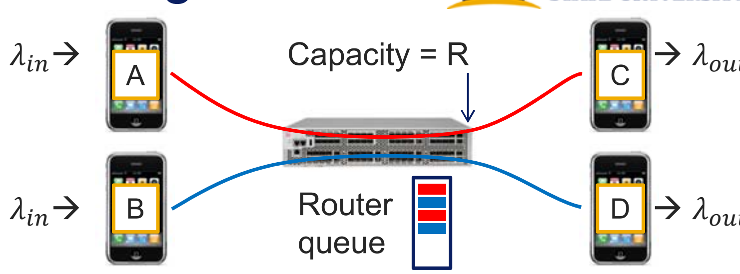
Congestion Control

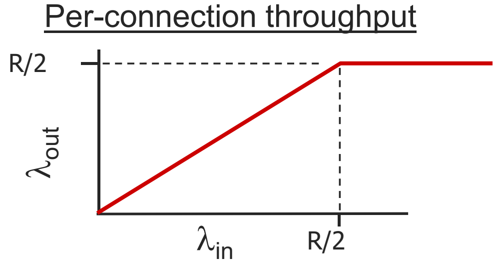
- Congestion:
  - Too many sources sending too much data too fast for network to handle
  - Consider the formula for traffic intensity: La/R
  - What are the effects of network congestion?
    - High queuing delay
    - Buffer overflows and packet loss
    - Retransmission and lost network resources

Idealized example of congestion

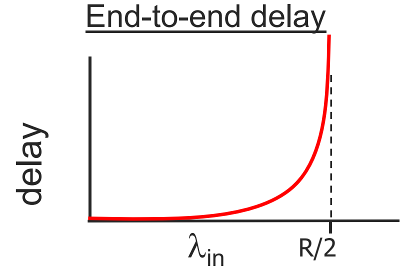
Scenario:
- Two senders, two receivers
- One router, infinite buffers
- Output link capacity: R
- Unlimited buffer space
- No retransmissions



Per-connection throughput



End-to-end delay

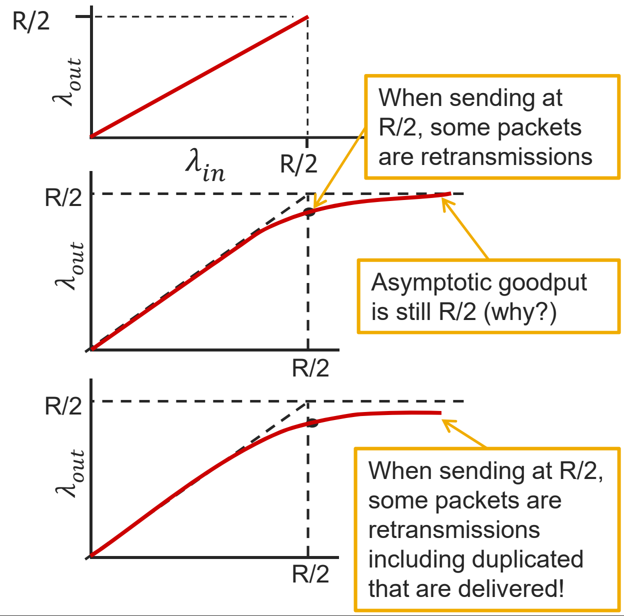


More realistic congestion model

MONTANA STATE UNIVERSITY

Draw per-connection throughput and end-to-end delay graphs when buffers are finite and:

1. Senders “magically” know when router has empty buffer space and send only when space available
2. Sender resends packets only when lost (not just delayed) due to full buffers
3. Sender retransmissions can result from premature timeout

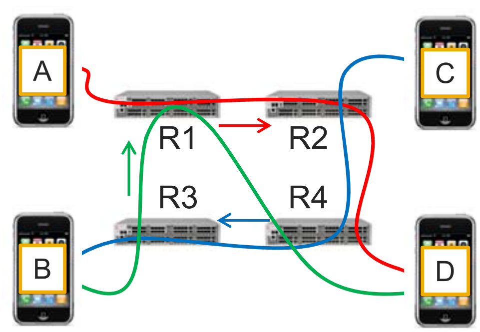


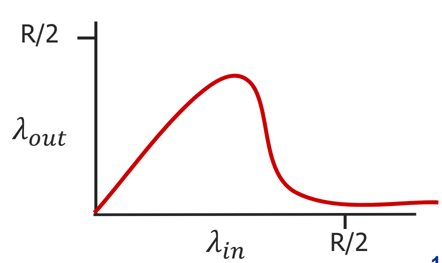
Mountains & Minds

More realistic congestion model

MONTANA STATE UNIVERSITY

- What happens to the red A-D flow as λ_{in} increases?
  - Red flow arrives at R2 with rate R
  - Blue flow arrives at rate λ_{in}> R
  - Red flow throughput drops to almost zero
- When packet dropped, any upstream transmission capacity used for that packet was wasted!
- What happens to the green B-D flow?
  - Red flow wastes resources at R1
  - Depending on the router switching fabric the green flow may suffer from congestion at R1 as λ_{in} increases





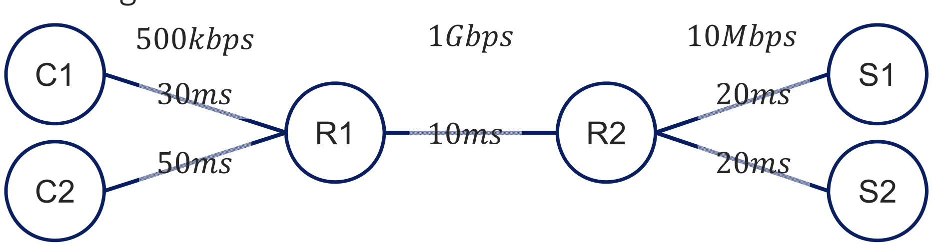
Mountains & Minds 148

Congestion control protocol

How to deliver the right (max) amount of data through the network? $\stackrel{?}{=}$ How to get the right (max) allowance from your parents?

Class Exercise

- Assume the following network, with finite buffers, and two clients downloading data from two servers



- How would you design a congestion control protocol for this network?
- Some things to consider: how is congestion detected, number of retransmissions, flow fairness, end-to-end delay, protocol complexity

Congestion Control

MONTANA STATE UNIVERSITY

End-to-end

- No explicit feedback from network
- Congestion inferred from end-system observed loss, or increasing packet delay

Network-assisted

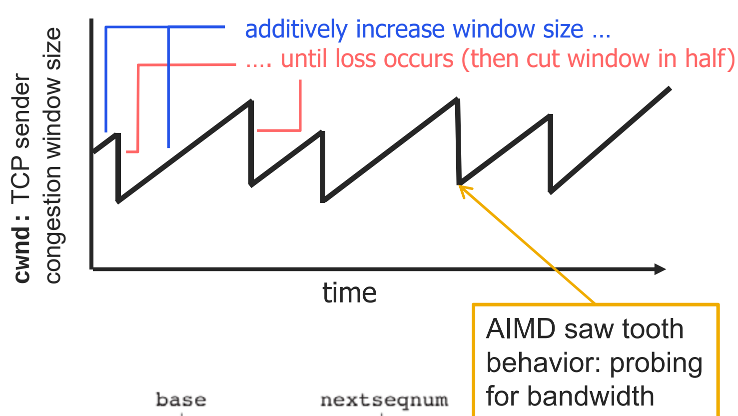
- Router feedback
  - Congested routers mark packets (IP header bit)
  - *Choke packets* returned to senders by network, or as ACKs

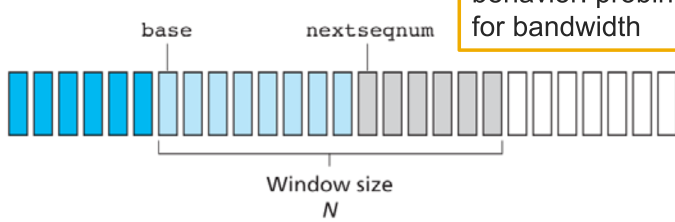
TCP Congestion Control

MONTANA STATE UNIVERSITY

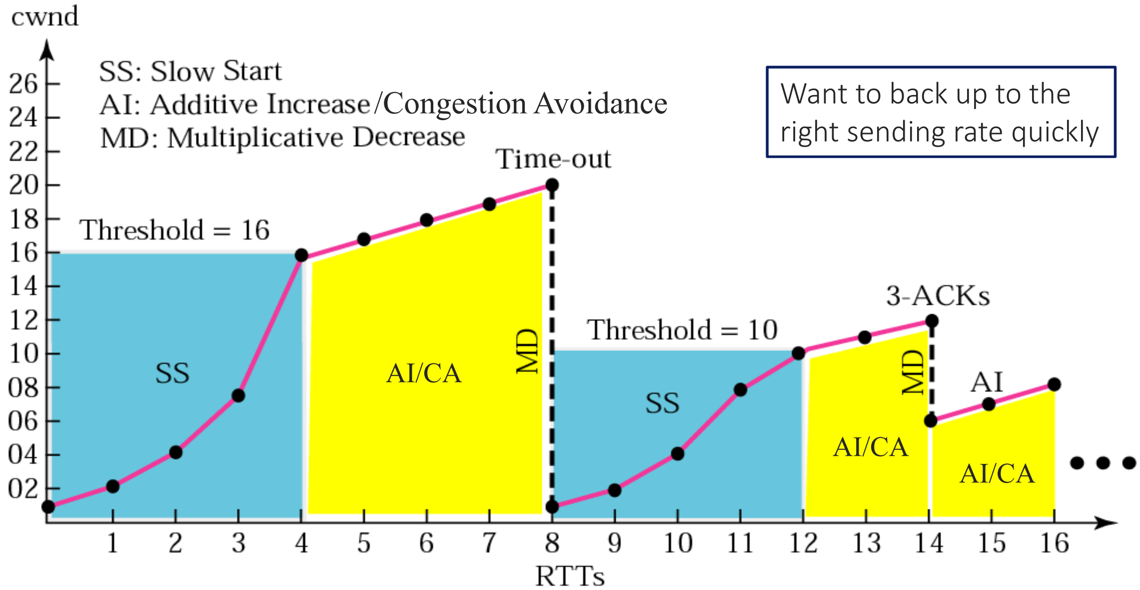
- Sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs
  - Additive increase: increase `cwnd` by 1 MSS every RTT until loss detected
  - Multiplicative decrease: cut `cwnd` in half after loss
- Window size:

```
LastByteSent-LastByteAcked
≤
min(cwnd, rwnd)
```





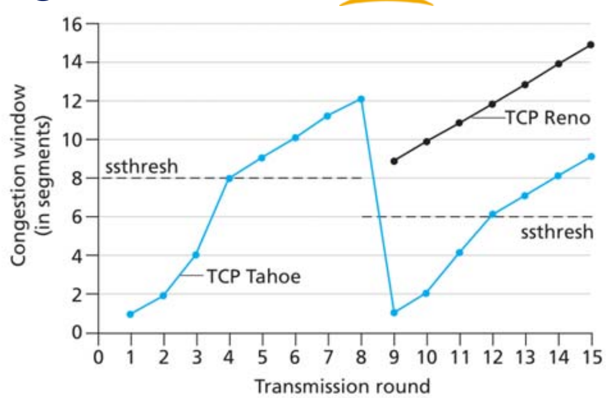
Standard TCP



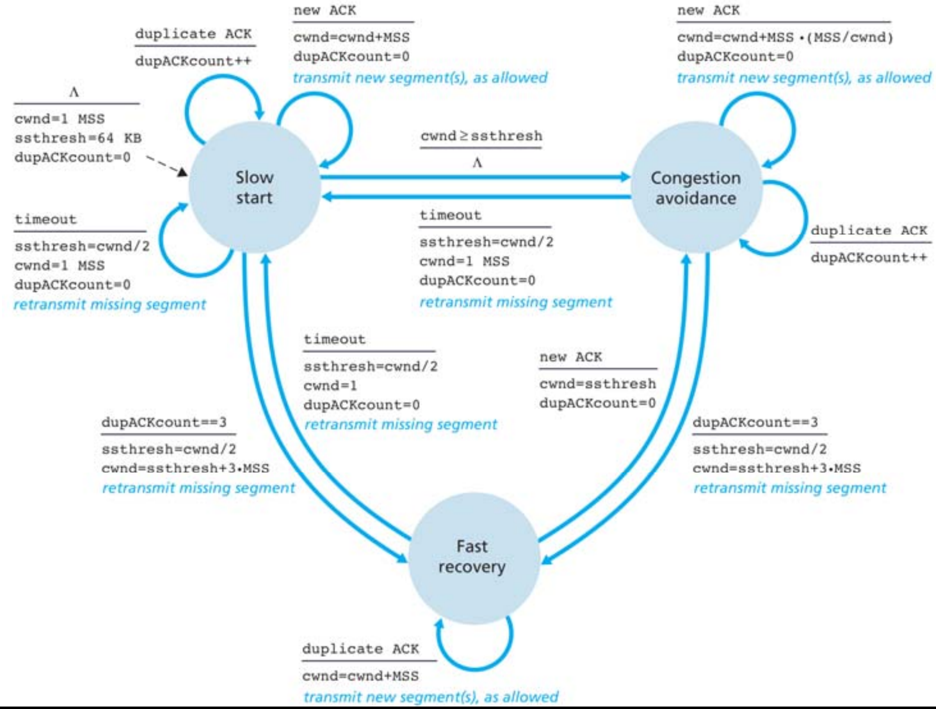
Want to back up to the right sending rate quickly

Mechanism of TCP Cong. Ctrl.

- In Slow start
  - Start with `cwnd = 1MSS`
  - Increment `cwnd` for every ACK
  - Timeout (loss)
    - Set `ssthresh = cwnd/2`
    - Set `cwnd = 1MSS`
- In Congestion avoidance
  - Starts when `cwnd = ssthresh`
  - Increment `cwnd` for every RTT
  - Timeout (nothing is getting through)
    - `ssthresh = cwnd/2`
    - `cwnd = 1MSS`
  - 3 duplicate ACKs (less severe loss) (TCP Reno)
    - `ssthresh = cwnd/2`
    - `cwnd = ssthresh + 3*MSS`
    - Enter fast recovery
- Fast recovery (TCP Reno)
  - Congestion avoidance from `ssthresh + 3*MSS`



- TCP throughput?
  - W = `cwnd` when loss occurs
  - On average `cwnd` is $\frac{3}{4}W$
  - Average TCP throughput $\frac{3}{4}\frac{W}{RTT}$ bps

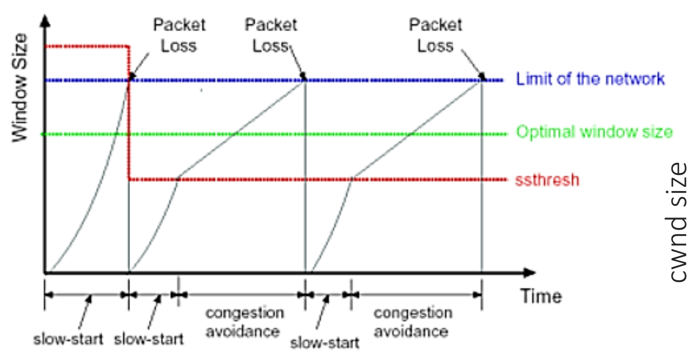


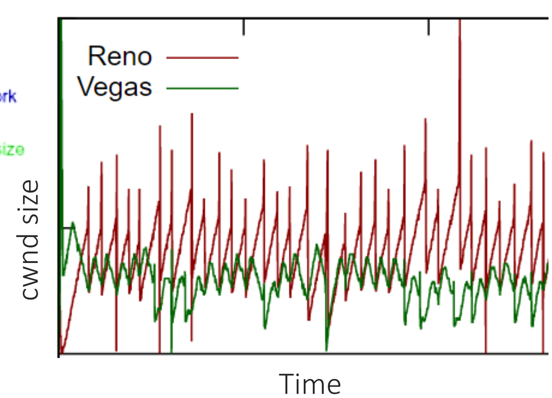
MONTANA STATE UNIVERSITY

155

TCP Vegas

MONTANA STATE UNIVERSITY





TCP Vegas adjusts sending rate in response to growing RTT, which is an indication of congestion and imminent packet drops

TCP Vegas reduces sending rate as a result of delay, before TCP reduces rate as a result of loss

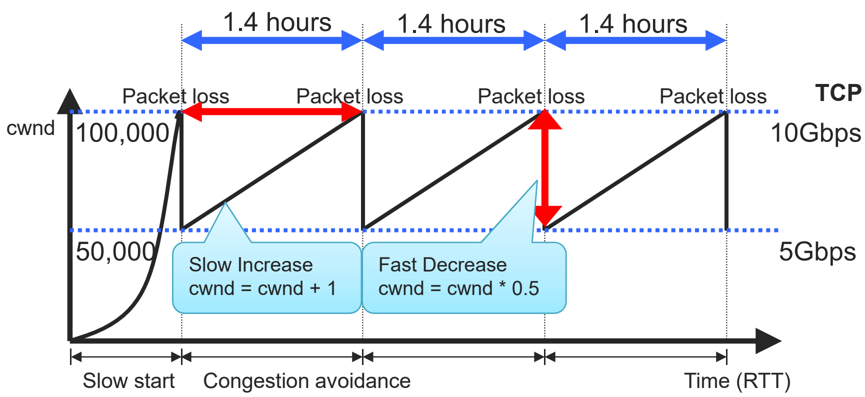
Mountains & Minds

156

Standard TCP

MONTANA STATE UNIVERSITY

- Low window size resilience to packet loss in high-speed networks



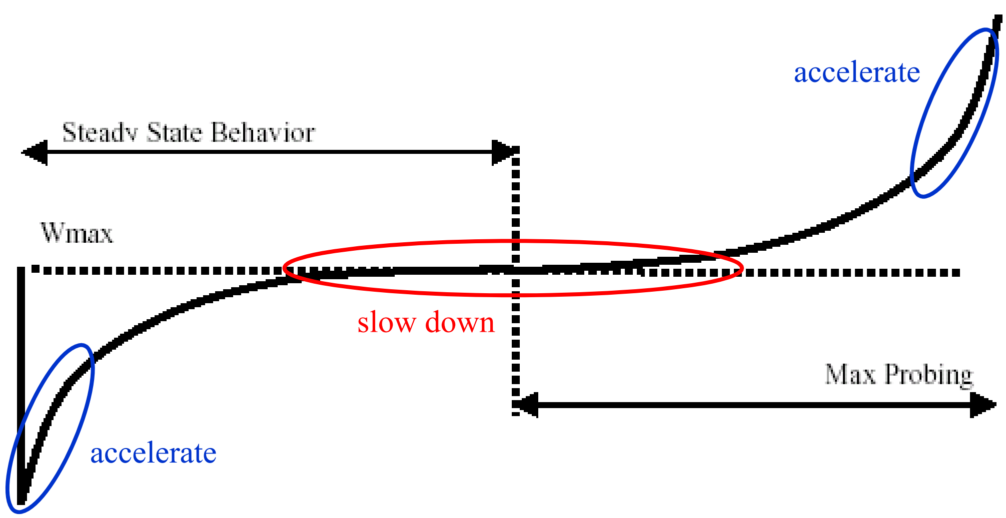
Mountains & Minds

Presentation: "Congestion Control on High-Speed Networks", Injong Rhee, Lisong Xu, Slide 7

157

CUBIC TCP

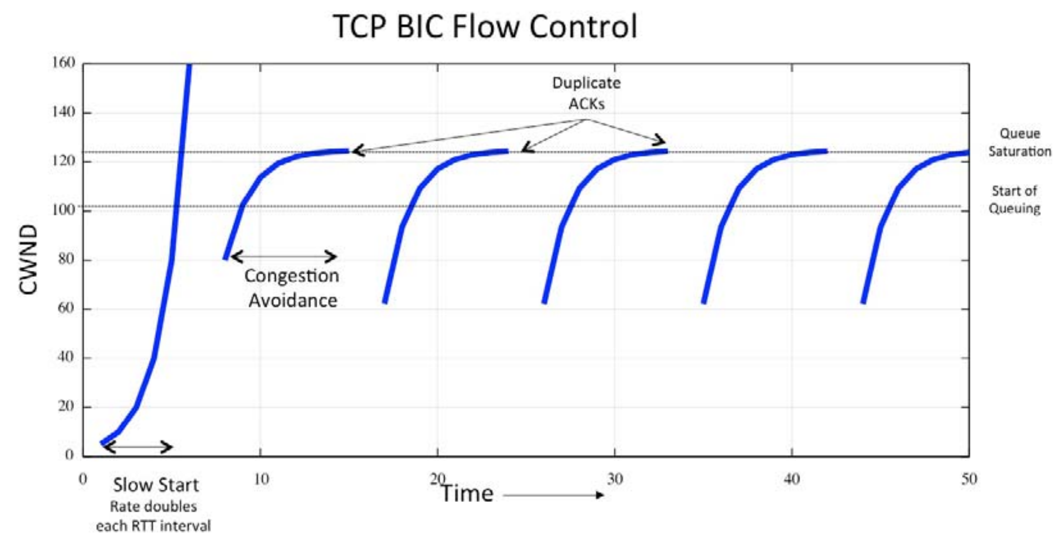
MONTANA STATE UNIVERSITY



Mountains & Minds

158

CUBIC TCP

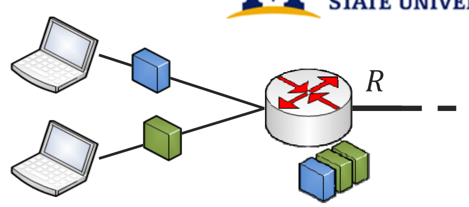


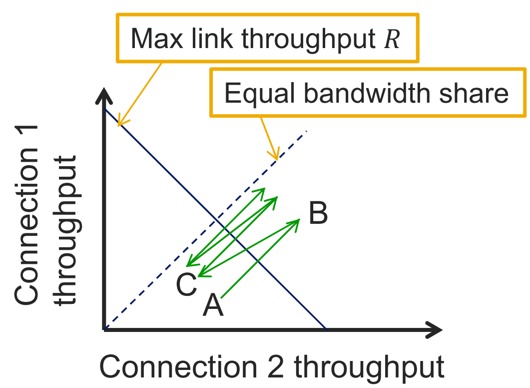
Mountains & Minds

159

TCP and Fairness

- If K TCP sessions share same bottleneck link of bandwidth R, each should have average rate of R/K
- Assume two competing sessions in congestion avoidance stage:
  - Additive increase gives slope of 1, as throughout increases
  - Multiplicative decrease reduces throughput proportionally!
- Unfairness caused by:
  - Parallel TCP connections
  - Unequal RTT – Why?
  - Competing UDP traffic – Why?





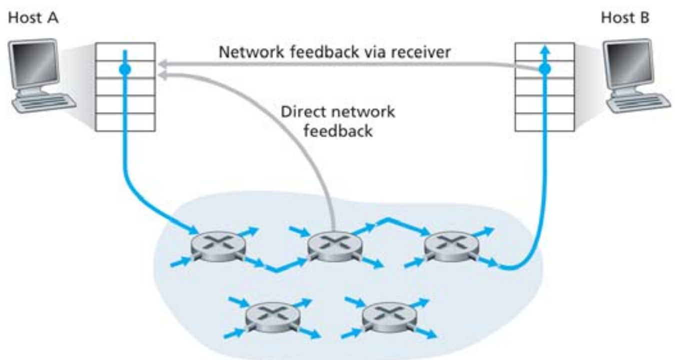
Mountains & Minds

160

Network-assisted congestion ctrl

MONTANA STATE UNIVERSITY

- Asynchronous Transfer Mode (ATM)
  - Direct feedback
  - Choke packets sent from router to sender
- Early Congestion Notification (ECN)
  - Network feedback
  - ECN set by router on queued packets
  - Receiver sets ECN Echo (ECE) bit in IP header of ACK packet
  - Sender treats as loss event



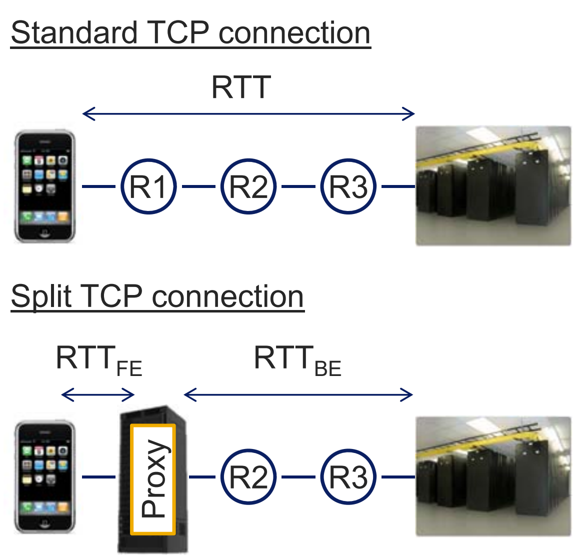
Mountains & Minds

161

TCP Splitting

MONTANA STATE UNIVERSITY

- How to improve TCP rate ramp up, without overwhelming the network?
- Standard TCP
  - Four RTTs per a typical web request
- Split TCP
  - Front end (FE) TCP proxy with negligible RTT_{FE} to user
  - Large congestion window between proxy and back end (BE)
  - Requests can be delivered in close to 1 RTT_{BE}
  - Quicker recovery from losses on last mile networks
- Downsides?
  - No guarantee of end-to-end delivery! Why not?



Mountains & Minds

162