CIS 457 - Data Communications

Nathan Bowman

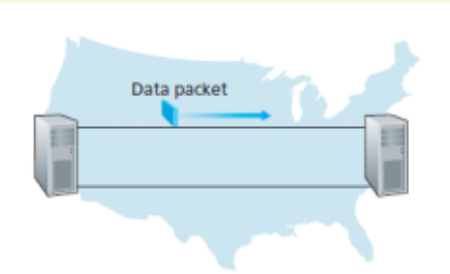
Images taken from Kurose and Ross book

Pipelined Reliable Data Transfer

Saw previously how to have reliable transfer over unreliable network

Unfortunately, performance is an issue

Protocols we developed were based on stop-and-wait, which makes them much too slow to be useful


Data packet

Consider sending a packet from east coast of USA to west coast and back

Due to speed of light, propagation delay for RTT is 30 ms

Assume sender has transmission rate of 1 gigabit per second (Gbps) = 1,000,000,000 bits per second

Assume packet size fixed at 1000 bytes = 8000 bits

transmission delay = L/R = 8000/10^9

Approximately 8 microseconds (10^-6 seconds)

Packet arrives at west coast at RTT/2 + L/R

15.008 milliseconds (ms)

Assume ACK is small enough that transmission delay is negligible

ACK arrives after another RTT/2 due to propagation delay

Total time from sending message to receiving ACK: 30.008 ms

Because of stop-and-wait, sender cannot send again until ACK received

In 30.008 ms, sender was transmitting for 0.008 ms

Utilization -- fraction of time sender is transmitting bits

$$U = (L/R) / (RTT + L/R)$$

$$= 0.008/30.008$$

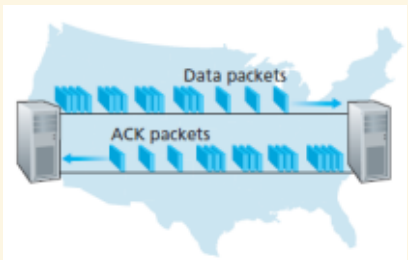
$$= 0.00027$$

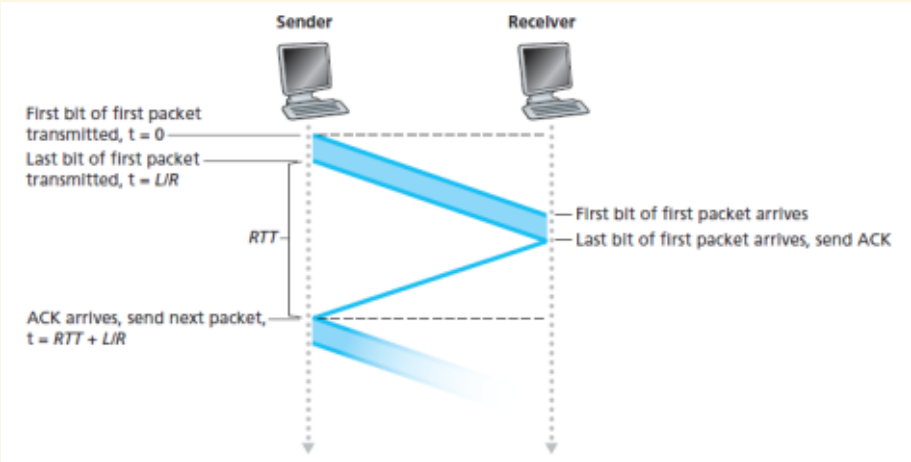
Another way to consider is that sender sent 1000 bytes in 30.008 ms

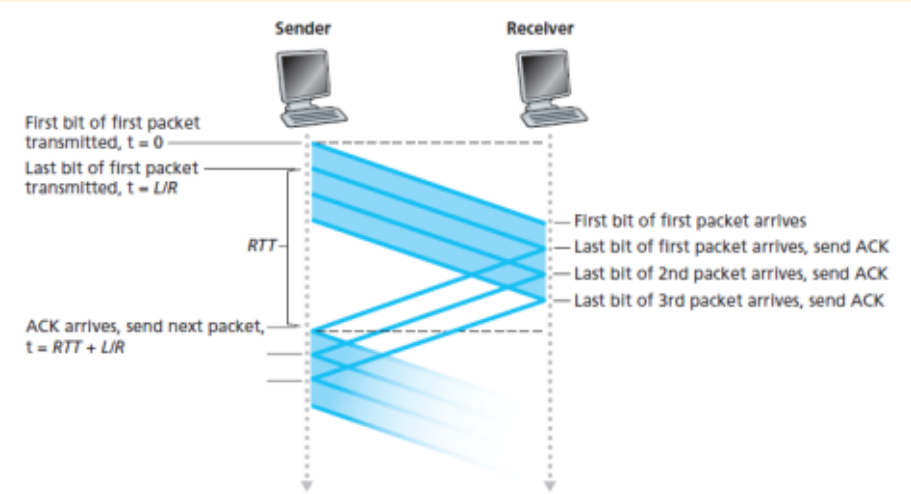
Effective bandwidth is 267 kbps

Try doing anything over a 267 kbps connection -- it would be painful

Solution must be to get rid of stop-and-wait


Data packets
ACK packets


Sender
Receiver
First bit of first packet transmitted, t = 0
Last bit of first packet transmitted, t = L/R
RTT
ACK arrives, send next packet, t = RTT + L/R
First bit of first packet arrives
Last bit of first packet arrives, send ACK


Sender
Receiver
First bit of first packet transmitted, t = 0
Last bit of first packet transmitted, t = L/R
RTT
ACK arrives, send next packet, t = RTT + L/R
First bit of first packet arrives
Last bit of first packet arrives, send ACK
Last bit of 2nd packet arrives, send ACK
Last bit of 3rd packet arrives, send ACK

As seen in image, sending three packets into channel at once essentially triples utilization

Must change reliable data transfer protocols accordingly

Will need to add more sequence numbers to handle more packets at once

Sender (and possibly receiver) must buffer packets to account for possible packet loss

Two abstract protocols to examine:

- Go-Back-N (GBN)
- Selective Repeat (SR)