CS120: Computer Networks

Lecture 18. Congestion Control 2

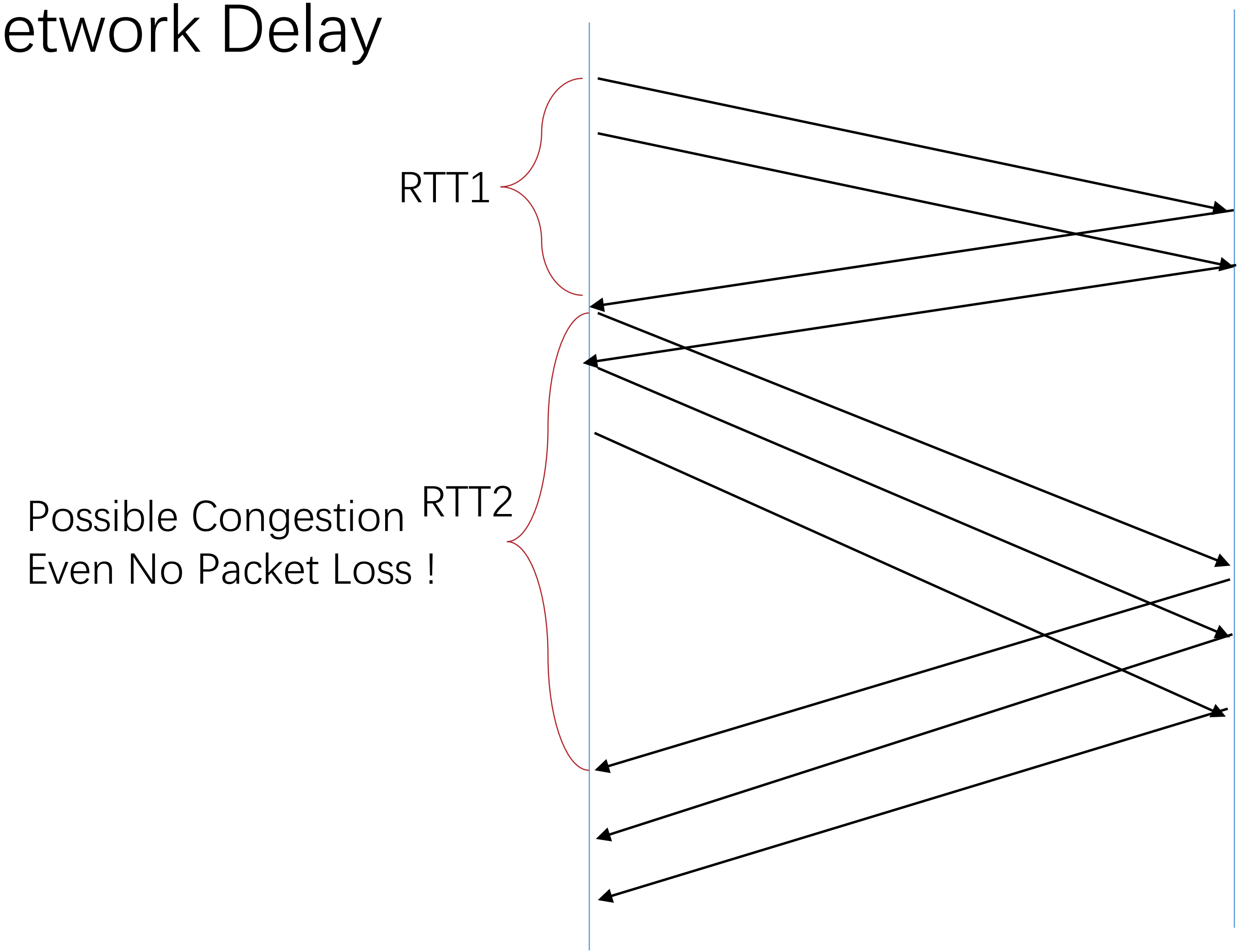
Zhice Yang

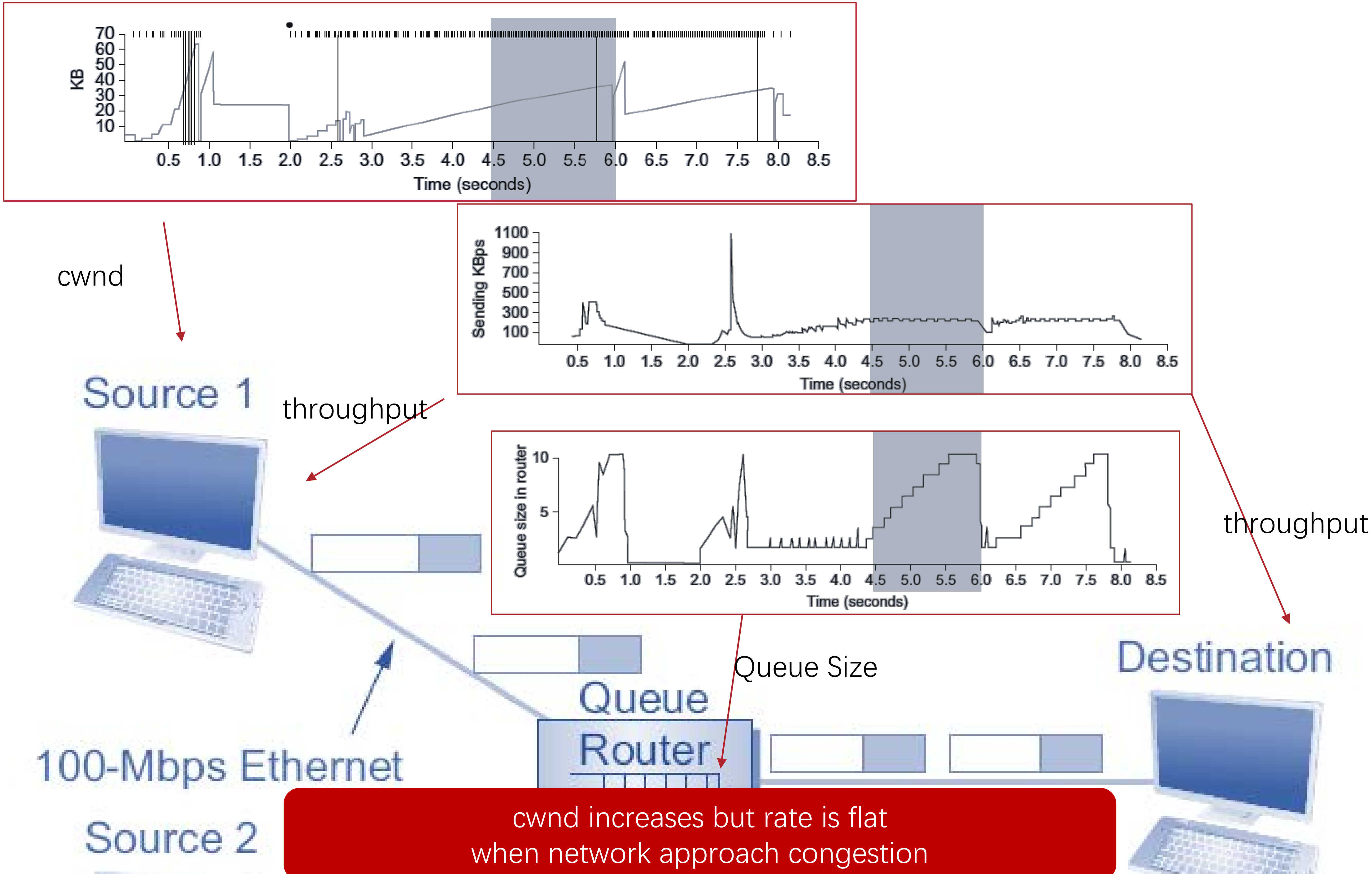
Congestion Control

- Host-based Congestion Control
  - Packet Loss
  - ➢Delay
- Router-based Congestion Control
  - Queuing Discipline

Avoid Congestion

- Feedback: Network Delay





cwnd increases but rate is flat
when network approach congestion

TCP Vegas

- Idea: TCP source uses RTT as the sign to avoid network congestion
  - Compare RTT with BaseRTT
- Method:
  - Reduce rate when congestion is about to happen
    - If Actual RTT >> Base RTT
  - Recover rate soon after bandwidth is available
    - if Actual RTT > Base RTT
    - Keep certain pressure on network

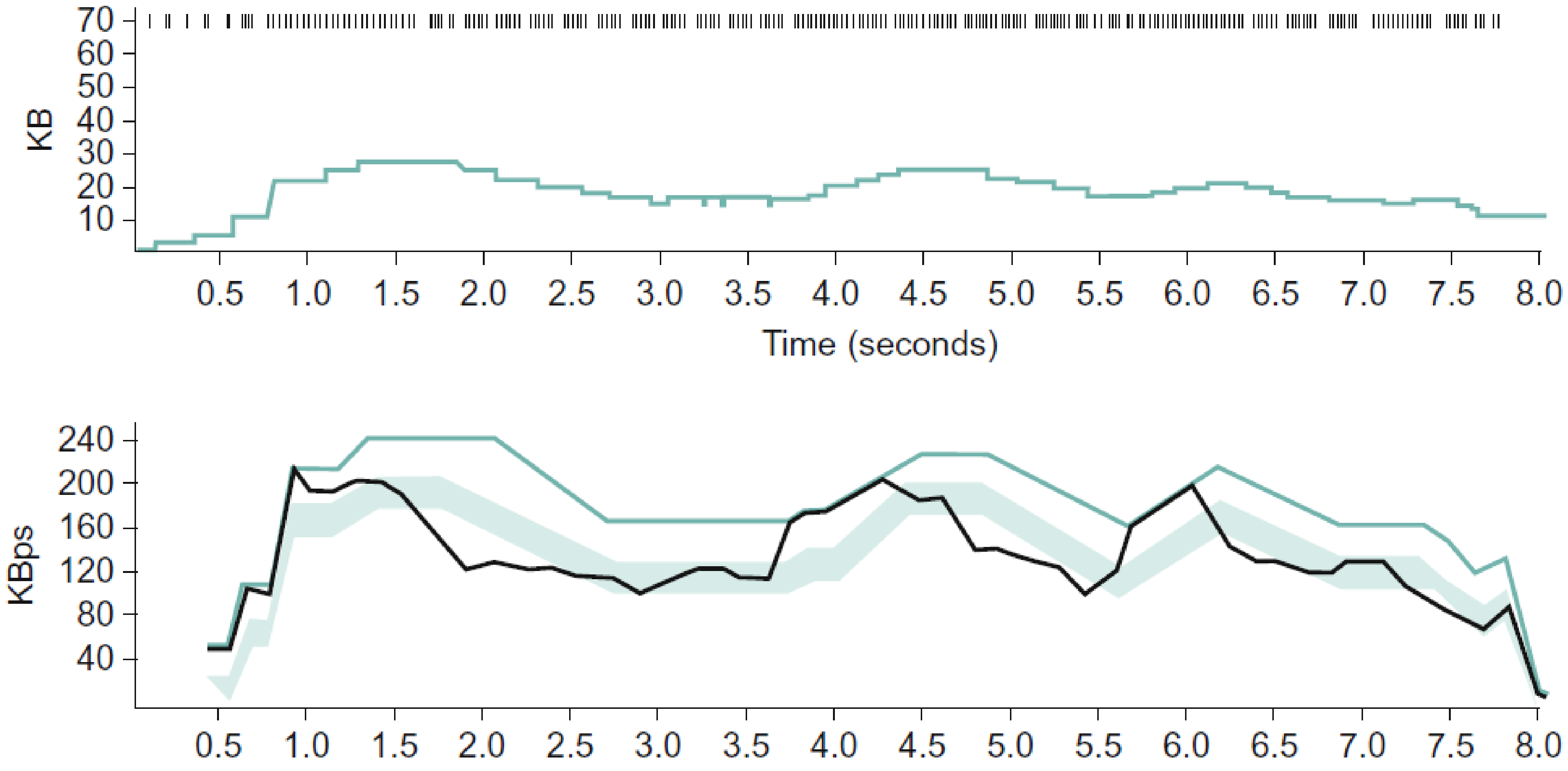
TCP Vegas Algorithm

- BaseRTT: the reference for increases in RTTs
  - The minimum RTT
    - If RTT < BaseRTT
      - BaseRTT = RTT
- ExpectRate:
  - CongestionWindow/BaseRTT
- ActualRate:
  - ActualRTT: according to timestamps
  - ActualRate=CongestionWindow/ActualRTT

TCP Vegas Algorithm

- Source compares ActualRate with ExpectRate
  - if ExpectRate - ActualRate $< \alpha$
    - cwnd++
  - if $\alpha <$ ExpectRate - ActualRate $< \beta$
    - cwnd = cwnd
  - if $\beta <$ ExpectRate - ActualRate
    - cwnd --

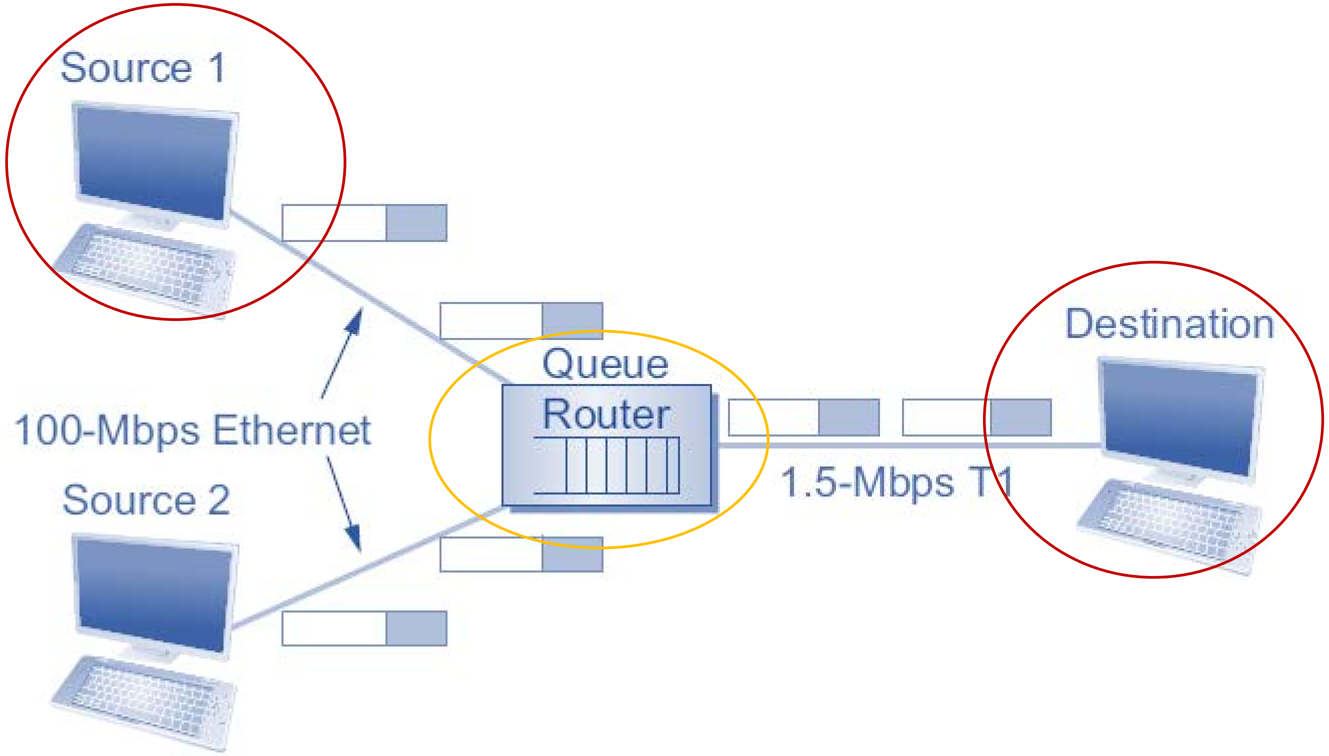
TCP Vegas



Top, congestion window; bottom, expected (colored line) and actual (black line) throughput. The shaded area is the region between the α and β threshold

Two Places to Handle Network Congestion

- End hosts
- ➢Routers



Congestion Control

- Host-based Congestion Control
  - Packet Loss
  - Delay
- Router-based Congestion Control
  - ➢Queuing Discipline

Queuing Discipline

- Why ?
  - Queuing discipline in routers determines how packets are transmitted (or dropped)
- Router Resource
  - Bandwidth
    - Which packets get transmitted
  - Queue Buffer
    - Which packets get discarded

Queuing Discipline

- First-In-First-Out (FIFO)

Congestion Control

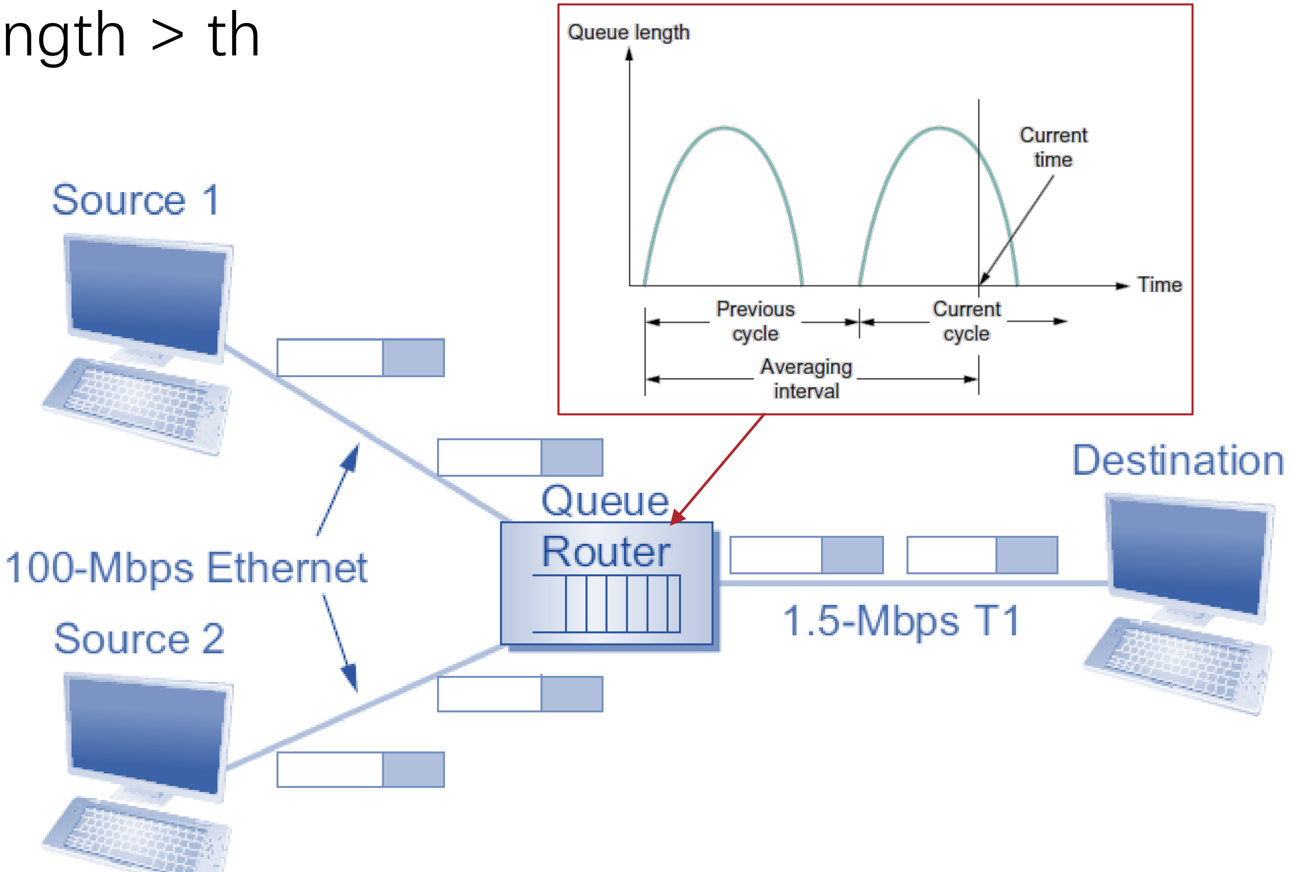
- Host-based Congestion Control
  - Packet Loss
  - Delay
- Router-based Congestion Control
  - Queuing
    - DECbit
    - Explicit Congestion Notification (ECN)
    - Random Early Detection (RED)

DECbit

- Developed for the Digital Network Architecture (DNA)
  - Before TCP/IP was “standardized”
- Idea: let routers explicitly indicate congestion
- Approach:
  - Router set congestion bit in passing packets if there is congestion
  - Destination echoes bit back to source through acks
  - Source adjusts cwnd according to congestion bit

DECbit

- Routers determine congestion
  - Monitor average queue length over last busy+idle cycle
  - If average queue length > th
    - set congestion bit

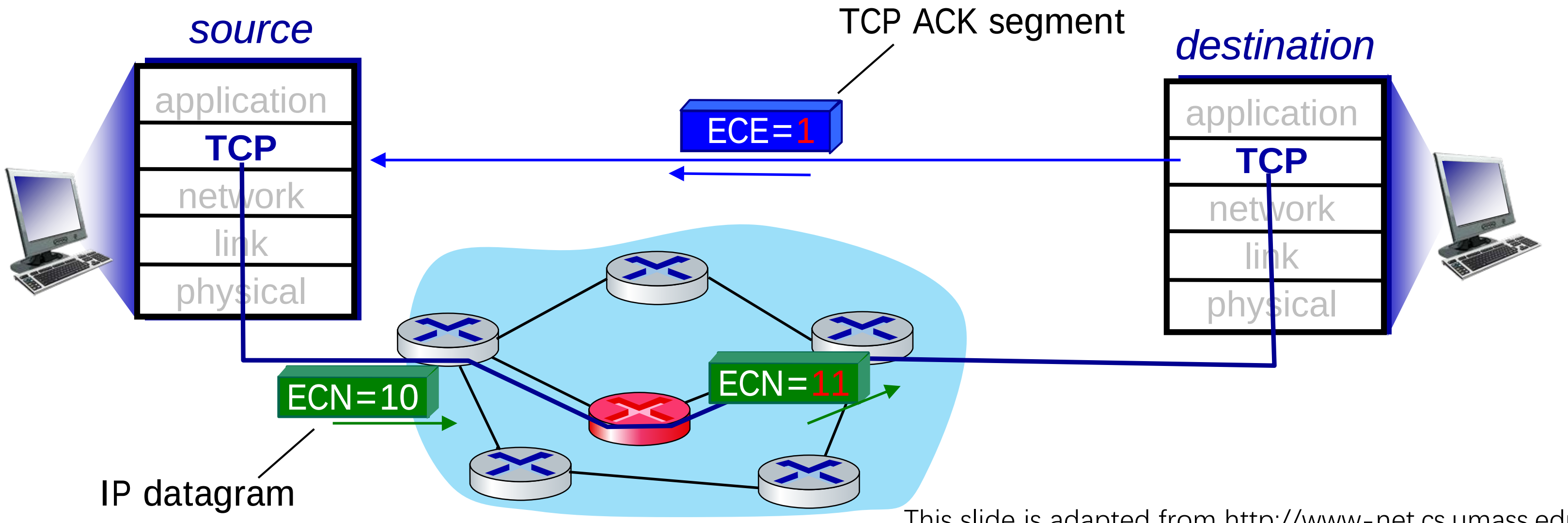


DECbit

- Source reacts to congestion bit
  - If < 50% of last window's packets had bit set
    - cwnd++
  - If > 50% of last window's packets had bit set
    - cwnd*0.875

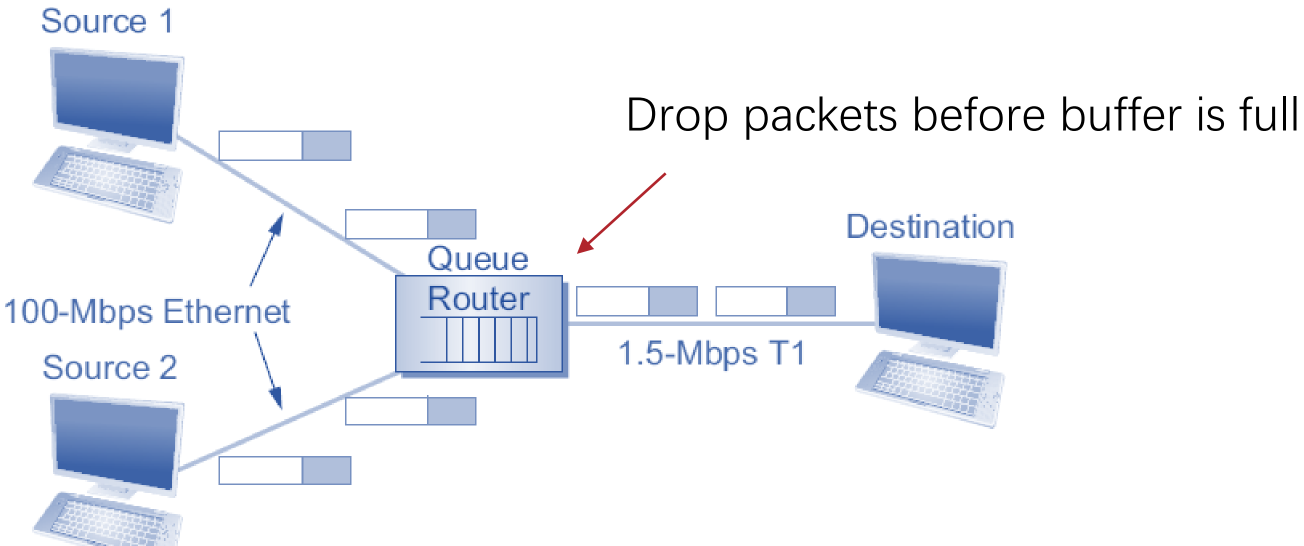
Explicit Congestion Notification (ECN)

- RFC 3168
  - Two bits in IP header (ECN in IP's ToS field) marked by network router to indicate congestion
  - Mechanism is similar to DECbit
  - Destination sets ECE bit (in TCP Header) on ACK segment to notify sender of congestion



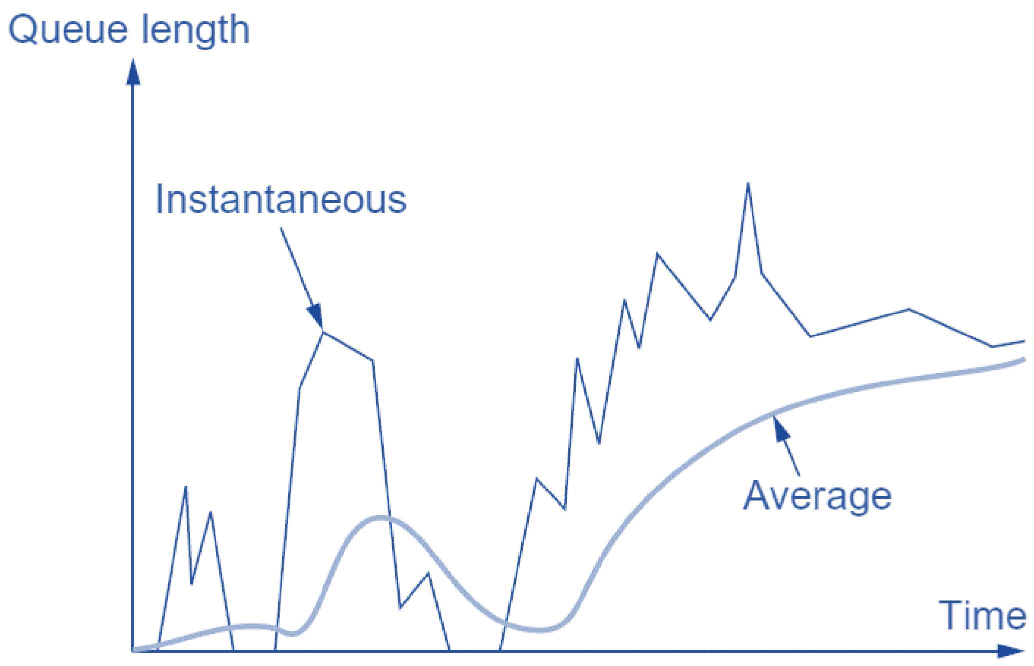
Random Early Detection (RED)

- Another way is to offload a part of congestion control to routers
- Idea: let routers implicitly indicate congestion
  - Router notices that the queue is getting backlogged
  - Router randomly drops packets to signal congestion
  - Source adjusts cwnd



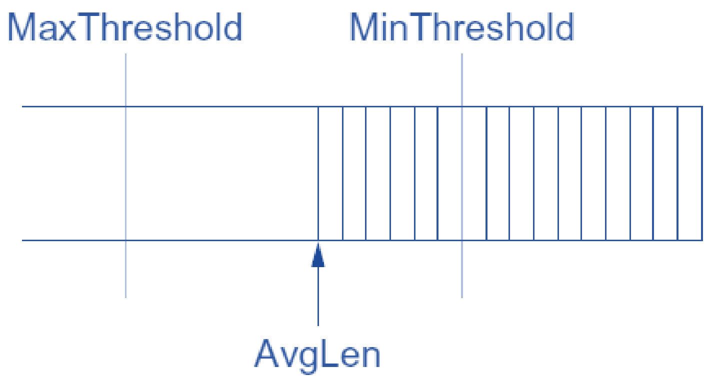
RED Algorithm

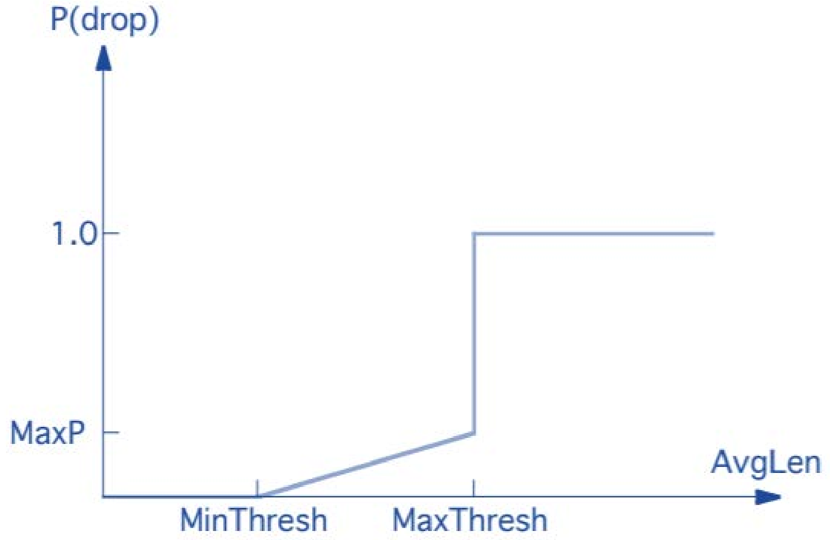
- Compute Average Queue Length
  - Moving average
    - AvgLen = (1 - Weight) * AvgLen + Weight * SampleLen



RED Algorithm

- Two queue length thresholds
  - if AvgLen <= MinThreshold
    - Enqueue the packet
  - if MinThreshold < AvgLen < MaxThreshold
    - Drop arriving packet with probability P
  - if MaxThreshold <= AvgLen then drop arriving packet





RED Algorithm

- Computing probability P
  - TempP = MaxP * (AvgLen - MinThreshold)/(MaxThreshold - MinThreshold)
  - P = TempP/(1 - count * TempP)
    - Count: number of continuously queued packets without drop
- Why?
  - Drops are spaced out in time
    - Count == 0 => P=TempP
    - Count == 1/TempP -1 => P=1
      - 1/TempP is expectation of number packets spaced between two drops

Reference

- Textbook 6.4