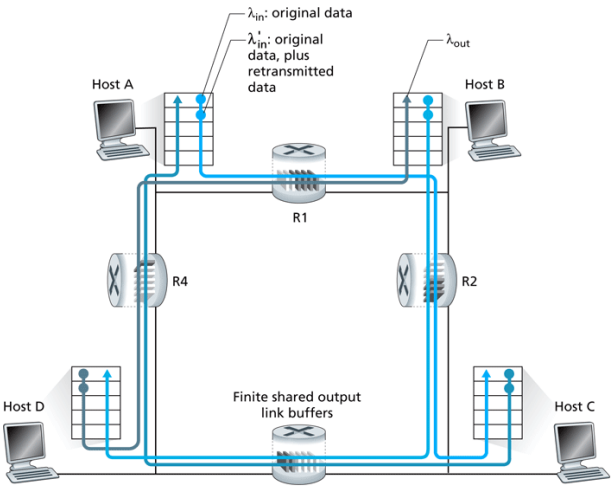
University of Massachusetts Amherst

ECE697AA – Lecture 10

Congestion Control: AIMD

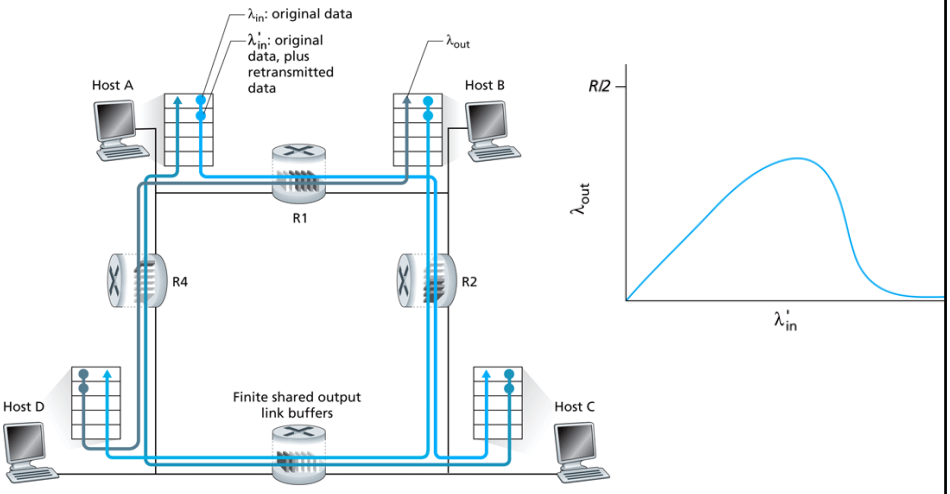
Tilman Wolf
Department of Electrical and Computer Engineering
10/07/08

What happens?



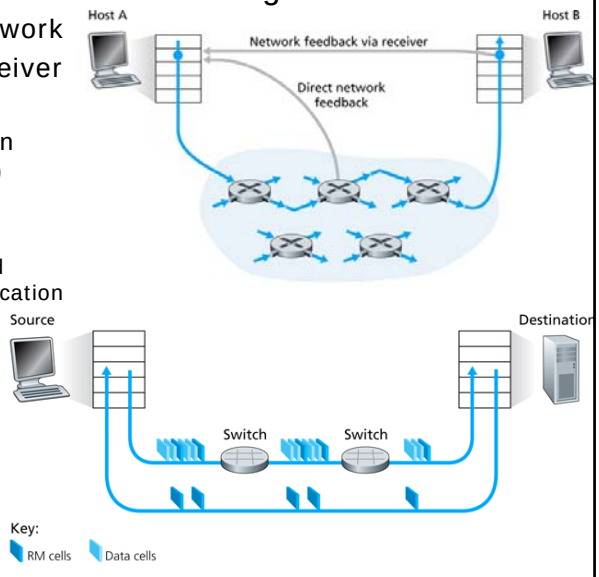
Network congestion

- Scenario with 4 senders



Explicit congestion control

- Sender needs to be notified of congestion
  - Feedback from network
  - Feedback from receiver
  - Examples
    - » Explicit congestion notification (ECN)
    - » Explicit resource management
      - Explicit forward congestion indication
      - No increase
      - Explicit rate

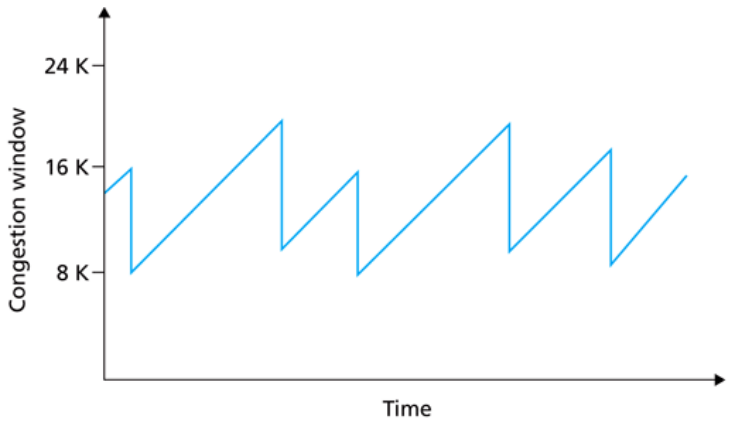


Implicit congestion control

- Sender observes network “from the outside”
  - Observation determines perceived congestion
  - TCP uses implicit congestion control
- Congestion indicators?
  - Packet loss
- Reaction to congestion
  - No congestion: increase rate
  - Congestion: decrease rate
- How is rate controlled in TCP?
  - Congestion window size
  - Rate approximately congestion window size / RTT

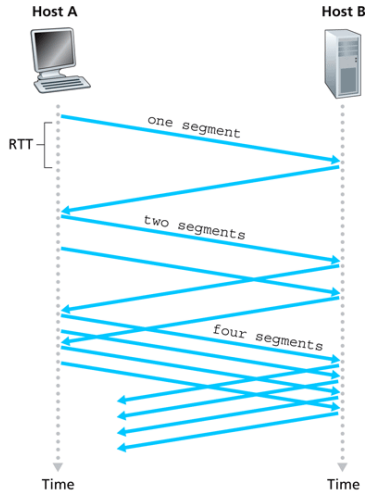
TCP congestion control

- How should connection react to (no) congestion?
  - Increase slowly
  - Decrease fast
  - “additive increase, multiplicative decrease” (AIMD)



TCP congestion control

- Startup
  - Increase by 1 MSS per RTT is very slow
    - » Target bandwidth unknown
  - "Slow start"
    - » Increase 1 MSS every time packet is acknowledged
  - Exponential bandwidth increase until loss
- How is loss determined?
  - Triple duplicate ACK
  - Timeout
- Reaction to timeout
  - TCP goes back to slow start
  - Exponential growth until half of window during timeout
    - » "Threshold" variable determines border between slow start and congestion avoidance



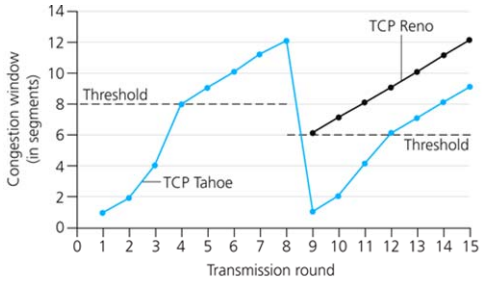
TCP congestion control

- Summary of behavior

State	Event	TCP Sender Congestion-control Action	Commentary
Slow Start (SS)	ACK receipt for previously unacknowledged data	CongWin = CongWin + MSS, If (CongWin > Threshold) set state to "Congestion Avoidance"	Resulting in a doubling of CongWin every RTT.
Congestion Avoidance (CA)	ACK receipt for previously unacknowledged data	CongWin = CongWin + MSS · (MSS/CongWin)	Additive increase, resulting in increase of CongWin by 1 MSS every RTT.
SS or CA	Loss event detected by triple duplicate ACK	Threshold = CongWin/2, CongWin = Threshold, set state to "Congestion Avoidance"	Fast recovery, implementing multiplicative decrease. CongWin will not drop below 1 MSS.
SS or CA	Timeout	Threshold = CongWin/2, CongWin = 1 MSS, set state to "Slow Start"	Enter slow start.
SS or CA	Duplicate ACK	Increment duplicate ACK count for segment being acknowledged	CongWin and Threshold not changed.

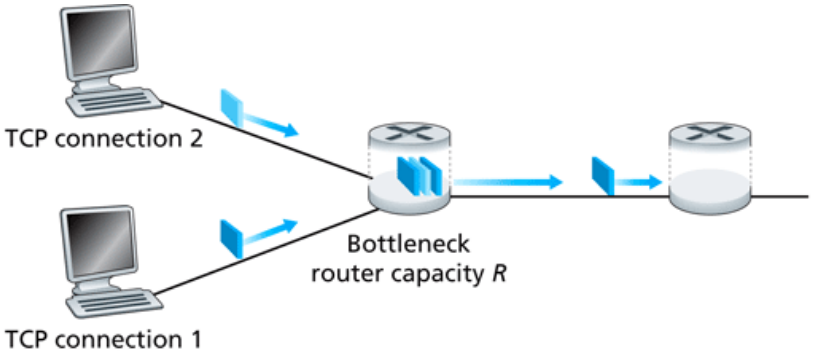
TCP congestion behavior

- TCP flavors
  - TCP Tahoe
    - » Unconditional reduction of window to 1 MSS under loss
  - TCP Reno
    - » Reduction of window to 1/2 when triple duplicate ACKS
    - » "Fast recovery"



Link sharing with TCP

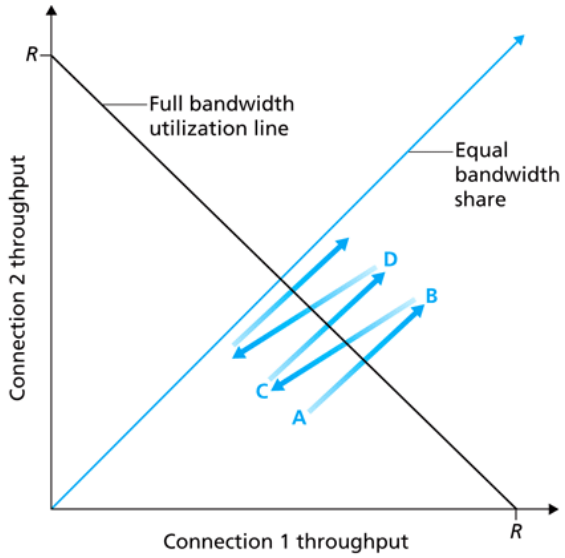
- Two TCP connections share link:



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

Link sharing with TCP

- Illustration of two connections



Assignments

- Read
  - Kurose & Ross: Chapters 3.6 & 3.7
- SPARK
  - Assessment quiz