Computer Networks
COL 334/672

Congestion Control

Slides adapted from KR

Sem 1, 2025-26

Recap: TCP Congestion Control

TCP Reno

End-to-end congestion control algorithms

1. Additive Increase, Multiplicative Decrease

on reception of new Ack — packet loss (Timeout & Triple Dup Acks)

2. Fast-recovery mechanism

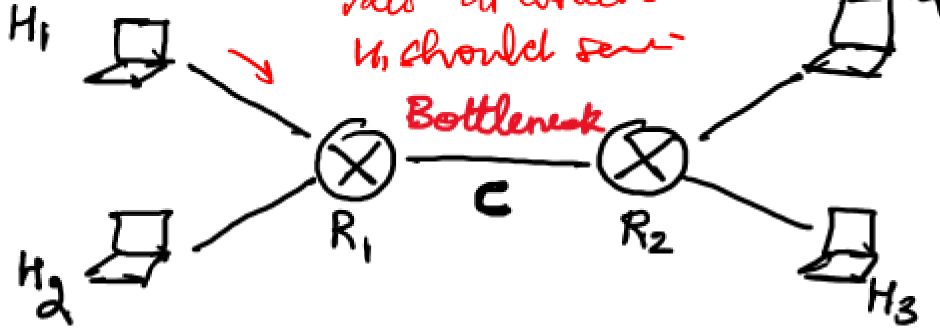
3. Slow-start phase

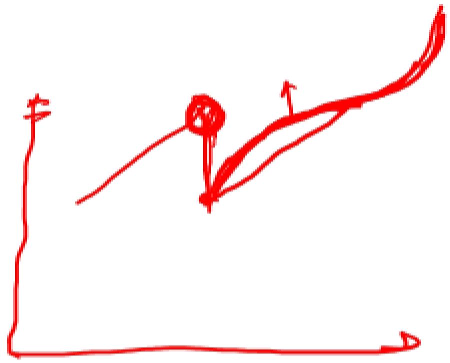
4. TCP CUBIC: congestion CA

(i). Increase was a function of elapsed time

(RTT fairness)

What is the rate at which H_1 should send

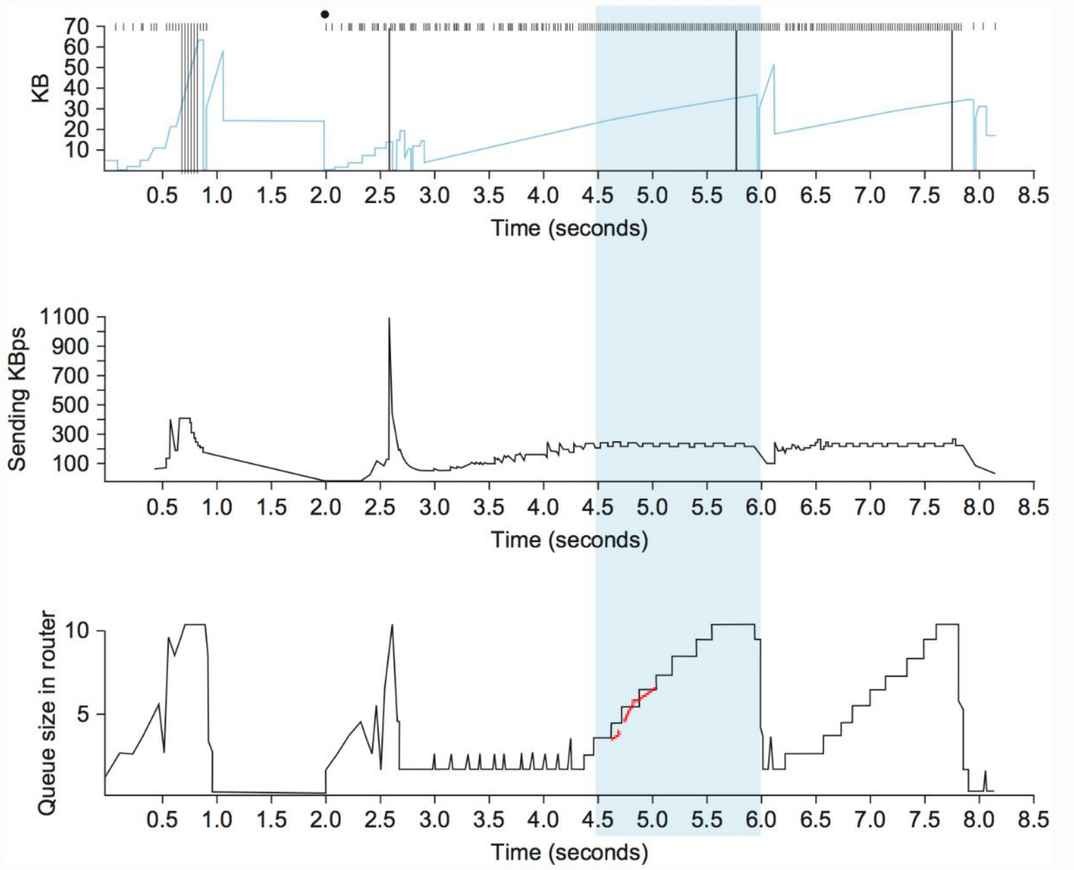




Limitations of a Loss-based CCA

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

cwnd



delays (or RTT)

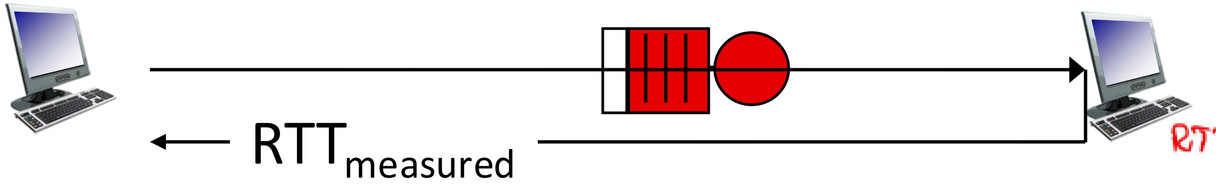
Can we think of another signal for detecting congestion?

BBR : Model-based CCA MD → New TCP CCA   Empirically decided

Delay-based TCP congestion control

What happens when VEGAS competes with the loss-based CCA?

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



constant

$RTT_{min} \times |\text{uncongested} - \text{measured tput}| > \alpha$ bps

CASE-1 congestion detected

↳ multiplicative decrease

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: $\dfrac{\text{\# bytes sent in last RTT interval}}{RTT_{min}}$

CASE-2

$\Delta = |\text{uncongested} - \text{measure tput}| \le \alpha$

↳ Additive increase

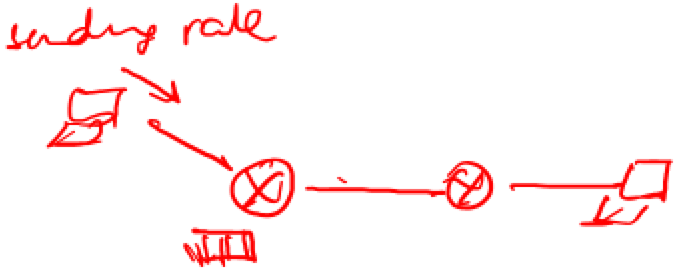
$\beta < \alpha$

β

$\alpha < \Delta < \beta$: do nothing

Hysteresis

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
  - Active queue management

send notification to end host that I am congested

Both are collaborating

DECbit

Data centers → DCTCP

short flows

Explicit congestion notification (ECN)

(1) When to set ECN bits?

(2) How to respond to multiple packets

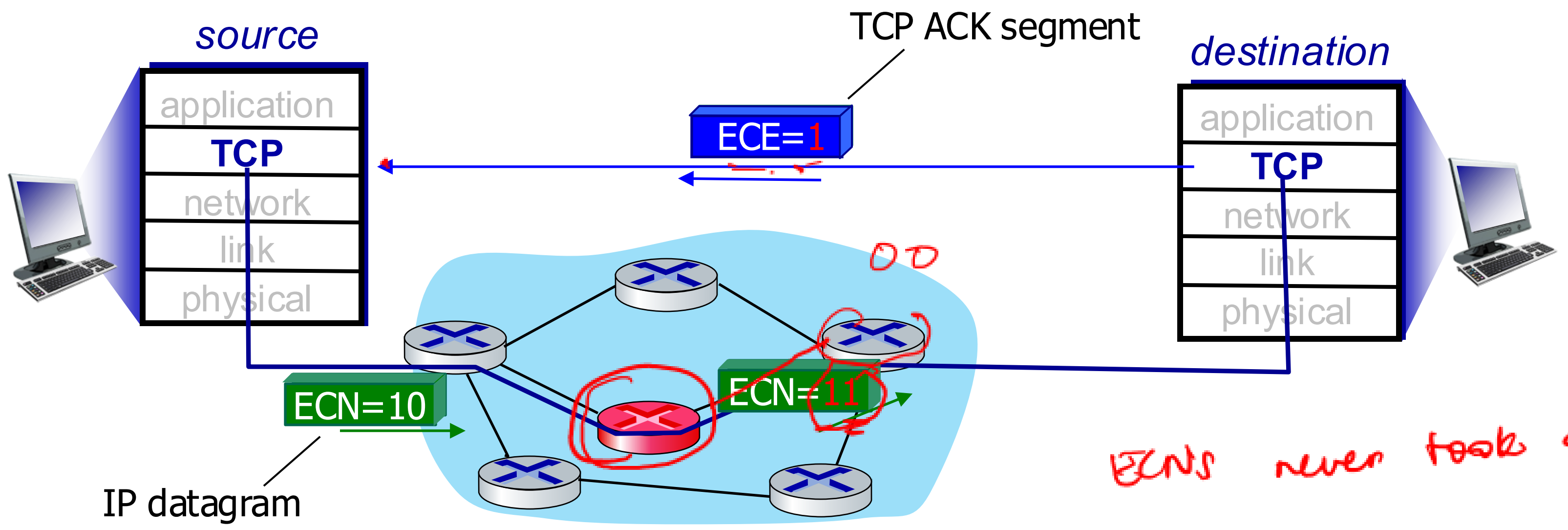
Use header in both Network and Transport Layer

Type of Service field

- 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

Reduce the sending rate

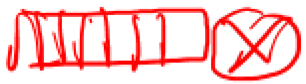
CWR



ECNs never took off

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
  - **Active queue management**



drops packets without notifying the end host → CCA at end host will react to dropped packet

Active Queue Management

- Routers actively control the buffer queues to indirectly aid congestion control
- Why routers? Routers can most accurately identify queuing delays

Random Early drop or Random Early detection

Key Idea: start dropping packets (probabilistically) as avg. queues lengths increases

[End hosts & Routers are working independently]

AQM Examples

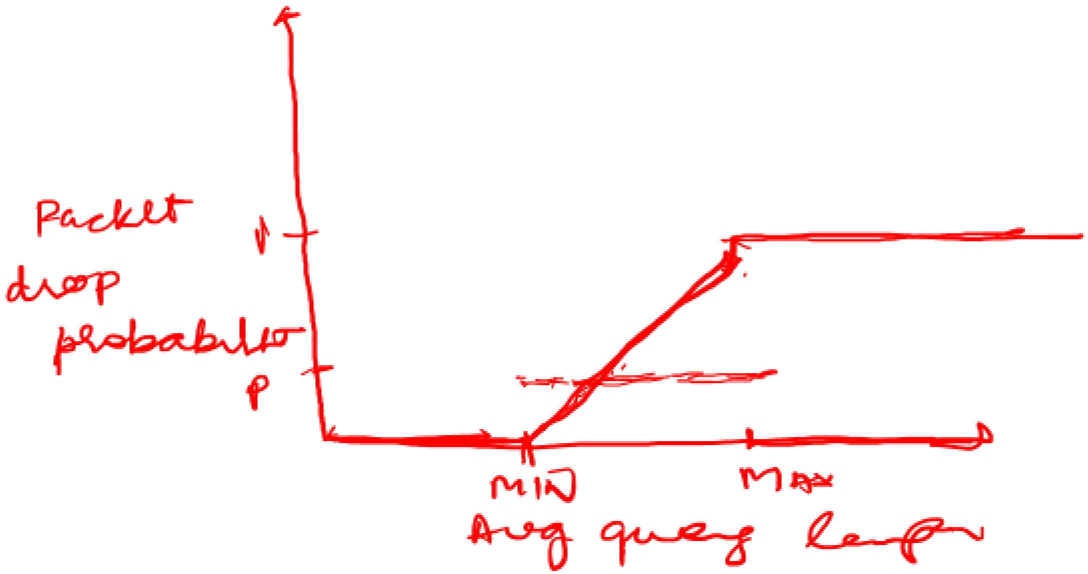
- **Random Early Detection (RED)**

Avg queue length > max
drop packets

p

Avg queue length < min
FORWARD

queue length ∈ (min, max)
drop packet : prob p



MIN

MAX

slope

CODEL

Fairness : larger flows will be penalized with a higher probability.

Evolving transport-layer functionality

- TCP, UDP: principal transport protocols for 40 years
- different "flavors" of TCP developed, for specific scenarios:

>4 papers out of 62 papers on TCP in Sigcomm24

Scenario	Challenges
Long, fat pipes (large data transfers)	Many packets "in flight"; loss shuts down pipeline
Wireless networks	Loss due to noisy wireless links, mobility; TCP treat this as congestion loss
Long-delay links	Extremely long RTTs
Data center networks	Latency sensitive
Background traffic flows	Low priority, "background" TCP flows

Sharing the network

Session Chair: Prateesh Goyal (Microsoft Research)

Keeping an Eye on Congestion Control in the Wild with Nebby Research Track

Ayush Mishra (National University of Singapore); Lakshay Rastogi (Indian Institute of Technology, Kanpur); Raj Joshi, Ben Leong (National University of Singapore)

SUSS: Improving TCP Performance by Speeding Up Slow-Start Research Track

Mahdi Arghavani, Haibo Zhang, David Eyers (School of Computing, University of Otago, New Zealand); Abbas Arghavani (School of Innovation, Design and Engineering, Mälardalen University, Sweden)

Principles for Internet Congestion Management Research Track

Lloyd Brown (UC Berkeley); Albert Gran Alcoz (ETH Zürich); Frank Cangialosi (BreezeML); Akshay Narayan (Brown University); Mohammad Alizadeh, Hari Balakrishnan (MIT); Eric Friedman (ICSI and UC Berkeley); Ethan Katz-Bassett (Columbia University); Arvind Krishnamurthy (University of Washington); Michael Schapira (Hebrew University of Jerusalem); Scott Shenker (ICSI AND UC Berkeley)

CCAnalyzer: An Efficient and Nearly-Passive Congestion Control Classifier Research Track

Ranysha Ware, Adithya Abraham Philip (Carnegie Mellon University); Nicholas Hungria (Carnegie Mellon Unversity); Yash Kothari, Justine Sherry, Srinivasan Seshan (Carnegie Mellon University)