Wide-Area Congestion Control

Lecture 19, Computer Networks (198:552)

Fall 2019

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Review: TCP congestion control

- Keep some in-flight (un-ACK'ed) packets: congestion window

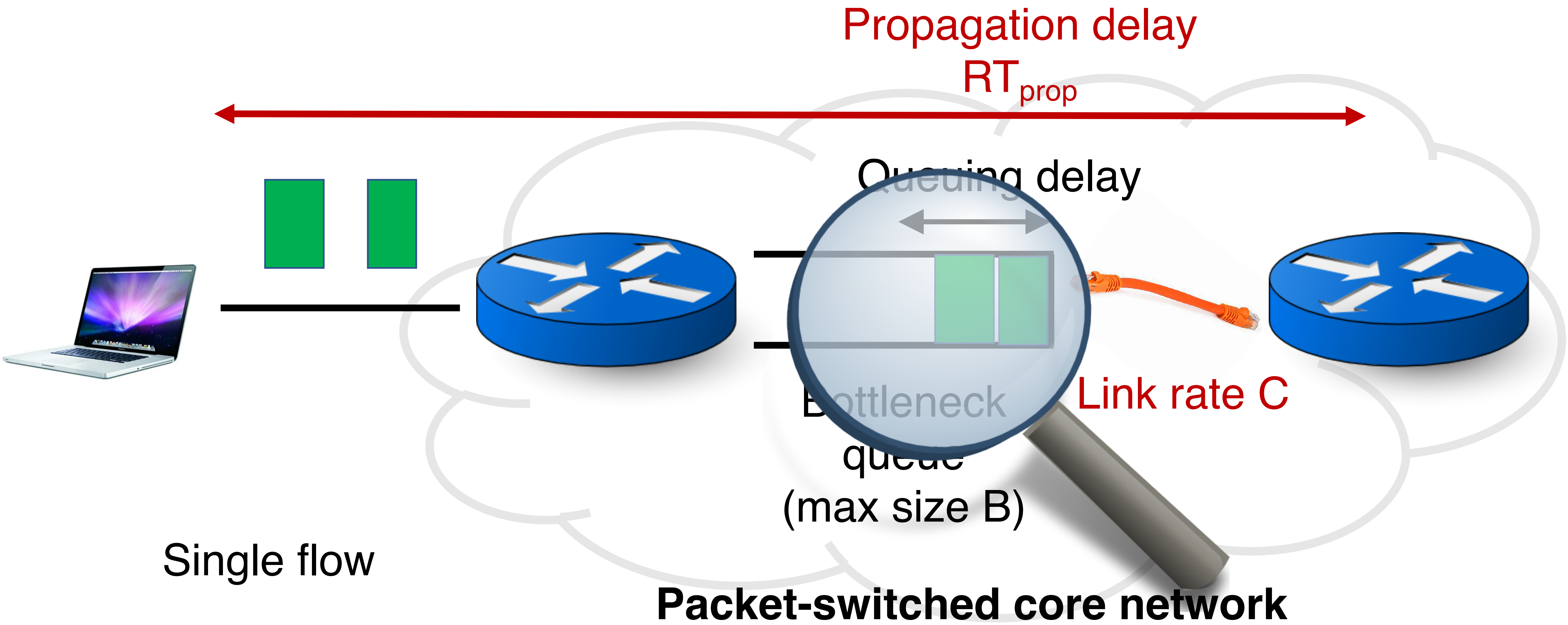
- Adjust window based on several algorithms:
    - Startup: slow start
    - Steady state: AIMD
    - Loss: fast retransmission, fast recovery

- Window versus rate-based protocols

Queue Dynamics with TCP

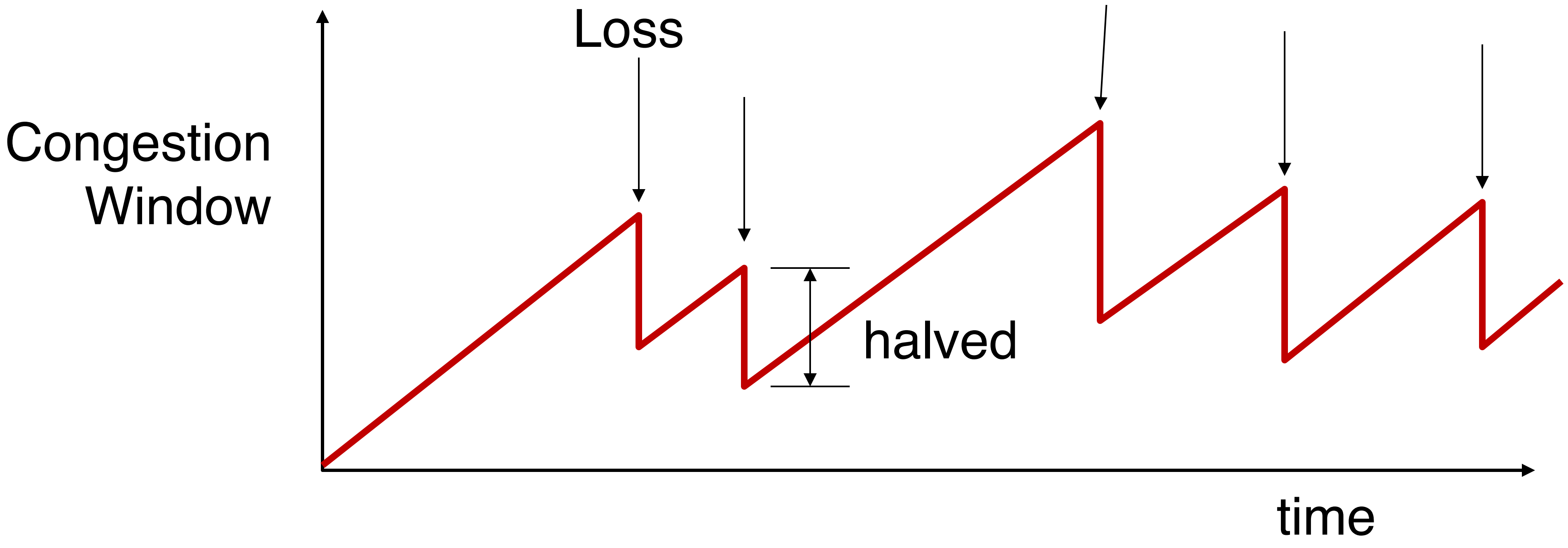
Steady-state behavior

Network model



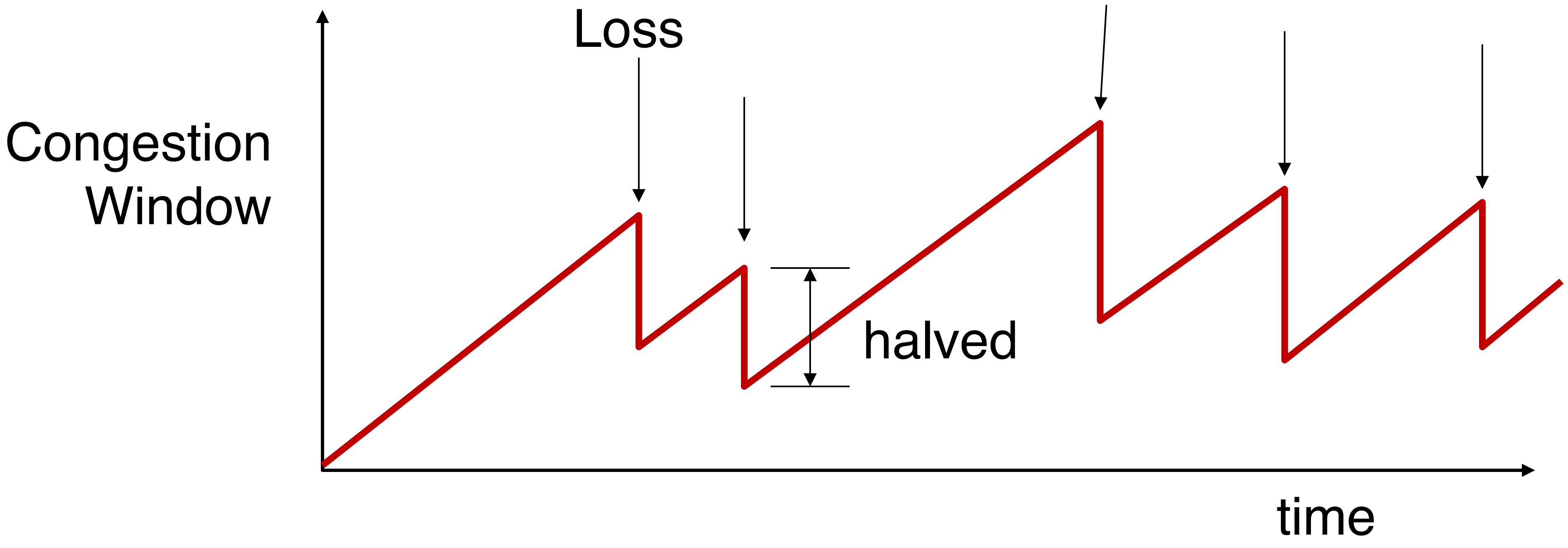
Sender behavior at steady state

- Congestion avoidance: Additive increase, multiplicative decrease (AIMD)
- Steady state isn’t static: lose pkts, grow cwnd, lose pkts, …

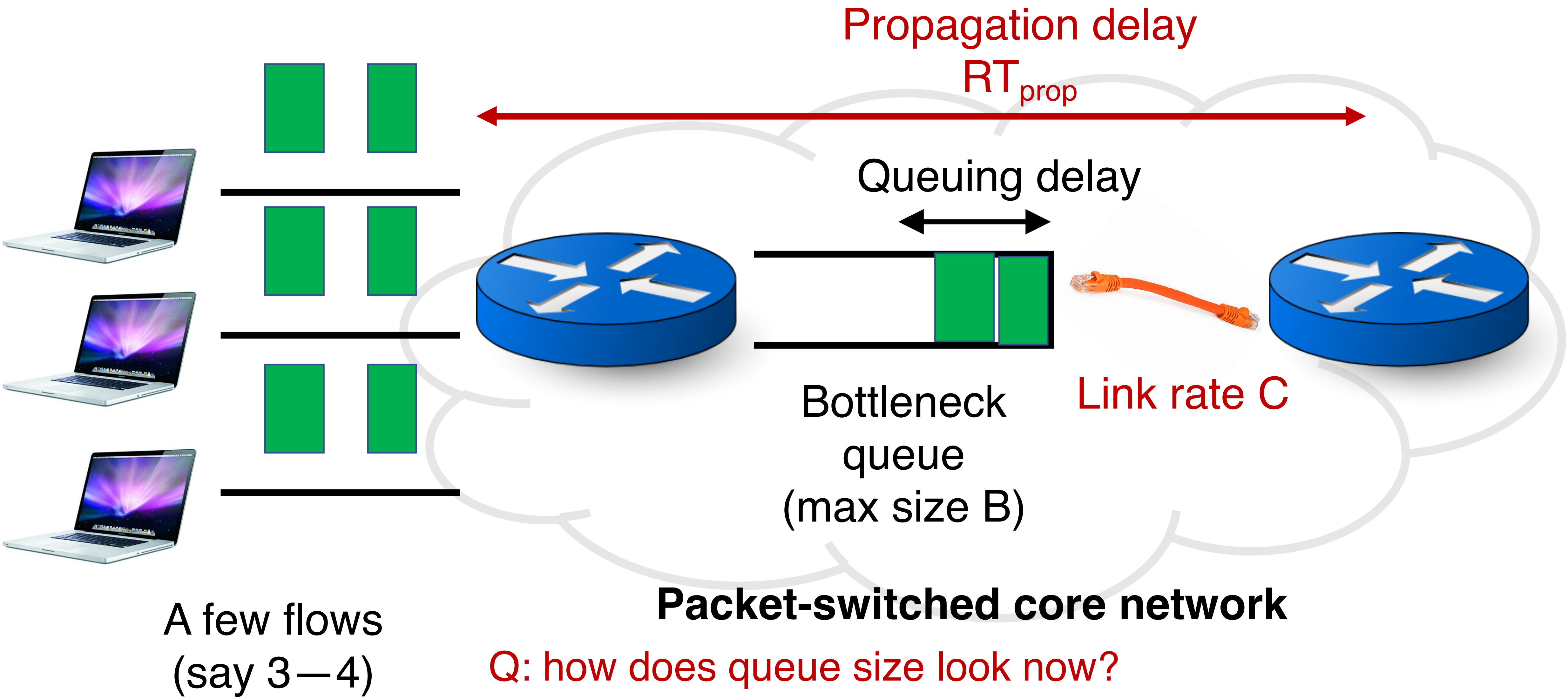


Sender behavior at steady state

- How does the queue size at the bottleneck look, over time?
  - Case 1: $B = C * RT_{prop}$
  - Case 2: $B > C * RT_{prop}$
  - Case 3: B < C * RT_prop

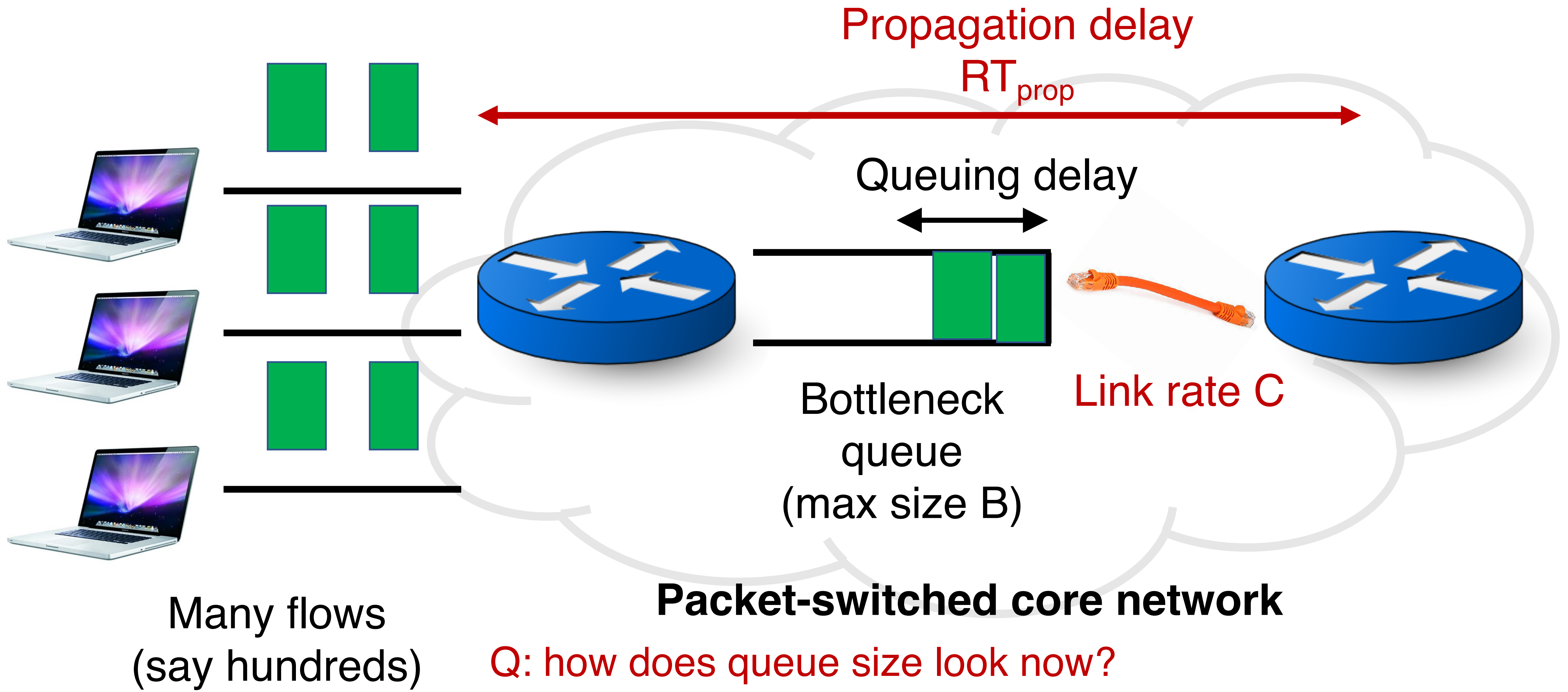


Network model



Q: how does queue size look now?

Network model



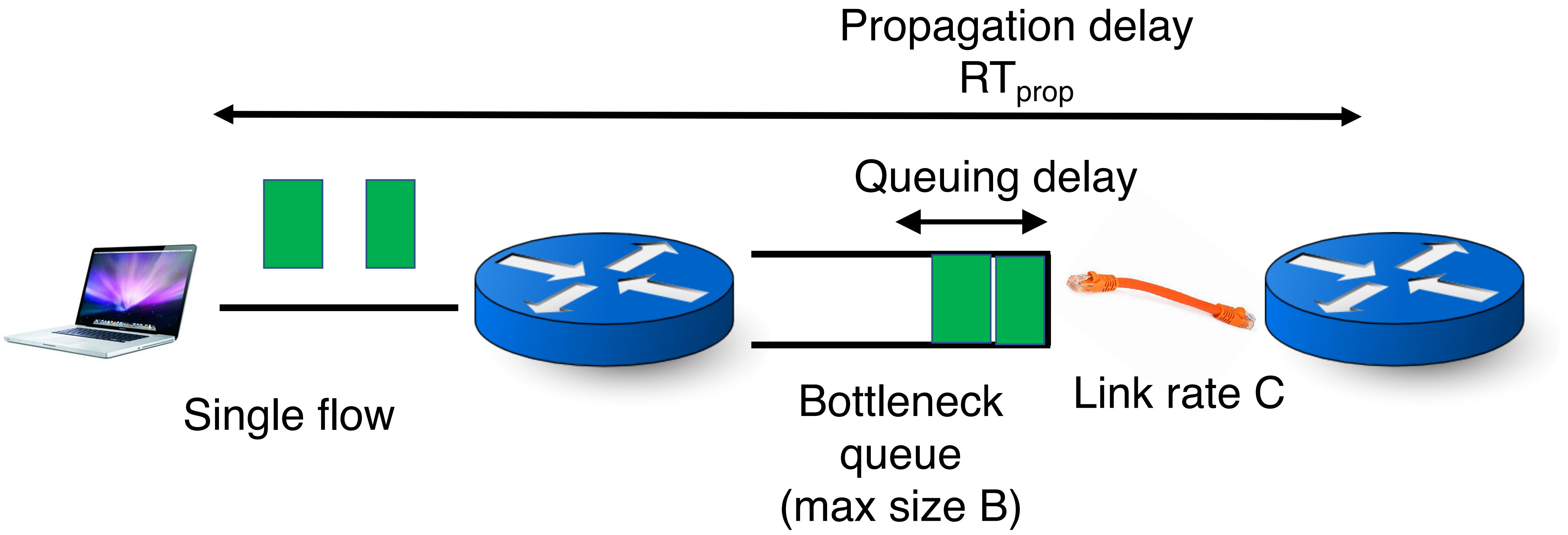
Q: how does queue size look now?

How big should router buffers be?

- Classic buffer-sizing rule: $B = C * RT_{prop}$
  - BDP buffer
  - Single TCP flow halving its window still gets a throughput of 100% link rate
- Q: should buffers be BDP-sized?
- Significant implications:
  - Massive pkt buffers (e.g., 40 Gbit/s with 200ms RT_{prop}): high cost
  - Massive pkt delays: bufferbloat

TCP BBR

Key ideas



1. Estimate the bottleneck link rate C
2. Estimate the propagation delay RT_{prop}
3. Send at rate C with at most k * C * RT_{prop} packets in flight

Pros and Cons?

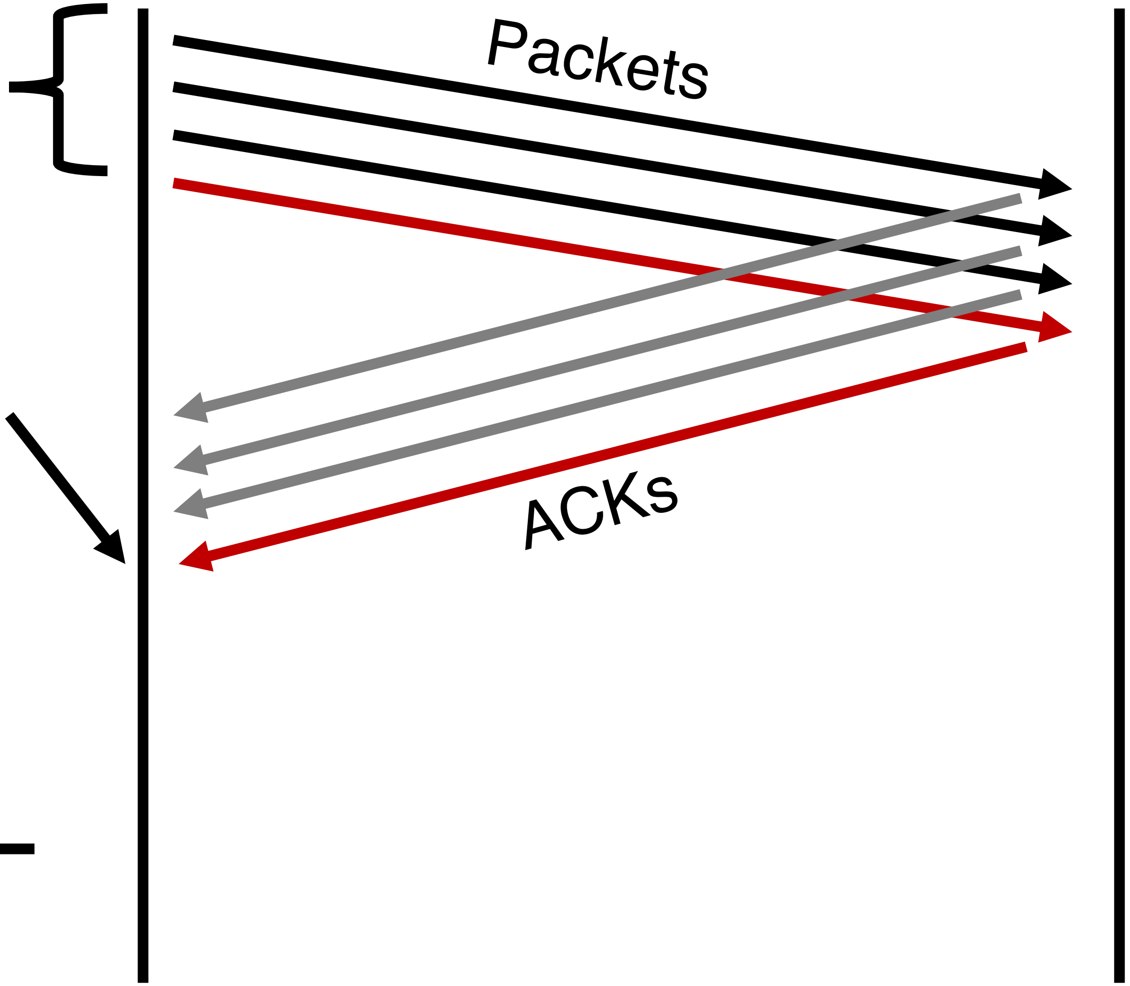
(1) Estimating the bottleneck link rate

- Data can't be delivered to a receiver faster than the bottleneck link rate
- Measure the data delivery rate
  - And use the maximum value over the recent past
  - Important: measurements time out after a certain period
  - Occasionally send higher (PROBE_BW cycling) to see if changed
- Q: how would you measure delivery rate at the receiver?
- Q: how would you measure delivery rate at the sender?

Measuring delivery rate at the sender

Data that is unACKed at the time of transmitting packet

Normal case: All that data (and only that data) is ACKed by this point



$$\frac{\text{unACKed data at pkt transmit time}}{\text{Round trip time between pkt-ACK}}$$

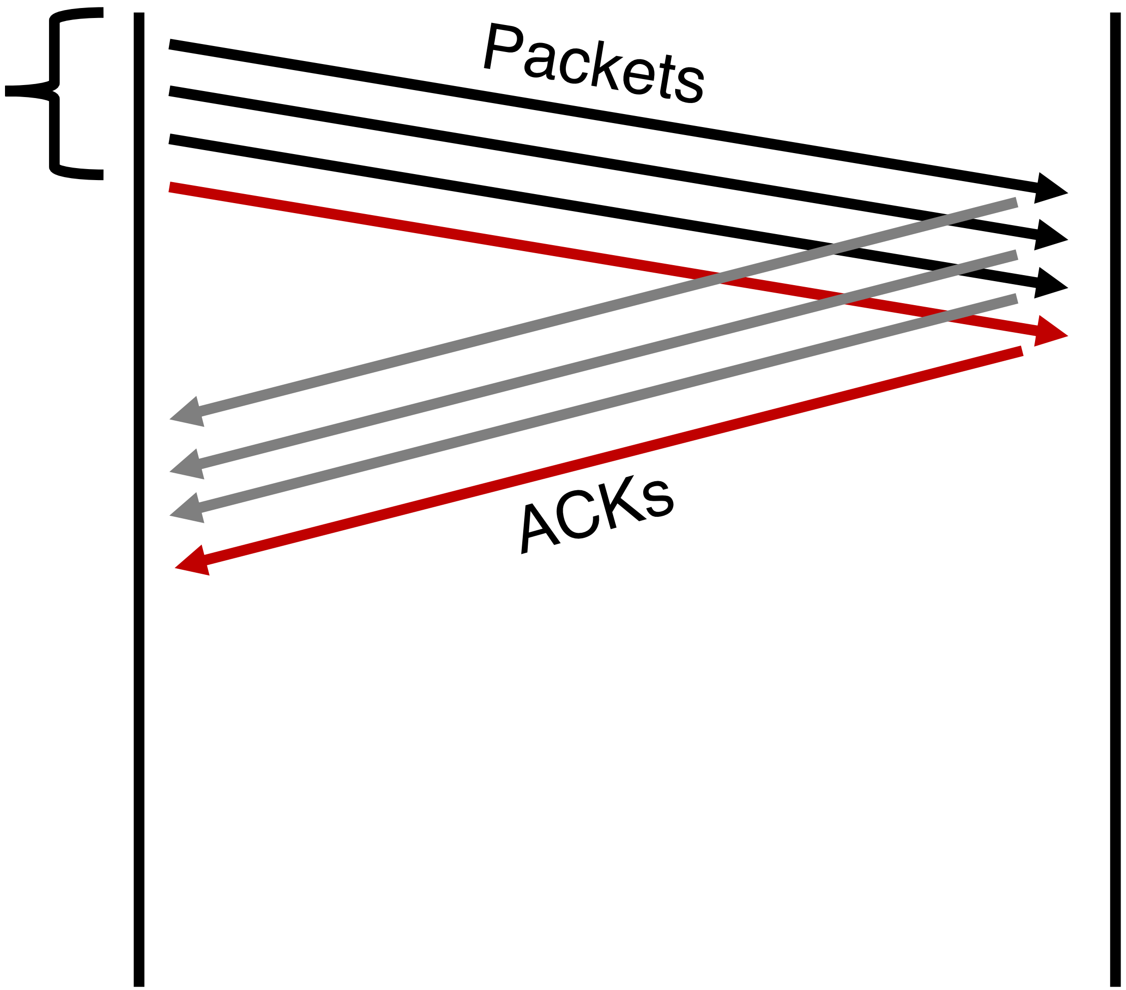
Quirk: Often, ACKs are “aggregated”

More data appears to be in flight than there actually is

Idea: use minimum of sent rate and received rate

Q: how would you measure the rate at which data was sent?

(Note: packets of received data and sent data must be the same)



(2) Estimating RT_{prop}

- Use the minimum of the RTT values experienced so far

- If you're sending at high rate, it is difficult to see the true RT_{prop} of the path
  - Q: why?

- Occasionally send just a few packets in an RTT to measure RT_{prop} (PROBE_RTT cycling)

- Also allows achieving fairness among BBR flows

Issues specific to wide-area

The Internet: Many things to consider…

- Bufferbloat
- Token-bucket policers
- Cellular base station scheduling
- Sometimes compete with few streams, sometimes many
- Delayed and aggregated ACKs (WiFi)
- Coexisting with legacy protocols (e.g., Cubic)