

R5) Since some loss with voice and video traffic is okay with same-time streaming, voice and video traffic is often sent over TCP rather than UDP in today's internet, because also since most firewalls are programmed to block UDP traffic, by using TCP for video and voice traffic, such traffic will be let through the presented firewall(s) with not much worries of the voice and video traffic being blocked by a firewall.

5/5

R6) Yes, it is possible for an application to enjoy reliable data transfer even when the application runs over UDP where the developer can add reliability at the application layer.

5/5

R10) We needed to introduce timers in our RTT protocols, because packet would get lost that the client would not receive which in turn will not notify anything was sent but was actually sent, so the sender would be waiting for an expediting acknowledgement message but would not receive one because the client didn't receive a packet that was initially lost. So having a timer would realize this issue and possibly a very long time/ endless time of waiting for that acknowledgement message, and the sender will have that "realization" of needing to resend that packet to the client because they didn't receive it, after a decent/pre-determined amount of time.

P15) Cross-Country Example, Figure 3.17
Channel Utilization > 98%
 $L = 1500 \text{ bytes} = 12000 \text{ bits}$

$$\text{RTT} \approx 30 \text{ ms} \approx 0.030 \text{ s}$$

$$R = 1 \text{ Gbps} = 10^9 \text{ bps}$$

window size = ?

$$W_{\text{sender}} = \frac{L/R}{RTT + L/R}$$

$$d_{\text{trans}} = \frac{12000}{10^9} = 1.2 \times 10^{-5} \text{ s}$$

$$0.98 = \frac{(1.2 \times 10^{-5})N}{0.03 + (1.2 \times 10^{-5})} \rightarrow N = 2450.98$$

$$d_{\text{trans}} = \frac{L}{R}$$

The window size would have to be about 2451 packets big.

- 4/10
- P24) a) ~~True~~ (2)
False, because the sender will only send packets within its window and only move on with its window when the earliest packet in its window have been acknowledged. Those ACK messages are the only way the sender's window would progress.
- b) ~~True~~ (2)
False, because the receiver will only send ACK messages for the next packet that it is expecting until it gets it no matter which packet is sent to it.
- c) ~~True~~ (2)
False, because the sender will only send ACK messages for the expected packet it's waiting for, and not send any other packet ACK message.
- d) True, because it will wait and only send ACK messages for the packet it is expecting.

P26) L Bytes

 $\text{NSS} = 536 \text{ Bytes}$ Host A \rightarrow Host B

(7/10)

- a) $L = ?$ so TCP sequence # is not exhausted
 TCP Sequence Number field = 4 Bytes

$4 \text{ Bytes} = 32 \text{ bits}$ ✓
 Maximum Value $L = [2^{32} \text{ bits}]$ so TCP sequence numbers not exhausted

b) dtrans = ?

66 Bytes added before packet sent over 1SS Mbps link
 ↳ Transport, Network, and Data-Link Header
 Ignore Flow Control and Congestion Control
 ↳ Allow A continuous back-to-back sent segments

$536 - 66 = 470 \text{ Bytes left of database}$

$$2^{32} (0.125) = 536,870,912 \text{ Bytes}$$

$$536870912 / 470 = 1,142,279 \text{ packets}$$

$$1142279 (536) = 6.12 \times 10^8 \text{ Bytes}$$

$$6.12 \times 10^8 \text{ Bytes} = 4.9 \times 10^9 \text{ bits} = 4898 \text{ Mbytes}$$

$$4898 / 1SS = 31.6 \text{ seconds}$$

(3) 249 seconds

It will take $\boxed{31.6 \text{ seconds}}$ to transmit the file.

P27) Host A \rightarrow Host B

Up to 126 Bytes have already been received
 Segment 1 = 80 Bytes, Sequence # 127, Source Port # 302
 Destination Port # 80
 Segment 2 = 40 Bytes
 ACK sent whenever segment received

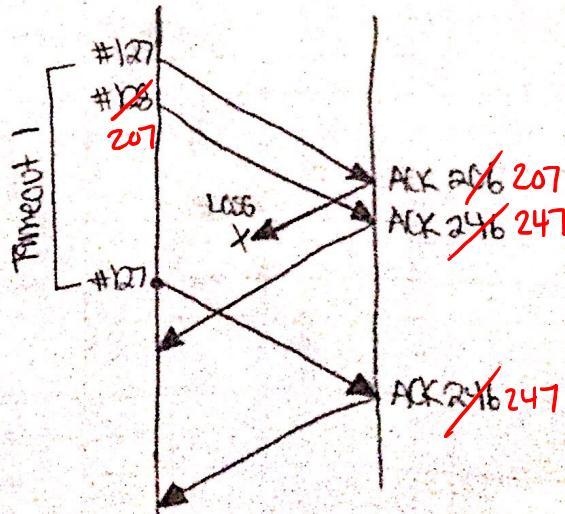
(12/15)

a) Sequence # ~~128~~ 207 ✓
 Source Port # 302 ✓
 Destination Port # 80 ✓

b) Acknowledgment # ~~206~~ 207 ✓
 Source Port # 80 ✓
 Destination Port # 302 ✓

c) Acknowledgment # ~~162~~ 127

d) Host A Host B



P40) Consider 3.58, Assuming Usage of TCP Reno Echan for

(23/25) a) Time Intervals when TCP Slow Start is Operating:

- 1 second to 6 seconds ✓
- 23 seconds to 26 seconds/beyond

b) Time Intervals when TCP Congestion Avoidance is Operating:

- 6 seconds to 16 seconds ✓
- 17 seconds to 22 seconds

c) Since the graph shows it didn't drop to zero, it dropped to about half the window size, it couldn't have been by a timeout, after the 16th transmission round, segment loss must have been detected by a triple duplicate ACK. ✓

d) Since the graph dropped all the way to the window size valued at zero, after the 22nd transmission round, segment loss must have been detected by a timeout. ✓

e) Initial Value of ssthresh at the First Transmission Round
= 32 segments ✓

f) Value of ssthresh at the 18th Transmission Round = 21 segments ✓
(Half of 42 segments)

g) Value of ssthresh at the 24th Transmission Round = 15 segments ✓
(Half of 30 segments)

h) 76th Segment is Sent at the 7th Transmission Round ✓
↳ Based of the initial exponential growth represented by the graph.

i) Packet loss detected after 26th Round by a Triple Duplicate ACK:

The Congestion Window Size = 8 segments ✓
ssthresh = 4 segments
(Half of 8 segments)

j) Suppose TCP Tahoe is Used
Triple Duplicate ACKs at 16th Round
At the 19th Round:

Congestion Window Size = 4 segments ✓
ssthresh = 2 segments
(Half of 4 segments)

TCP Tahoe always sets
congestion window size to 1 seg

k) Suppose TCP Tahoe is Used
Timeout Event at 20th Round

Number of Packets Sent from 17th Round to 20th Round = ?
↳ Interval would have experienced a Slow Start State

$$1 + 2 + 4 + 8 + 16 + 32 = \boxed{63 \text{ packets}} \quad \checkmark$$

17th 18th 19th 20th 21st 22nd

Homework #3

CS 372

Rhea Mae Edwards

PSS) investigate whether either UDP or TCP provides degree of end-point authentication.

15/15

- a) Request and Respond within UDP Packet Server Client with IP Address X Spoofs with Address Y Will Server send its Response?

Yes, the server will send its response because it does have the same IP address since it was given a request with the same UDP packet it would respond to of the request. ✓

- b) Server receives SYN with IP Source Address Y
Responds with SYNACK

Receives ACK with IP Source Address Y with correct acknowledgement #
Assuming server chooses random initial sequence # ✓

No "man-in-the-middle"

Can server be certain client is indeed at Y?

The server can be certain that the client is indeed at Y...
But there's really no way mostly that the server can be 100% certain.
There are always ways to manipulate the facade of the client without being 100% certain of the truth of the client entirely.