CSE 4153/6153 Assignment 4

(1) **Connection establishment in TCP:**

The connection establishment process involves exchange of TCP packets with no TCP payload (the TCP packets have only the TCP headers)

Initiator chooses a sequence number. Sends SYN packet (TCP packet with no data; the SYN flag is set in the TCP header). The 32-bit sequence number field indicates the chosen sequence number. The window size chosen by the initiator (the storage allocated for that connection) is indicated in the Window size field.

Responder chooses its sequence number. Send a SYN-ACK packet (the SYN and ACK flags are set). The sequence number field indicates the sequence number. The acknowledgment field is one higher than the sequence number chosen by the the initiator. The window size is also depicted.

The initiator sends an ACK packet. ACK flag is set. The acknowledgment field is one higher than the sequence number sent by the responder.

Connection Termination:

One of the two (say A) sends a FIN packet (a packet with the FIN flag set in the the TCP header), indicating that A does not have any more application data to send. The other end (say B) sends an ACK for the FIN. The other end B then sends a FIN packet. A responds with an ACK packet acknowledging the FIN packet. The ACK and FIN packets form B could be one packet in which both FIN and ACK flags are set.

(2) Three timers:

1. Retransmission timer
2. Persistence timer
3. Keep-alive timer

Retransmission timer: Usually set to 1.5 to 2 times the estimated round-trip timer. Timer started for every packet sent. If no acknowledgment is received before the timer fires the packet is resent.

Persistence timer: Used top prevent deadlocks. This timer is started whenever *a non zero window size advertisement is sent following a previous zero window size advertisement*. If no response is forthcoming from the other end before the timer expires the non zero window size advertisement is resent.
Without this timer the following deadlock can occur: A has sent a window size of 0 earlier, thus requesting B to stop sending packets. A later sends a TCP packet indicating a non zero window size. However this packet is lost. Now A is waiting for B to send application data, while B is waiting for A to send a message that B can start sending.

Keep-alive Timer: In some scenarios the processes at the two ends may wish to maintain a TCP connection even when there is no application data to be sent for long periods of time. In order to differentiate between such connections and connections that have died because of untimely death of a process at one end, the TCP layers at both ends send keep-alive packets when the keep-alive timer expires.

(3) Dynamic Window Sizes:

The window size in the TCP header sent by A indicates the remaining storage buffer for the connection. The initial window size advertisement reflects the storage buffer allocated. Subsequently, depending on how much of the buffer is read and cleared by the application, the remaining storage will vary dynamically. If only hundred bytes remain in A's buffer at some time, A needs to inform B to not send over 100 application bytes.

(4)

1. A → B: SYN packet; sequence number 100,000; SYN flag set; Window size 8000;
2. B → A: SYN/ACK packet; seq 200,000; ack 100,001; SYN and ACK flags set; Window 8000

3. A → B: ACK ; ack 200,001; ACK;
4. A → B: 1400 bytes data; seq 100,000;
5. B → A: 1400 bytes data; seq 200,000; ACK; ack 101400; W = 6600;
6. (A can send two MTUs)
7. A → B: 1400 bytes data; seq 101400; ACK; ack 201400; W=6600;
8. A → B: 1400 bytes data; seq 102800; W = 8000;
9. (B can send TWO MTUs)
10. B → A: 1400 bytes data; seq 201400; ACK; ack 104200; W=3800;
11. B → A: 1400 bytes data; seq 202800; W=4400;
12. (A can send four MTUs as per the rules for congestion window size. However A can send only three MTUs due to remaining buffer space)
13. A → B: 1400 bytes data; seq 104200; ACK; ack 204200; W=5200;
14. A → B; 1400 bytes data; seq 105600; W = 7000;
15. A → B: 1400 bytes data; seq 107000; W = 8000;
16. (B can send four MTUs – but need to send only 800 bytes – one MTU)
17. B → A: 800 bytes, seq 204200; ACK; ack 108400; W=6000
18. (A can send eight MTUs according to congestion control rule; 4 MTUs according to buffer size rule; but A needs to send only two more)
19. A → B: 1400 bytes data; seq 108400; ACK; ack 205000; W=8000;
20. A → B: 200 bytes data; seq 109800;
21. B → A: FIN; ACK; ack 110000;
22. A → B: FIN; ACK; ack 205000;
23. B → A: ACK; ack 110000;

No difference for the three cases. Such differences are transparent to the TCP layer. Only the IP layer has to deal with such differences.