Xiangling Wu

110990426

xianwu

09/30/2019

CSE 310, HW 2

Ch. 3

P1) a) Source: 1234, Dest: 23

b) Source: 1235, Dest: 23

c) Source: 23, Dest: 1234

d) Source: 23, Dest: 1235

e) Yes.

f) No.

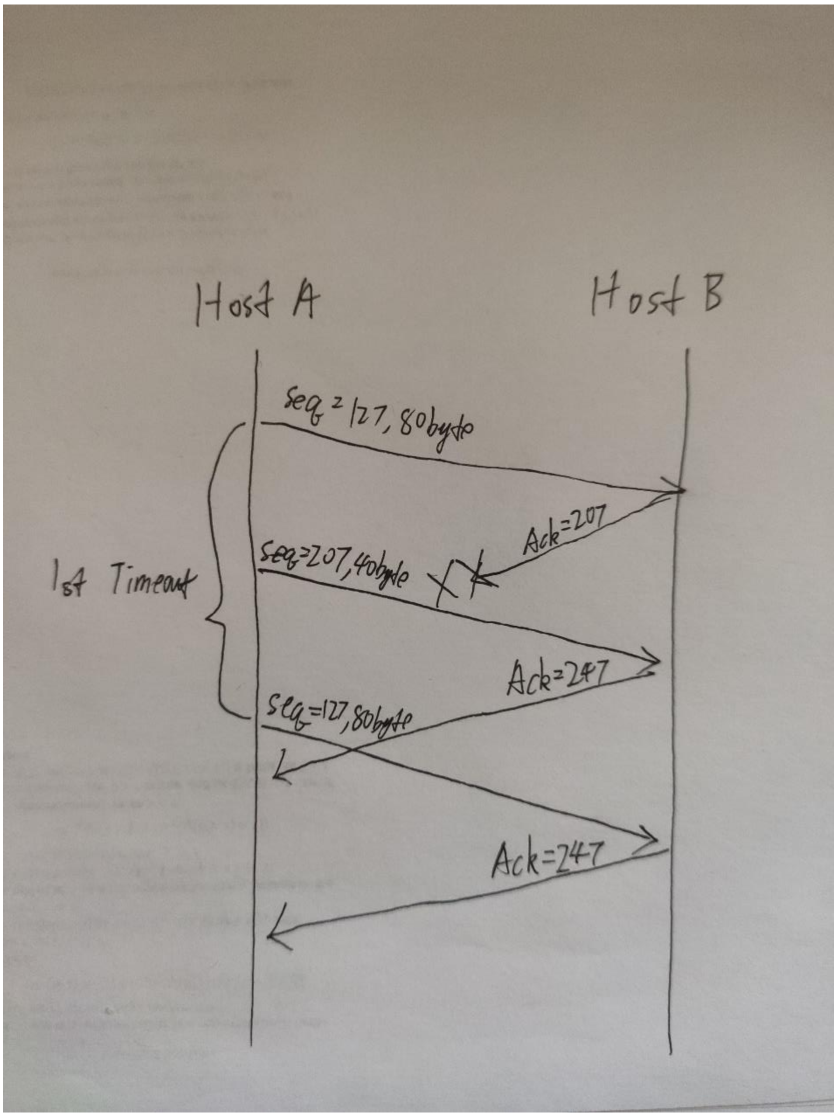
P2) The source and dest port number will be the port numbers bound to B and A, respectively. The source IP addr is B and the dest IP addr is A.

P25) In order to send data, TCP will first set up a send buffer then encapsulate and send what is written to the buffer. The data may not be entirely written to the buffer, and sometimes there may be retransmissions. On the other hand, UDP simply encapsulate whatever needs to be sent into a segment and send. Therefore UDP offers better application control of what data is sent than TCP does.

TCP has flow control and congestion control mechanisms controlling the sending rate. Therefore there could be more delay when sending the segment. On the other hand, UDP has no delay due to flow control and congestion control and could send segments immediately. Therefore UDP offers better application control of when segment is sent than TCP does.

P27)

a) The sequence number is 207, the source port number is 302, and the dest port number is 80.
b) The ACK number is 127+80=207, the source port number is 80, and the dest port number is 302.
c) The ACK number is 127.
d)



P33) The retransmitted packet and the original packet will have the same sequence number, therefore the ACK numbers responding to both packets will be the same. If we receive an ACK number for a retransmitted packet, we could not know that is it responding to the original packet or the retransmitted packet. This will mess up the calculation of Sample RTT.

P34) SendBase-1 <= LastByteRecv

SendBase-1 is the sequence number of the last cumulatively ACKed byte, and LastByteRecv is the sequence number of last byte received and placed in the buffer. Therefore SendBase-1 must be not greater than LastByteRecv, and only equal when the ACK of previously sent packet arrives before the new packet sent arrives.

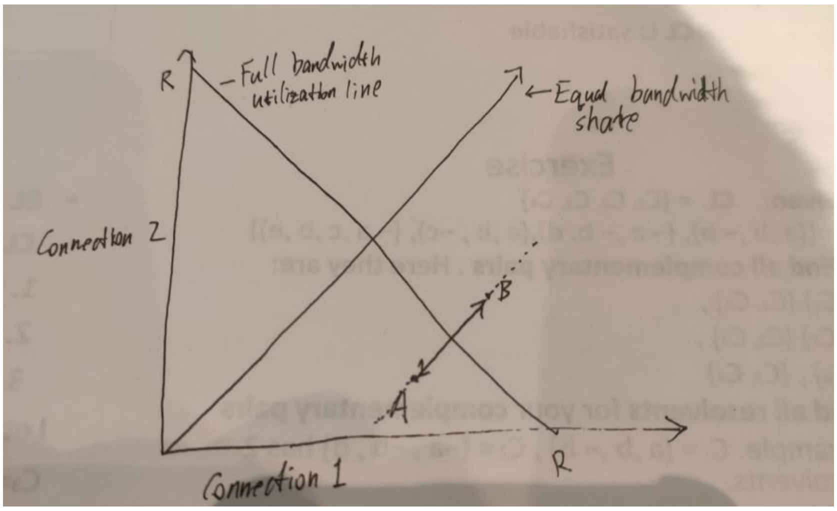
P35) y-1 <= LastByteRecv

y is the last ACK number of received packet, thus byte y-1 must have already been received. Therefore y-1 must be no greater than LastByteRecv, and only equal when no new packets are received before the latest ACK arrives.

P36) Because the packets may arrive out-of-order instead of being lost. Allowing tolerance of 2 duplicates can avoid redundant retransmissions caused by out-of-order packets. 3 or more duplicates ACKs indicates a high possibility of lost packets.

P41) Suppose that connection 1 and 2 realize throughputs are indicated by point A in the diagram below. As A is below line R, both connections will increase their window additively. That is, A will move up-right along a 45 degree line. As the connections consume more than the full bandwidth R, packet loss will occur and the window of both connections will be decrease additively. That is, the throughputs will get above line R to point B, and then move down-left along a 45 degree line (since both connection decrease at the same rate).

As the increment and decrement of throughputs continue, the throughputs will be moving along the same 45 degree line passing through A and B and never get nearer to the line of equal bandwidth share. The AIAD algorithm doesn’t converge to an equal share algorithm.



P42) Even though doubling timeout interval provides congestion control over retransmissions, it does not provide control over packets which are sent for the first time. Window-based algorithm is needed to prevent TCP from sending packets to congested network.

P43) Flow control controls the rate of transmission to avoid overwhelming the receive buffer at the destination host. In this case, the receive buffer is large enough to hold the entire file, therefore it is not the matter of flow control.

Congestion control controls the rate of transmission to avoid overwhelming intermediate network links. In this case, the transmission rate of the Internet is limited to R bps, but the sending rate S could be 10x faster than R, therefore congestion control will throttle the transmission rate on the sender side to prevent the network from being overwhelmed.