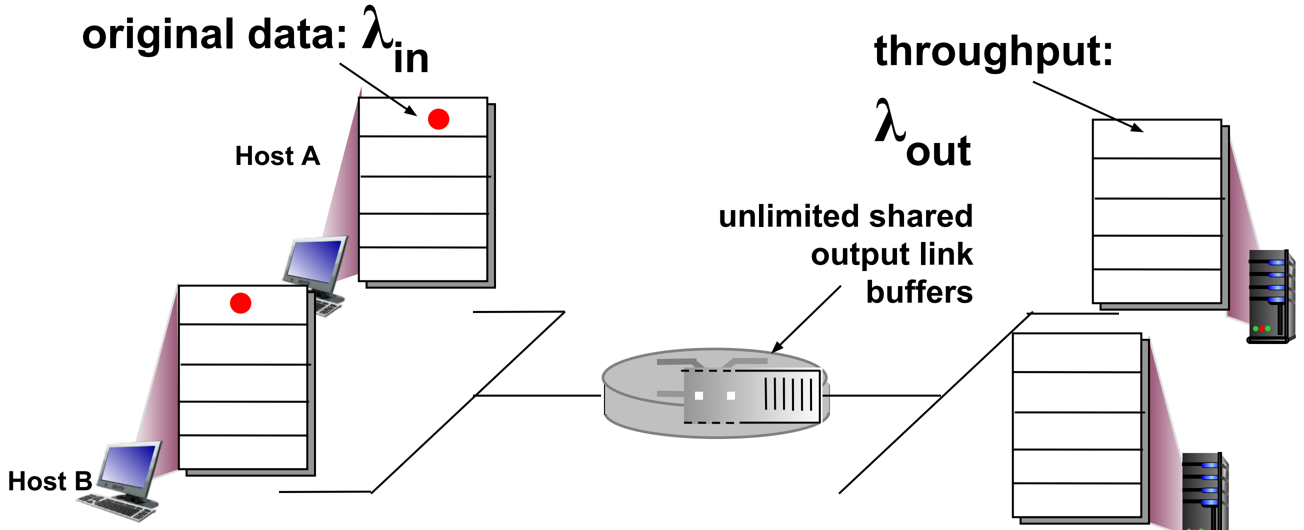
  - **What are possible ways for providers to satisfy users?**

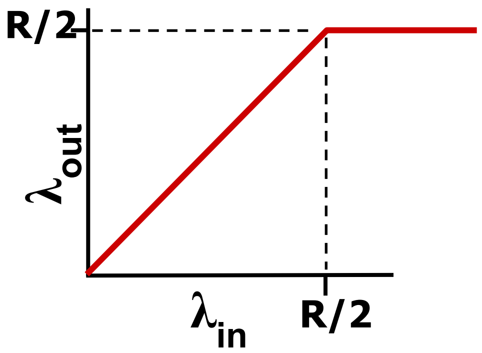
Network Congestion

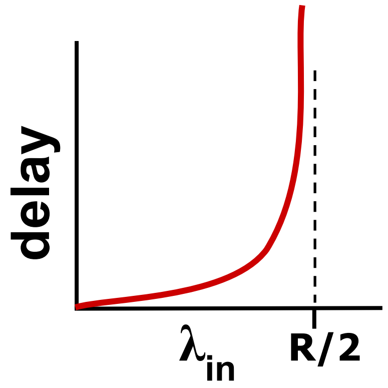
- What is the problem when all providers send always at “full speed”?
  - Network resources are limited
  - Network links will be fully utilized
  - Router buffers may fill up and drop packets
- Network congestion occurs when too much traffic is sent by end-systems
  - Can be localized: congested link
  - Can be larger: multiple congested links and routers

Whiteboard: Cause of Congestion - Scenario 1

- Baseline scenario:
  - Two senders, two receivers
  - one router, infinite buffers
  - output link capacity: R
  - No retransmission





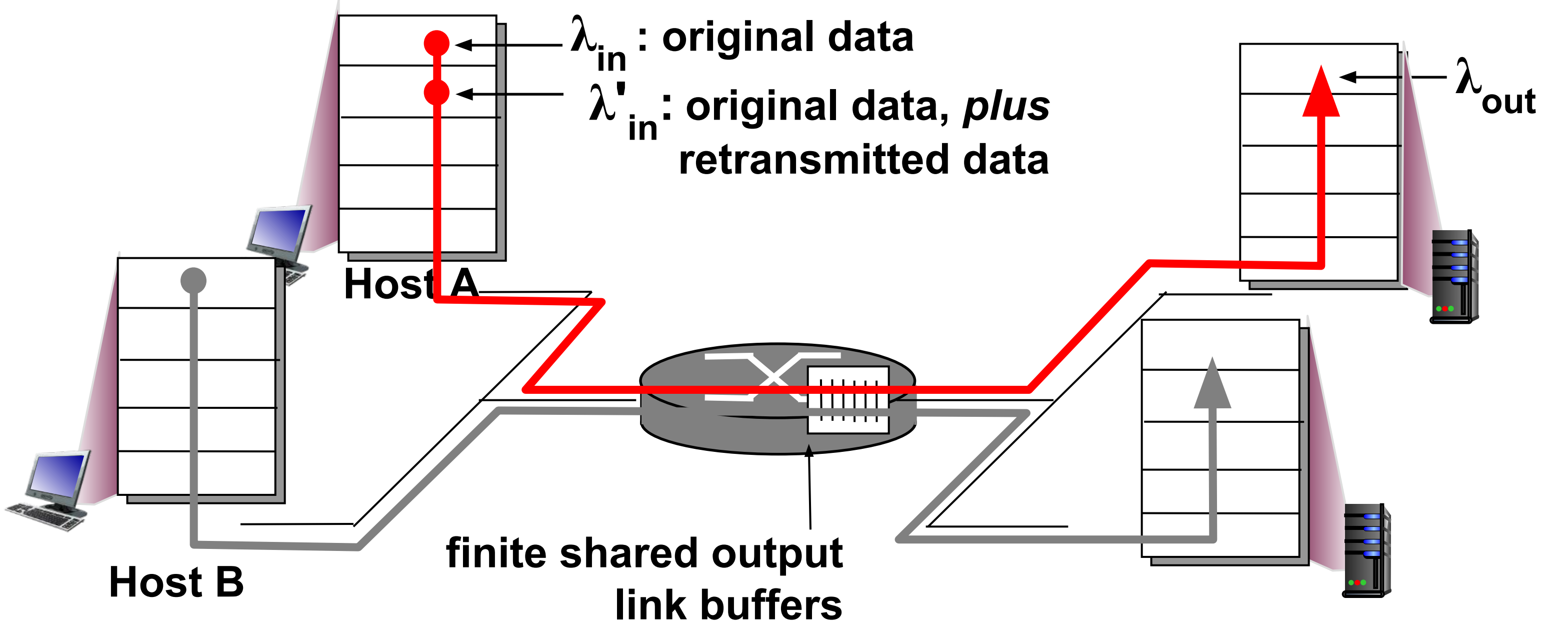


- maximum per-connection throughput: R/2

- large delays as arrival rate, λ_{in}, approaches capacity

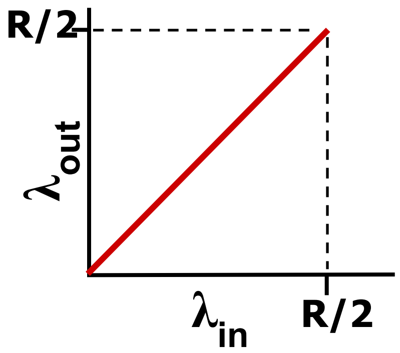
Cause of Congestion: Scenario 2

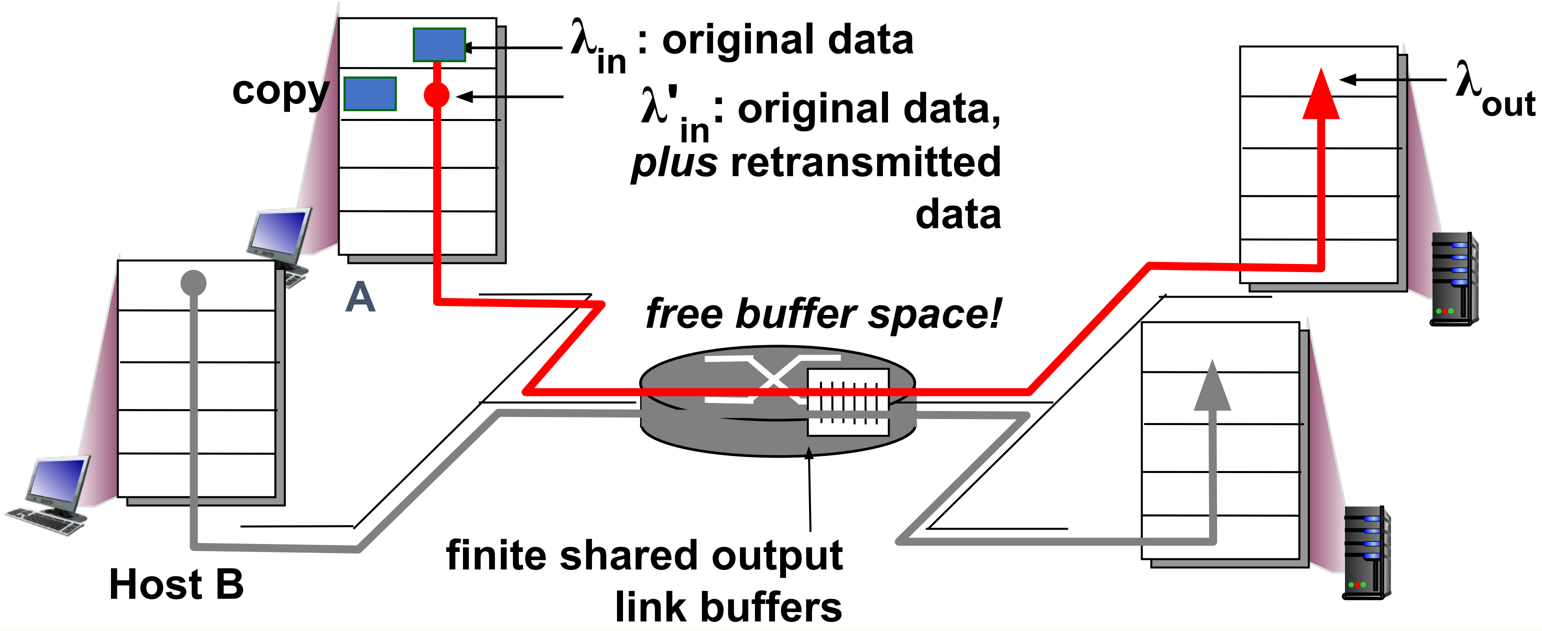
- One router, <u>finite</u> buffers
- Sender retransmission of timed-out packet
  - Application-layer input = application-layer output: $\lambda_{in} = \lambda_{out}$
  - Transport-layer input includes <u>retransmissions</u> : $\lambda_{in}' \geq \lambda_{in}$



Cause of Congestion: Scenario 2

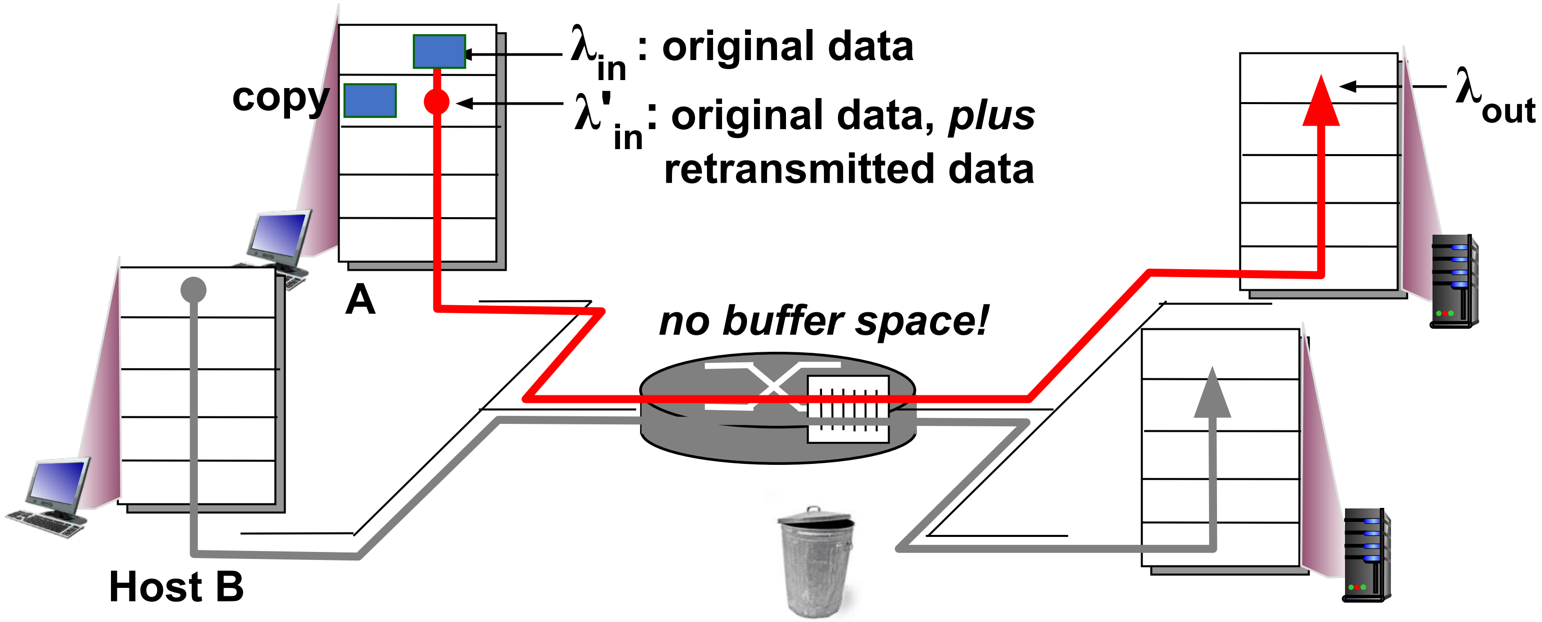
- Idealization: perfect knowledge
  - Sender sends only when router buffers available





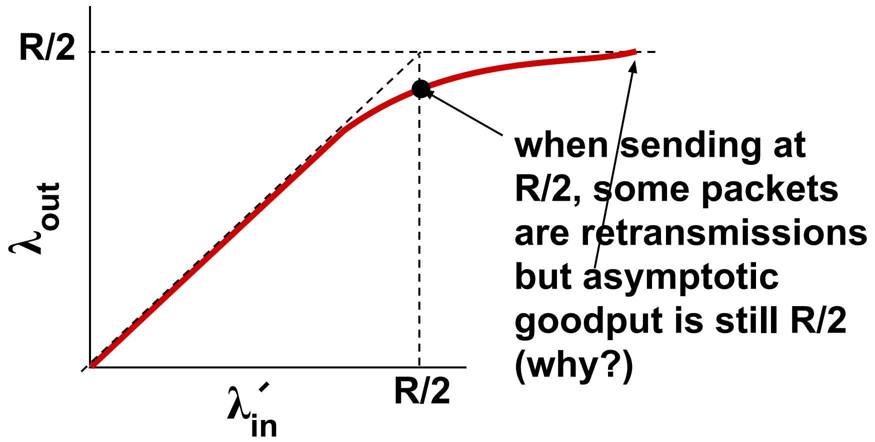
Cause of Congestion: Scenario 2

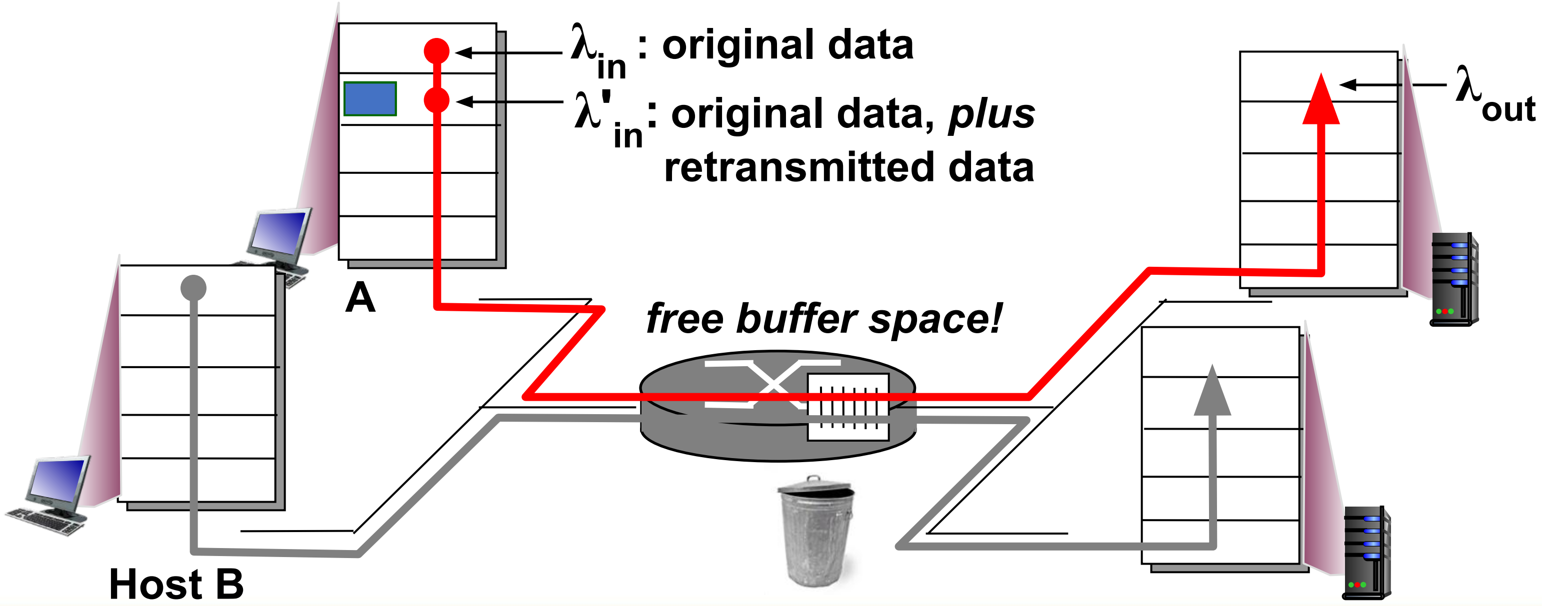
- <u>Idealization: known loss</u>
  - Packets can be lost, dropped at router due to full buffers
  - Sender only resends if packet <u>known</u> to be lost



Cause of Congestion: Scenario 2

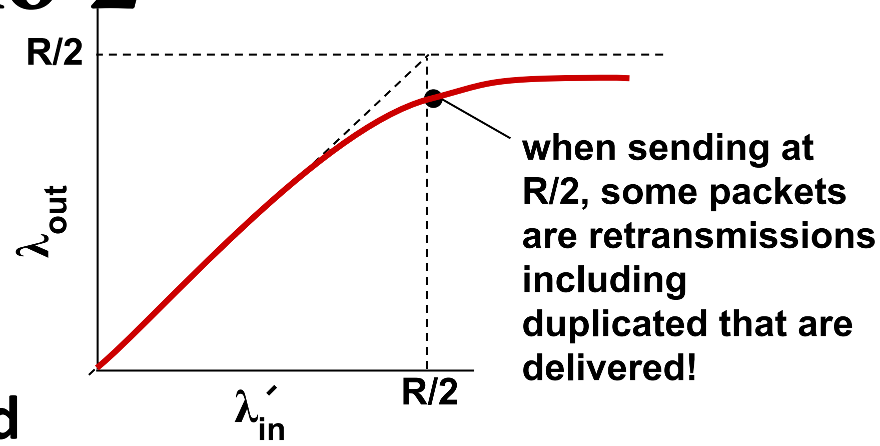
- **Idealization: known loss**
  - Packets can be lost, dropped at router due to full buffers
  - Sender only resends if packet known to be lost

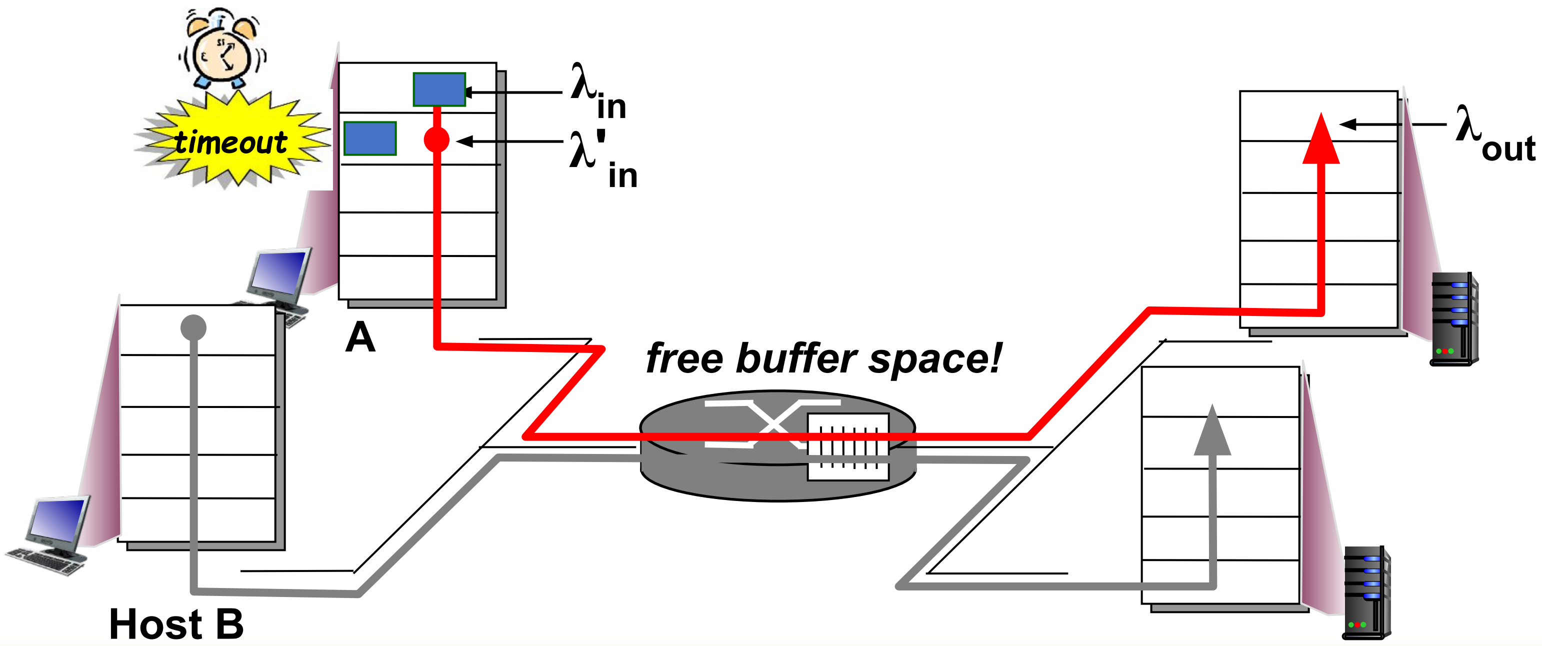




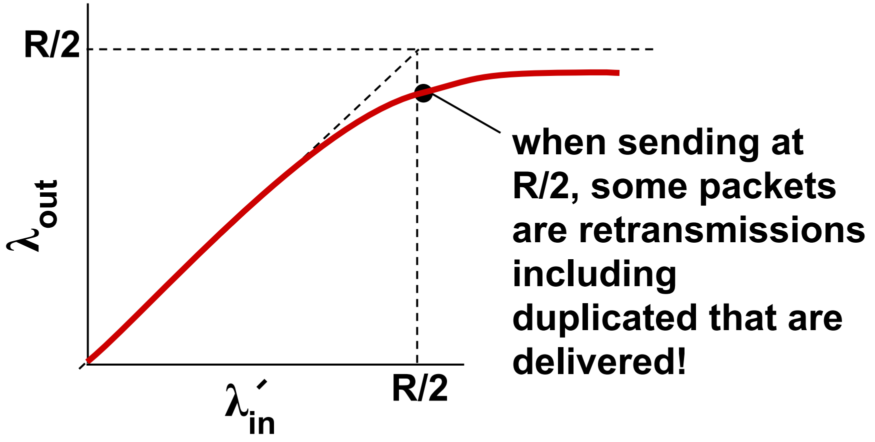
Cause of Congestion: Scenario 2

- **Realistic: duplicates**
  - Packets can be lost, dropped at router due to full buffers
  - Sender times out prematurely, sending two copies, both of which are delivered





Cause of Congestion: Scenario 2



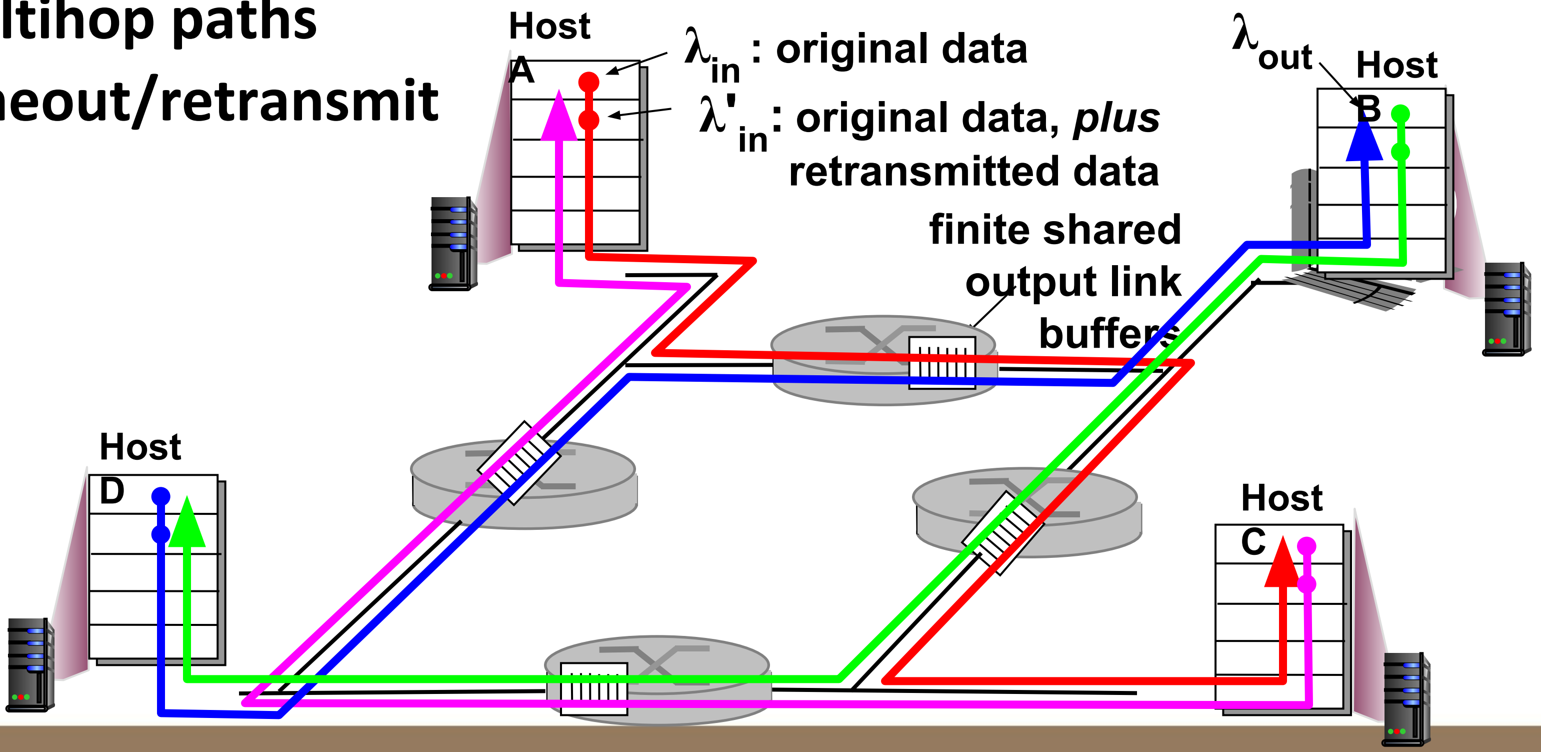
- **<u>Realistic: duplicates</u>**
  - Packets can be lost, dropped at router due to full buffers
  - Sender times out prematurely, sending two copies, both of which are delivered
- "Costs" of congestion:
  - More work (retransmission) for given "goodput"
  - Unneeded retransmissions: link carries multiple copies of packets
  - Decreasing goodput

Cause of Congestion: Scenario 3

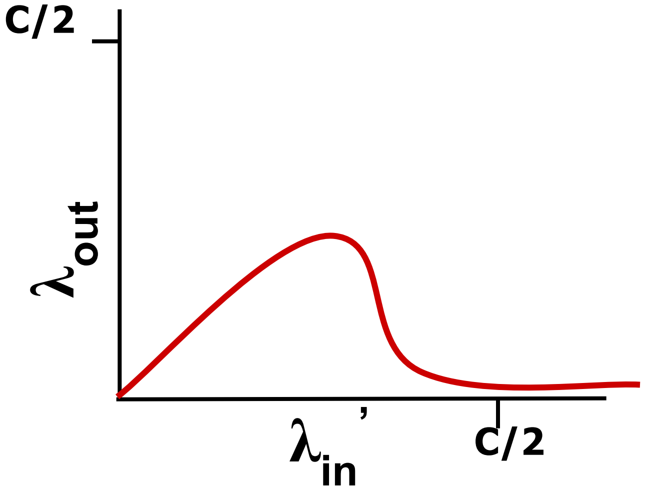
Q: What happens as λ_{in} and λ_{in}' increase ?

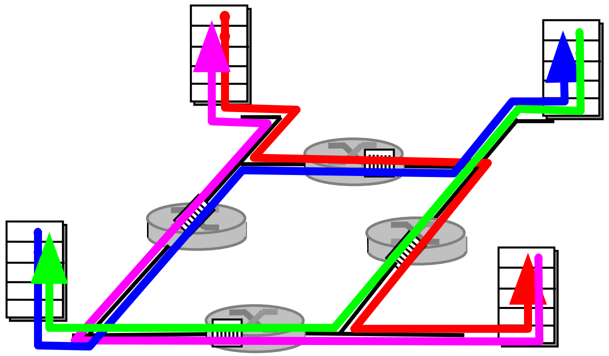
A: As red λ_{in}' increases, all arriving blue packets at upper queue are dropped, blue throughput → 0

- Scenario:
  - Four senders
  - Multihop paths
  - Timeout/retransmit



Cause of Congestion: Scenario 3





- Another "cost" of congestion:
  - When packet dropped, any "upstream transmission capacity used for that packet was wasted!

Orchestrated Discussion (Hand Raise): Solving the Congestion Problem

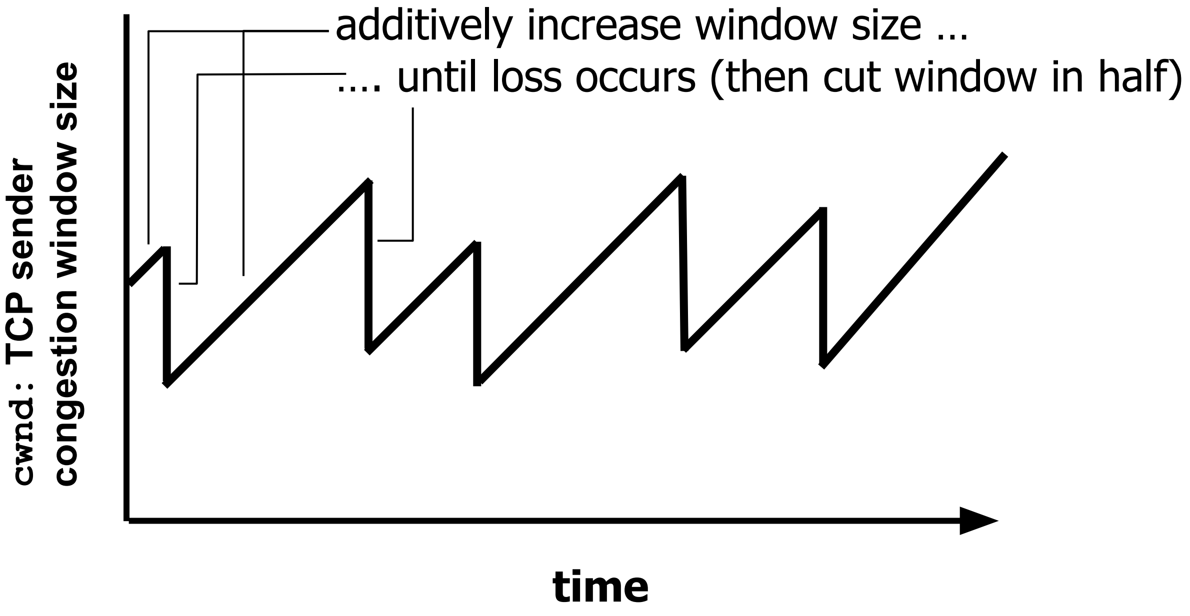
- What can we do about the congestion problem?
  - We will brainstorm as a class

Congestion Control Approaches

- Reserving resources (cf. circuit switching)
  - As for permission before sending
  - Send at permitted rate
  - Congestion can be avoided through managing reservations
- Adapting sending rate (cf. packet switching)
  - Detect congestion (e.g., packet losses cause NAKs)
  - Reduce sending rate to avoid congestion collapse
- Tradeoff: central control vs. distributed approach

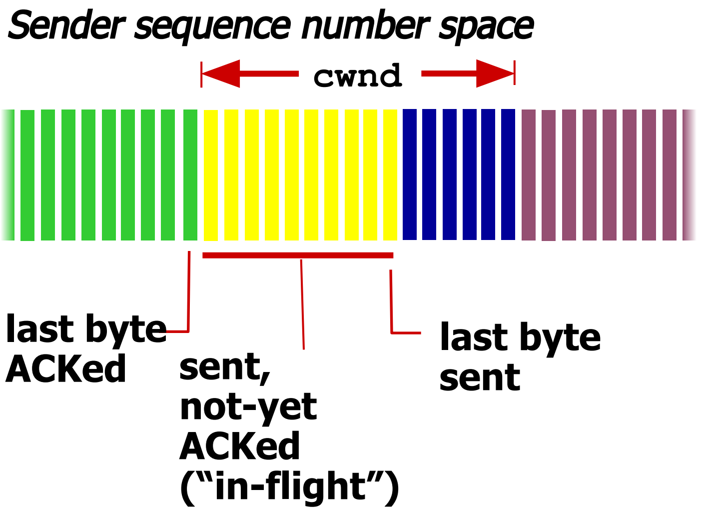
TCP Congestion Control

- **Additive increase, multiplicative decrease (AIMD)**
- **<u>Approach:</u> sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs**
  - **<u>Additive increase:</u> increase `cwnd` by 1 MSS every RTT until loss detected**
  - **<u>Multiplicative decrease:</u> cut `cwnd` in half after loss**



AIMD saw tooth behavior: probing for bandwidth

TCP Congestion Control



- **Sender limits transmission:**

$$\texttt{LastByteSent-LastByteAcked} \leq \texttt{cwnd}$$

- **`cwnd` is dynamic, function of perceived network congestion**

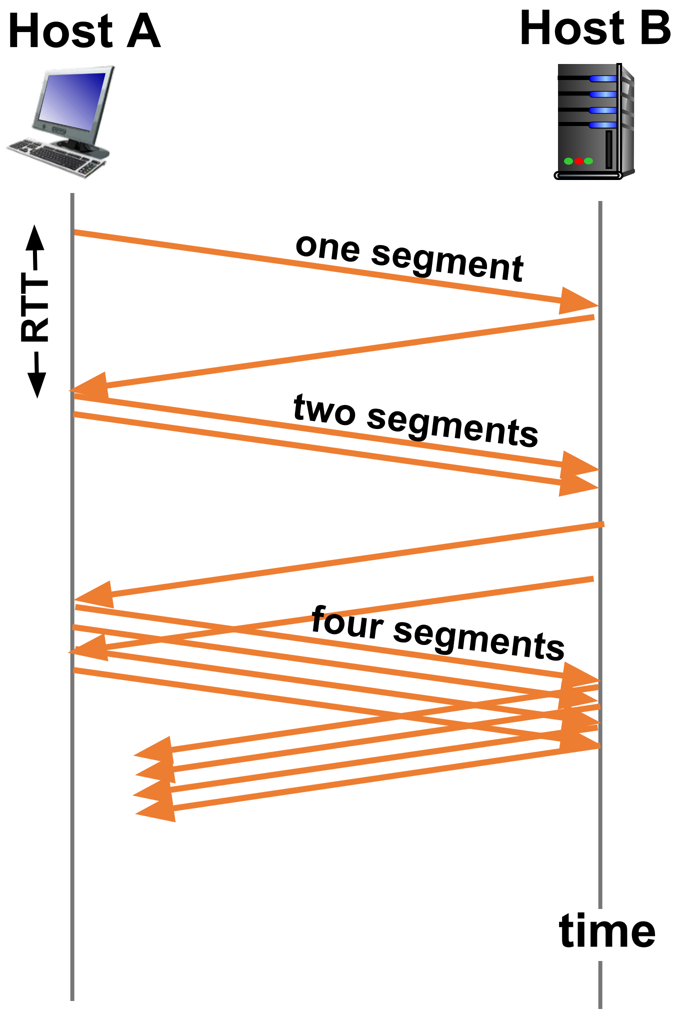
***TCP* sending rate:**

- **Roughly: send cwnd bytes, wait RTT for ACKS, then send more bytes**

$$\text{rate} \approx \frac{\text{cwnd}}{\text{RTT}} \text{ bytes/sec}$$

TCP Slow Start

- When connection begins, increase rate exponentially until first loss event:
  - Initially `cwnd` = 1 MSS
  - Double `cwnd` every RTT
  - Done by incrementing `cwnd` for every ACK received
- **Summary:** initial rate is slow but ramps up exponentially fast



TCP Loss Detection and Reaction

- Loss indicated by timeout:
  - `cwnd` set to 1 MSS;
  - Window then grows exponentially (as in slow start) to threshold, then grows linearly
- Loss indicated by 3 duplicate ACKs: TCP RENO
  - Duplicate ACKs indicate the network is capable of delivering some segments
  - `cwnd` is cut in half, window then grows linearly
- TCP Tahoe always sets `cwnd` to 1 (timeout or 3 duplicate ACKs)

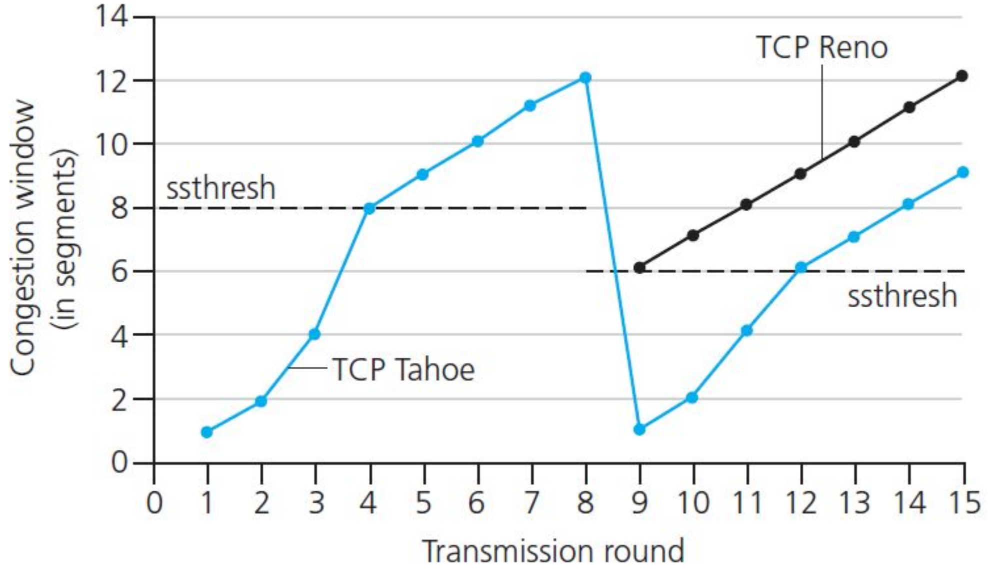
TCP Slow Start and Congestion Avoidance

Q: When should the exponential increase switch to linear?

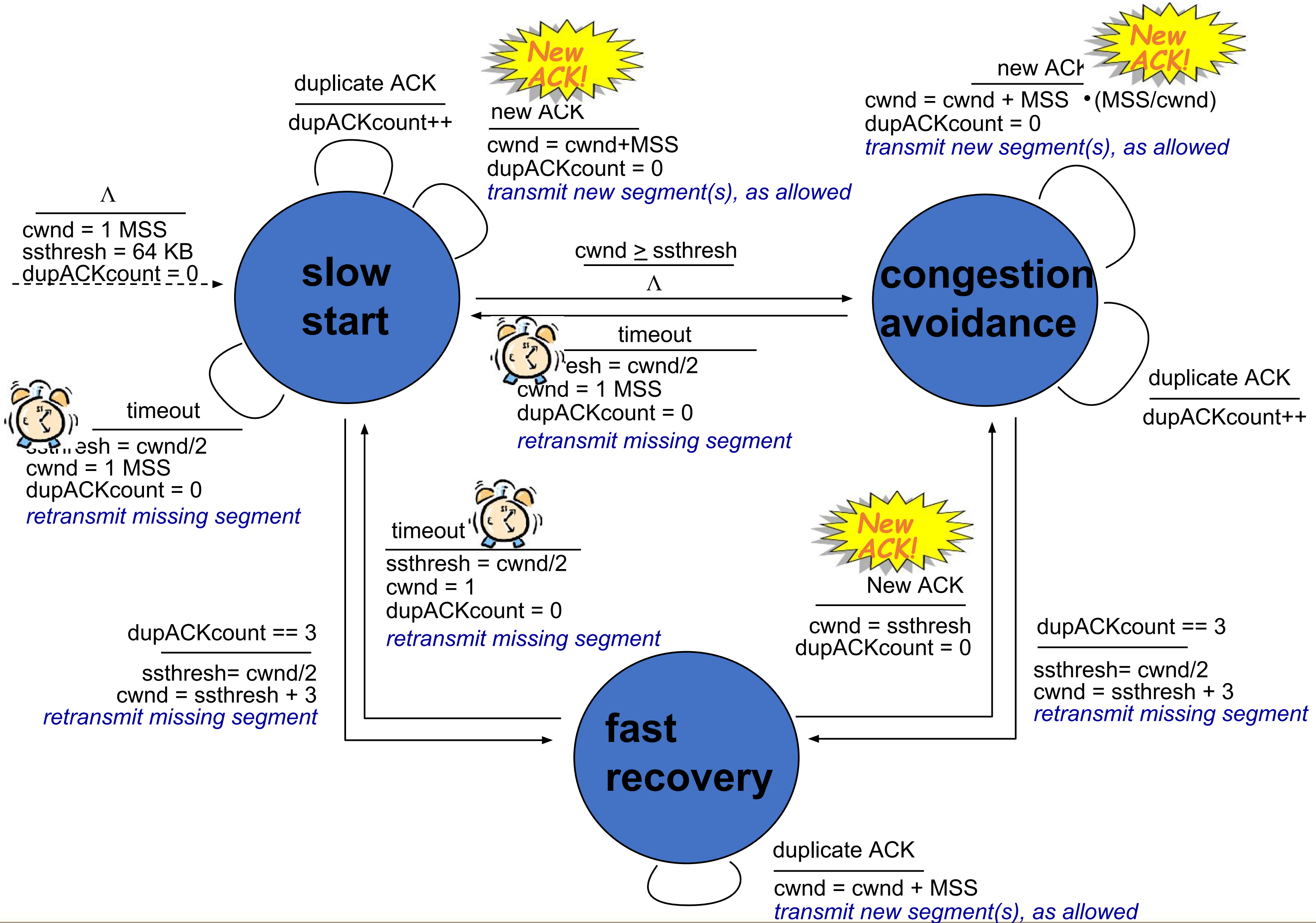
A: When `cwnd` gets to 1/2 of its value before timeout.

Implementation:

- Variable `ssthresh`
- On loss event, `ssthresh` is set to 1/2 of `cwnd` just before loss event

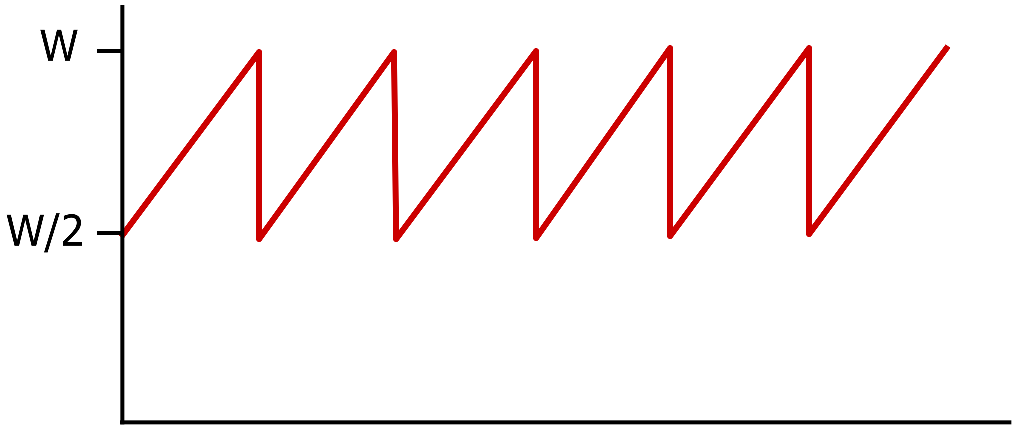


TCP Congestion Control Summary



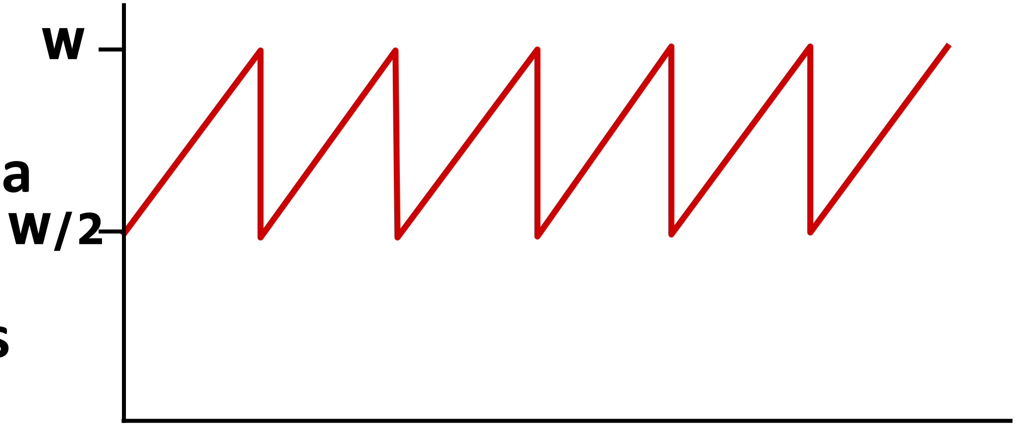
Poll: TCP Throughput

- **What is the average TCP throughput as a function of window size W and RTT?**
  - **Ignore slow start, assume always data to send**



TCP Throughput

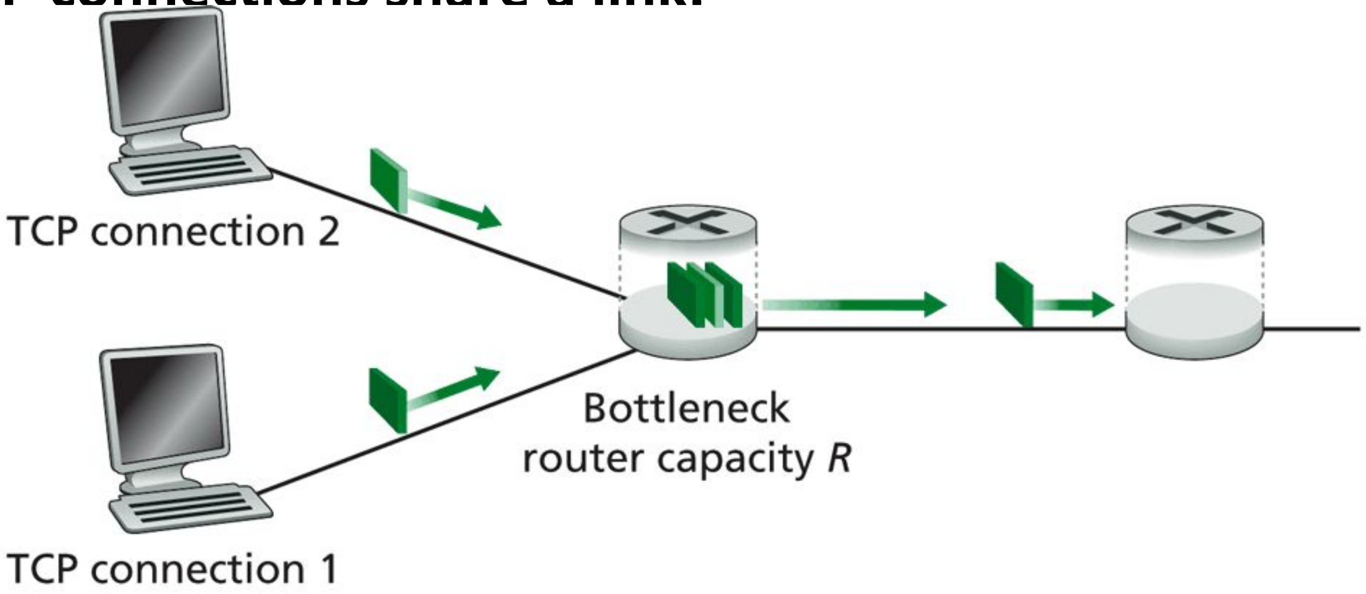
- What is the average TCP throughput as function of window size W and RTT?
  - Ignore slow start, assume always data to send
- W: window size (measured in bytes) where loss occurs
  - Average window size (# in-flight bytes) is ¾ W
  - Average throughput is 3/4W per RTT



$$\text{Avg. TCP throughput} = \frac{3}{4}\frac{W}{RTT} \text{ bytes/sec}$$

Link Sharing with TCP

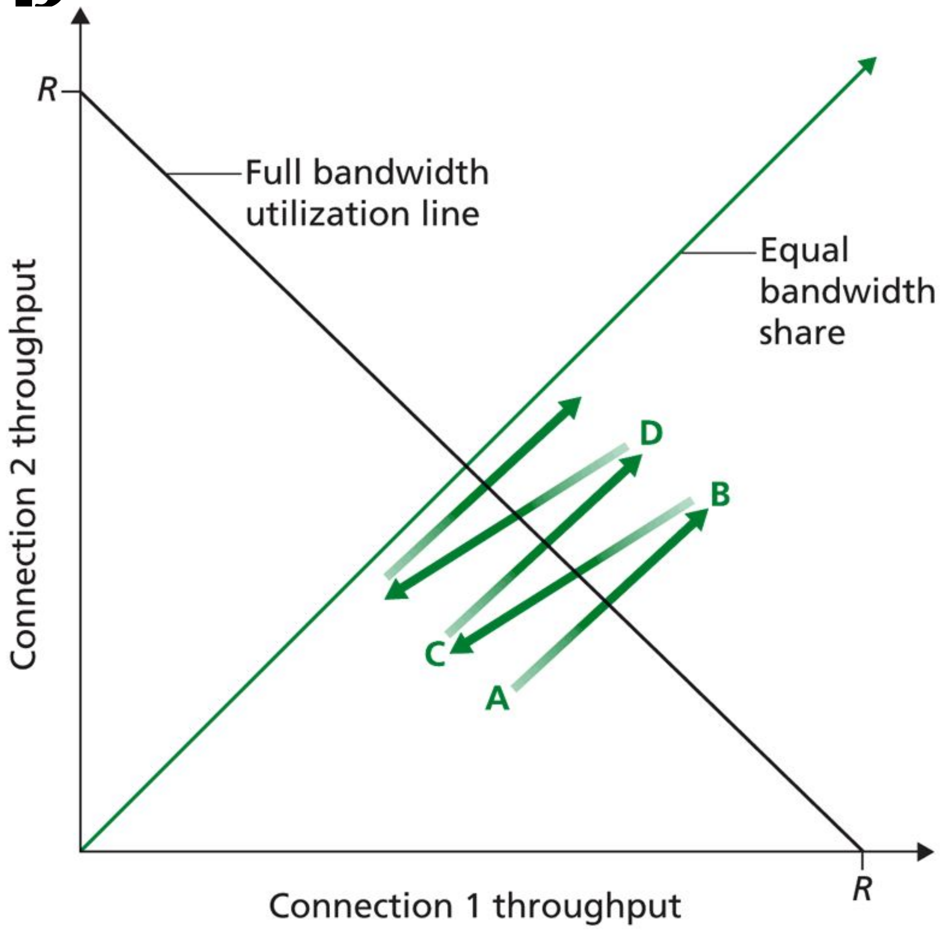
- **Two TCP connections share a link:**



- **Eventually each connection receives a fair share**
  - **How can this be shown?**

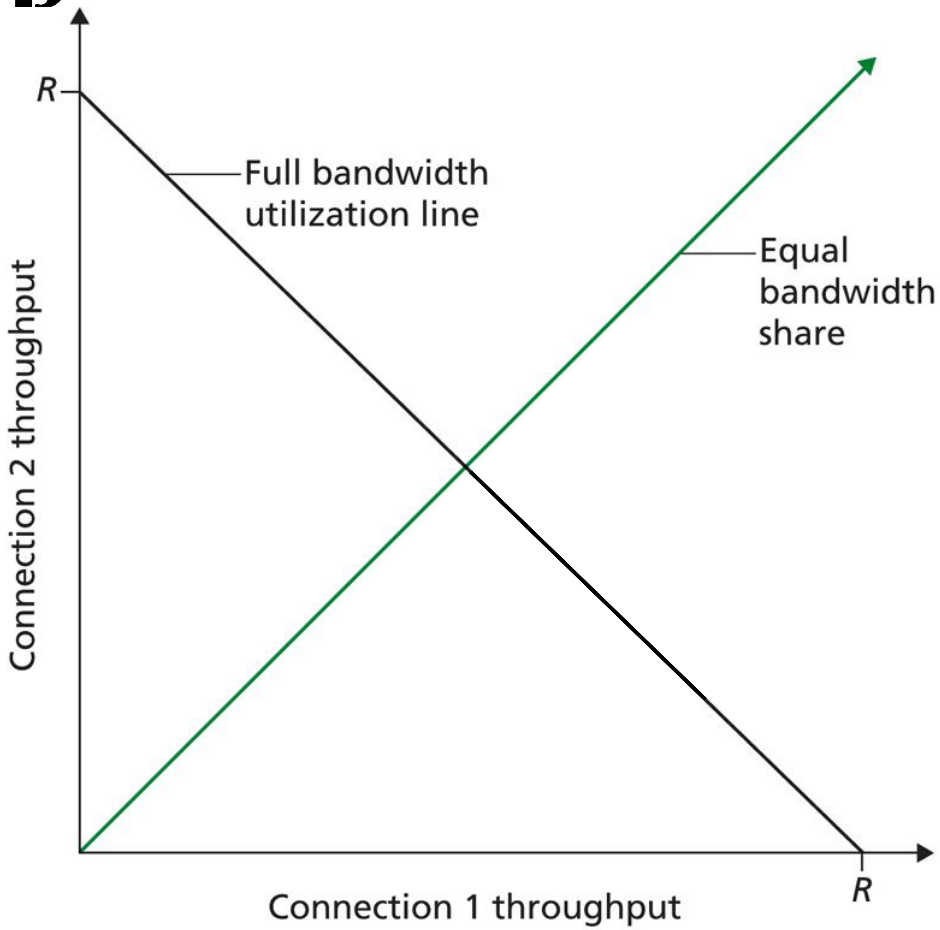
Link Sharing with TCP

- Illustration of two connections



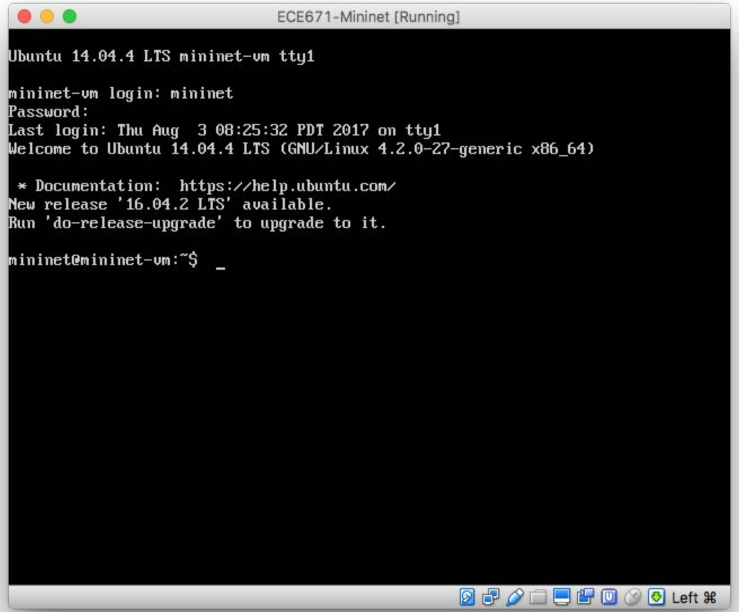
Link Sharing with TCP

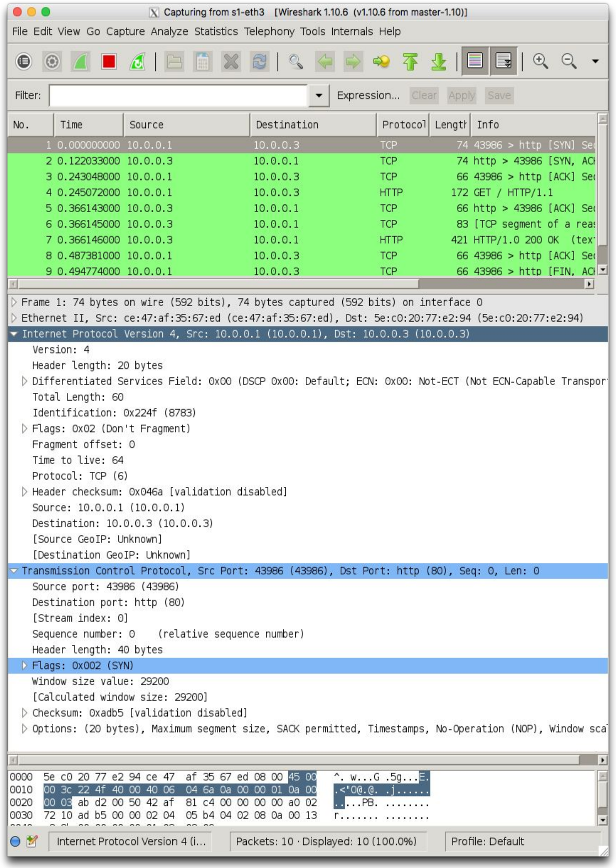
- Illustration of two connections



Connected Device: HTTP & TCP Demo

- Mininet demonstration in virtual machine





Summary of Lesson

- Connection setup and teardown
- Reasons for congestion
- Congestion control in TCP
- Fairness between two connections
- Demo

Post-work for Lesson 5

<u>Homework #3</u>

- **After the Live Lecture, you will complete and submit a homework assignment. Go to the online classroom to view and submit the assignment.**

To Prepare for the Next Lesson

- Complete and submit the Post-work for Lesson 5.
- Read the Required Readings for Lesson 6.
- Complete the Pre-work for Lesson 6.

Go to the online classroom for details.