Computer Networks
COL 334/672

Congestion Control

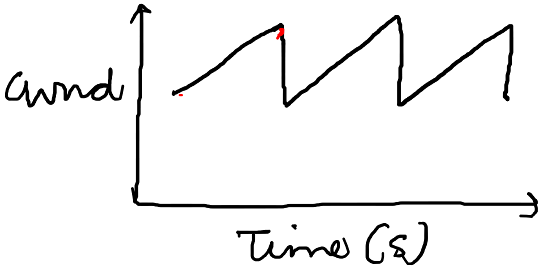
Tarun Mangla

Slides adapted from KR

Sem 1, 2024-25

Recap: TCP Congestion Control

- End-to-end congestion control algorithms (CCAs)
- Classic CCAs: TCP Reno, TCP Vegas
  - Additive Increase, Multiplicative Decrease (AIMD)
- Slow in case of “long, fat pipes” or networks with high bandwidth-delay product
- TCP CUBIC
  - Increase fast when further away from cwnd where last loss occurred
  - Increase slowly when around cwnd where last loss occurred

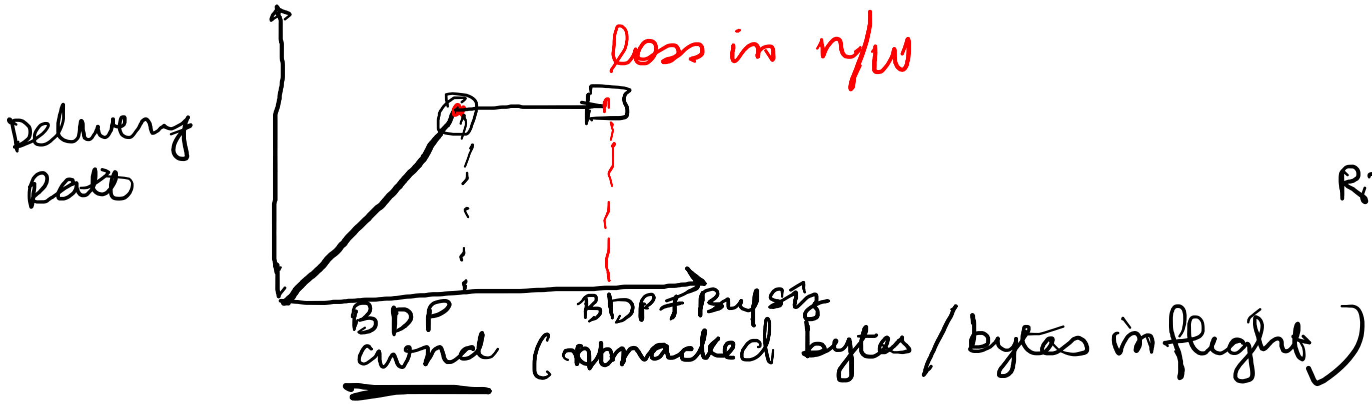


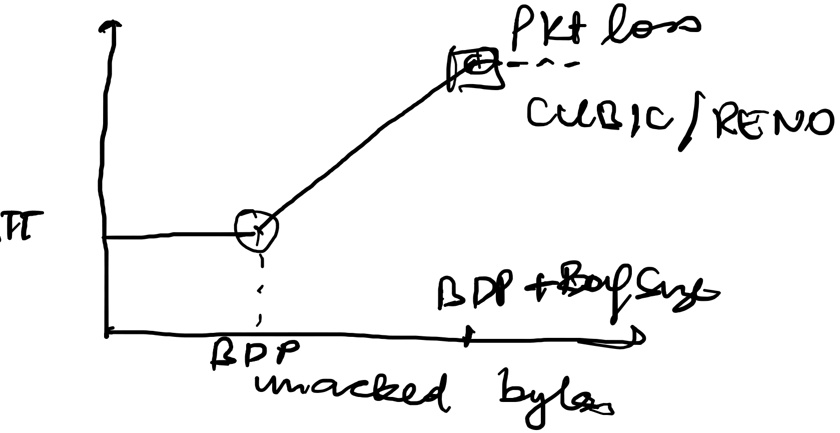
Limitations of a Loss-based CCA

→ Not all losses are due to congestion

→ ~~losses~~ occur after buffers are full [Bufferbloat] →

- Relying on loss to detect congestion is too reactive
  - Why?
- Waits for queues to build up in the router



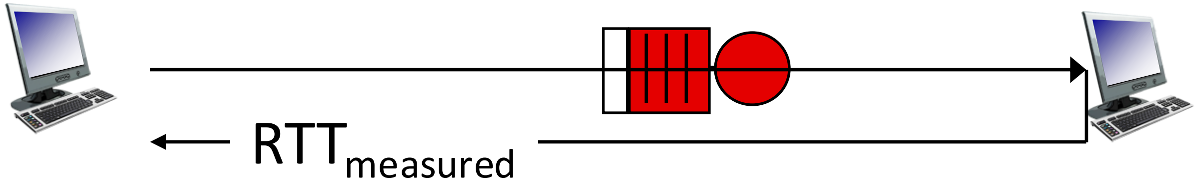


Can we think of another signal for detecting congestion?

Increase in RTTs

Delay-based TCP congestion control

Keeping sender-to-receiver pipe “just full enough, but no fuller”: keep bottleneck link busy transmitting, but avoid high delays/buffering



One Example – TCP Vegas

- RTT_{min} - minimum observed RTT (uncongested path)
- measured throughput: $\dfrac{\text{\# bytes sent in last RTT interval}}{RTT_{measured}}$
- uncongested throughput: $\dfrac{\text{\# bytes sent in last RTT interval}}{RTT_{min}}$

$|T_{measure} - T_{uncongested}| < \alpha$

Action : ↑ cwnd (linearly)

$|T_{measure} - T_{uncongested}| > \beta$

Action : ↓ cwnd (multiplicati

$\alpha \leq diff \leq \beta$ Action : Same cwnd

① RTT calculations can be unreliable

Challenge with Delay-based CCAs

TCP BBR → Google

(Bottleneck B/w
RTT)

- Don't interact well with loss-based CCAs

- What happens when a delay-based CCA competes with a loss-based CCA?

delay ~~bo~~ -based CCA ↓ cwnd

loss -based CCA wait for loss to happen

- Other limitations?

Network-assisted Congestion Control

- Routers in the network help in congestion control
- What are the possible approaches?
  - Tell end points about congestion Explicit Congestion Notification
  - Manage router buffer but let end-points figure Active queue management

Explicit congestion notification (ECN)

↦ Data center Networks

(DCTCP)

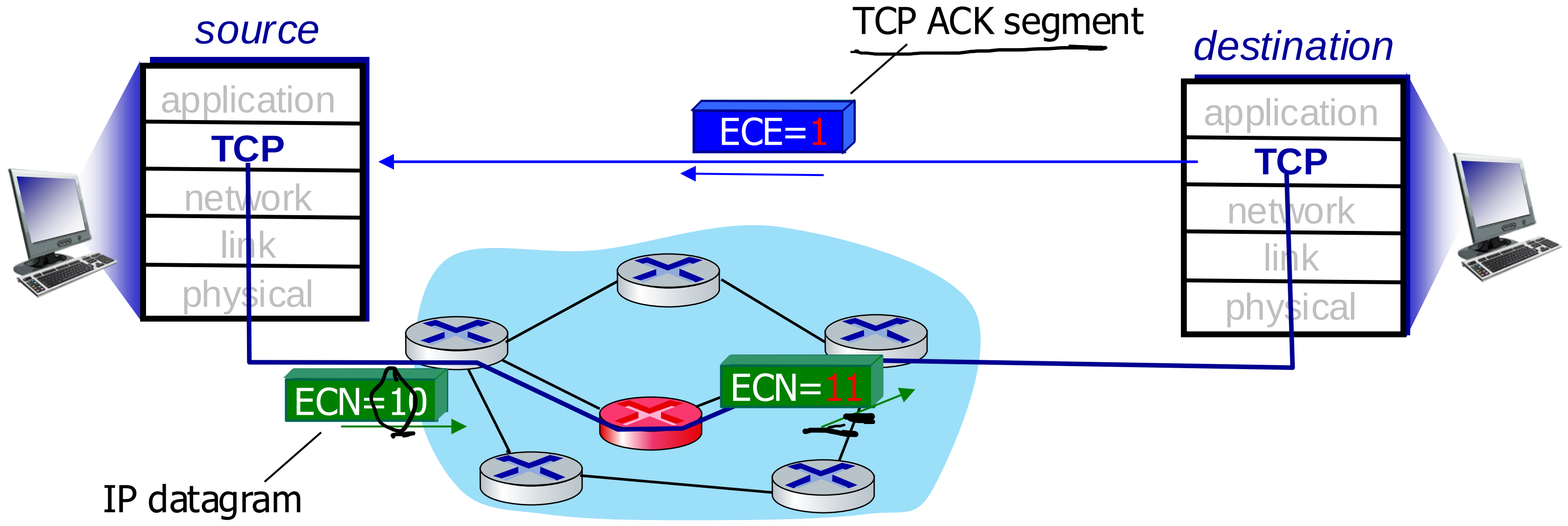
↓

ECN notification

8-bits → 2 bits

Use header in both Network and Transport Layer

- two bits in IP header (ToS field) marked *by network router* to indicate congestion
- congestion indication carried to destination
- destination sets ECE bit on ACK segment to notify sender of congestion
- sender reduces the congestion window on receiving an ACK with ECE bit set
- Limitation: Requires support from all router in the network path

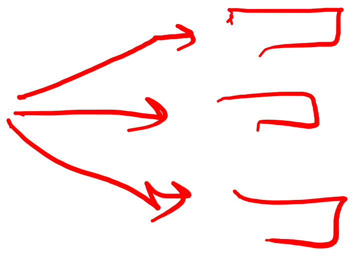


Active Queue Management

- Routers actively control the buffer queues to indirectly aid congestion control
- Why routers? Routers can most accurately identify queuing delays
- Any AQM techniques? Fair queuing, weighted fair queuing

Queuing Discipline

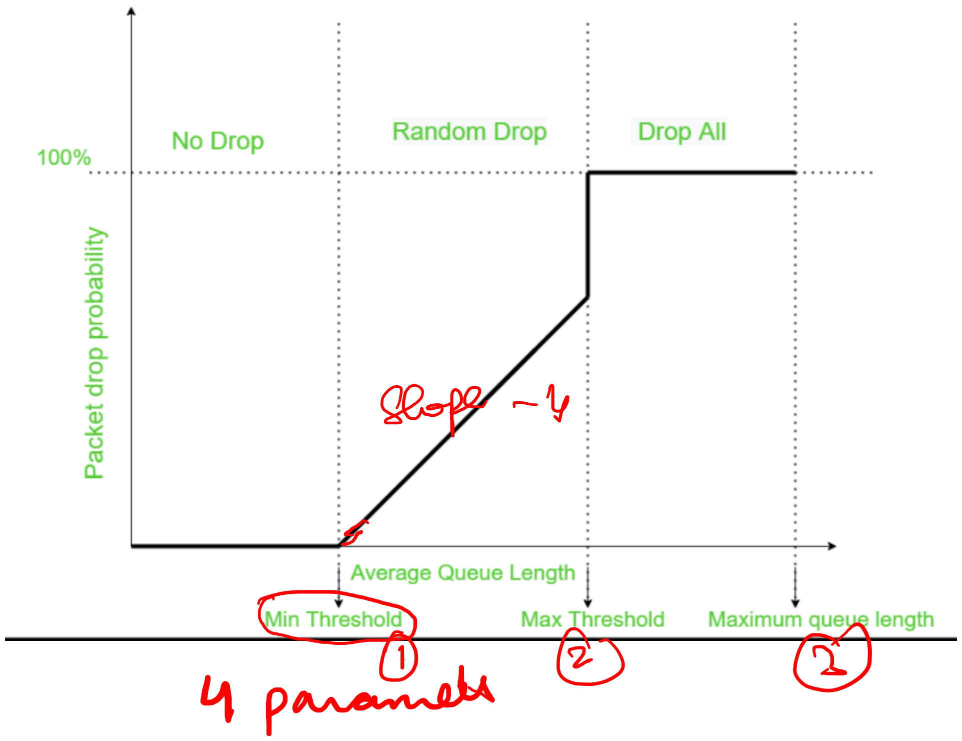
FQ



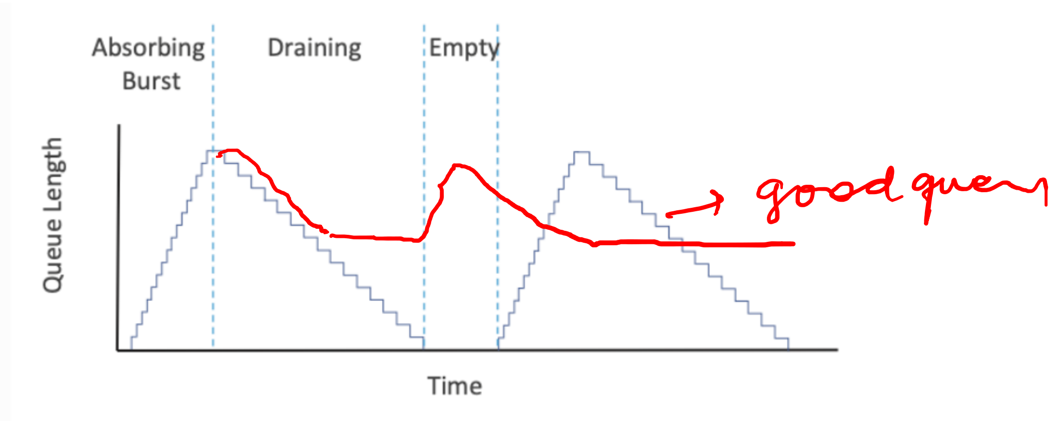
AQM Examples

Drop

- **Random Early Detection (RED)**



- **CoDel (Controlled Delay)**



- **Intuition**: CoDel largely ignore queues that last less than an RTT, but starts taking action as soon as a queue persists for more than an RTT

queuing delay > T_1
&
happens for T_2 times

Attendance