Problems 3

R2:

For sending a letter:

1. Family member is required to give the delegate
   1. the letter
   2. Address of the destination house
   3. Name of the recipient.
2. The delegate clearly writes the recipient's name on the top of the letter.
3. The delegate puts the letter in an envelope and writes the address of the destination house on the envelope.
4. The delegate gives the letter to the planet's mail service.

At the receiving side:

1. The delegate receives the letter from the mail service
2. The delegate takes the letter out of the envelope, and takes note of the recipient name written at the top of the letter.
3. The delegate gives the letter to the family member with this name.

Nowhere is it needed for the envelop to be opened since all the information needed for delivery is on the outside.

R4:

There are a few reasons why an application developer would choose UDP over TCP

1. Since TCP has congestion control, there are times when TCP is much slower during congestions periods. Also UDP doesn’t keep track of parameters and doesn’t maintain a connection state. So it’s faster.
2. The server can handle more clients with UDP
3. Applications that don’t rely on connections use UDP
4. IP video conferences run on UDP to avoid TCP congestions.

R9:

Sequence numbers are required for a receiver to find out whether an arriving packet contains new data or is a retransmission.

P1:

|  |  |  |
| --- | --- | --- |
|  | Source Port Numbers | Destination Port Numbers |
| 1. A -> S | 100 | 68 |
| 1. B -> S | 230 | 68 |
| 1. S -> A | 68 | 100 |
| 1. S -> B | 68 | 230 |

1. Yes they can be the same
2. No they cant

P3:

Three 8-bit bytes: 01010011, 01100110, 01110100.

01010011 + 01100110 = 10111001 + 01110100 = 00101110 there is a 1 in the carry bit

Ones complement: 11010001

To detect errors, the receiver adds the three original words and the checksum. If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected.

P15:

Transmission rate is 1 Gbps

1,500 bytes to bits = 1500 \* 8bits

Round trip time is 30 milliseconds

So…(1500 \* 8bits) / 10^9 = 12 milliseconds per packet

Solve for N:

.98 = (.012 \* N) / (30.012) 🡺 2451

P17:

Also attached at a pdf



P24:

a) True, Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at t 0 . At t1 (t1 > t 0) the receiver ACKS 1, 2, 3. At t 2 (t 2 > t1) the sender times out and resends 1, 2

b) True, by essentially the same scenario as in (a).

c) True, in an alternating-bit protocol, the sequence number of the packet alternates between 0,1. In SR protocol, the sender and receiver windows will not coincide. The sender and receiver window size of alternating bit protocol and SR protocol is 1.

d) True, Note that with a window size of 1, SR, GBN, and the alternating bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window). A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

P38:

When the loss event occurs, the rate is approximately equal to the cwnd segments per RTT. So the sender rate is cwnd/RTT. This is because cwnd is a variable operated at the sender side by the congestion control mechanism. Ssthresh is a variable used to store second state value. So when a segment is loss the ssthresh is reset.

P40:

a) TCP slow start is operating in the intervals [1,6] and [23,26]

b) TCP congestion avoidance is operating in the intervals [6,16] and [17,22]

c) After the 16th transmission round, packet loss is recognized by a triple duplicate ACK. If there was a timeout, the congestion window size would have dropped to 1.

d) After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

e) The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.

f) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

g) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 26. Hence the threshold is 13 during the 24th transmission round.

h) During the 1st transmission round, packet 1 is sent; packet 2-3 are sent in the 2nd transmission round; packets 4-7 are sent in the 3rd transmission round; packets 8-15 are sent in the 4th transmission round; packets 16-31 are sent in the 5th transmission round; packets 32-63 are sent in the 6th transmission round; packets 64 – 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

i) The threshold will be set to half the current value of the congestion window (8) when the loss occurred and congestion window will be set to the new threshold value + 3 MSS . Thus the new values of the threshold and window will be 4 and 7 respectively.

j) Threshold is 21, and congestion window size resets to 1. At the 19th round, the window is 4.

k) Round 17, 1 packet; round 18, 2 packets; round 19, 4 packets; round 20, 8 packets; round 21, 16 packets; round 22, 21 packets. So, the total number is 52.

P43:

The receiver has enough buffer for entire file, so the flow control won't affect the send rate of sender. And because there is no packet loss and never timeout, the congestion control won't affect the send rate, either. But since the link conn Host A to the Internet is R bps, which is much smaller than the capable of sending data into TCP socket, TCP buffer will be always full, the process will passing data to its TCP socket at R bps on average.

P55:

1. Like TCP, UDP can be flooded as well. The characteristics of a UDP Flood attack are somewhat different, but TCP and UDP flood attacks do have their similarities and similar grounds through which attackers attack as well as similar defense mechanisms such as the initial sequence number. The server will send its response to the spoofed address Y as it is the matching IP address. Spoofing IP addresses is a common form of attack for attackers with UDP flooding as spoofing an IP address of the valid client will make sure the packet doesn’t get sent to the valid client. Like with TCP, IP spoofing is one way attackers can make a server send a response away from the intended client as it can duplicate the IP address to match the valid client’s IP address.
2. Like with TCP, the server assigns an initial sequence number to a SYNACK packet. It’s difficult for the server to determine whether this is the correct IP source address as there’s no way to detect whether it is a spoofed address or valid address. The client could’ve retrieved the sequence number that corresponded with the correct acknowledgment number. So the server can never really be certain that it was sent to the valid IP address Y as opposed to a spoofed address that might actually be address X. Recall that with the correct acknowledgment number, it is possible to manipulate the server into thinking it’s a valid client as the acknowledgment number corresponds with the SYN segment sent earlier.