Fall 2024, CS 3953: Computer Networks

Homework 3 Solution

Solution to problem 1

Yes, both segments will be directed to the same socket. For each received segment, at the socket interface, the operating system will provide the process with the source IP addresses to determine the origins of the individual segments.

Solution to problem 2

a)~ d):

	source port numbers	destination port numbers
a) A→S	4467*	23
b) B→S	5513*	23
c) S→A	23	4467*
d) S→B	23	5513*

* Note here A and B can have any possible port numbers larger than 1024.

e) Yes.

f) No, it's impossible that A and B have the same port number.

Solution to problem 3

UDP has better control over what and when is sent in a segment. The reasons that some applications choose to run over UDP includes but not limited to:

- no connection establishment (which can add delay)
- no retransmission (which can add delay)
- simple: no connection state at sender, receiver
- small segment header: 8 bytes
- no flow control and congestion control: UDP can blast away as fast as desired

Solution to problem 4

The 32-bit sequence number can have up to 2^{32} sequence numbers. So in TCP case, the whole sequence number space can send up to 2^{32} bytes before wrapping around.

1) If the line is 56-kbps, the time to transmit the 2^{32} bytes is: $2^{32} * 8/56k$ =170.4hours

2) If the line is 10-Mbps, the time to transmit the 2^{32} bytes is: $2^{32} * 8/10M$ = 28.5minutes

3) If the line is 4-Gbps, the time to transmit the 2^{32} bytes is: $2^{32} * 8/4G$ = 8.6 seconds

4) Suppose a packet can stay in Internet for 0.5 minute at most, the cost time to transmit a 32-bit sequence number in 1G network is 34s, which is larger than 0.5 minute, i.e., so, it is enough.

Solution to problem 5

a) Consider sending an application message over a transport protocol. With TCP, the application writes data to the connection send buffer and TCP will grab bytes without necessarily putting a single message in the TCP segment; TCP may put more or less than a singe message in a segment. UDP, on the other hand, encapsulates in a segment whatever the application gives it; so that, if the application gives UDP an application message, this message will be the payload of the UDP segment. Thus, with UDP, an application has more control of what data is sent in a segment.

b) With TCP, due to flow control and congestion control, there may be significant delay from the time when an application writes data to its send buffer until when the data is given to the network layer. UDP does not have delays due to flow control and congestion control.

Solution to problem 6

a)

GoBackN:

A sends 10 segments in total. They are initially sent segments 1, 2, 3, 4, 5, 6 and later re-sent segments 3, 4, 5 and 6. B sends 9 ACKs. They are 1 ACKS with sequence number 1, 4 ACKS with sequence number 2, and 4 ACKS with sequence numbers 3, 4, 5 and 6.

Selective Repeat:

A sends 7 segments in total. They are initially sent segments 1, 2, 3, 4, 5, 6 and later re-sent segments 3. B sends 6 ACKs. They are 5 ACKS with sequence number 1, 2, 4, 5, 6. And there is one ACK with sequence number 3.

Solution to problem 7

a) TCP slow start is operating in the intervals [1,6] , [15,19] and after round 24

b) After the 10th 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.

c) After the 14th transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

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

e) The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 10, the congestion windows size is 36. Hence the threshold is 18 during the 12th transmission round.

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 14, the congestion windows size is 24. Hence the threshold is 12 during the 15th transmission round.

g) 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; Thus packet 60 is sent in the 6th transmission round.

h) The congestion window and threshold will be set to half the current value of the congestion window when the loss occurred. Thus the new values of the threshold will be 5 and window will be 5+3=8.

i) Threshold is 18, and the congestion window size of 11th round is 1.

Solution to problem 8

In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgements are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate or R << S.