HW 3 Chapter 3 – Part 1 Solutions

AR7. 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 IP addresses to determine the origins of the individual segments.

AR 8. For each persistent connection, the Web server creates a separate "connection socket". Each connection socket is identified with a four-tuple: (source IP address, source port number, destination IP address, destination port number). When host C receives and IP datagram, it examines these four fields in the datagram/segment to determine to which socket it should pass the payload of the TCP segment. Thus, the requests from A and B pass through different sockets. The identifier for both of these sockets has 80 for the destination port; however, the identifiers for these sockets have different values for source IP addresses. Unlike UDP, when the transport layer passes a TCP segment's payload to the application process, it does not specify the source IP address, as this is implicitly specified by the socket identifier.

AR 9. Sequence numbers are required for a receiver to find out whether an arriving packet contains new data or is a retransmission.

AR 10. To handle losses in the channel. If the ACK for a transmitted packet is not received within the duration of the timer for the packet, the packet (or its ACK or NACK) is assumed to have been lost. Hence, the packet is retransmitted.

AR12.

a) The packet loss caused a time out after which all the five packets were retransmitted.

b) Loss of an ACK didn't trigger any retransmission as Go-Back-N uses cumulative acknowledgements.

c) The sender was unable to send sixth packet as the send window size is fixed to 5.

AR13.

a) When the packet was lost, the received four packets were buffered the receiver. After the timeout, sender retransmitted the lost packet and receiver delivered the buffered packets to application in correct order.

b) Duplicate ACK was sent by the receiver for the lost ACK.

c) The sender was unable to send sixth packet as the send window size is fixed to 5.

When a packet was lost, GO-Back-N retransmitted all the packets whereas Selective Repeat retransmitted the lost packet only. In case of lost acknowledgement, selective repeat sent a duplicate ACK and as GO-Back-N used cumulative acknowledgment, so that duplicate ACK was unnecessary.

AP4.
a) Adding the two bytes gives 11000001. Taking the one's complement gives 00111110.

	01011100
	01100101
Sum	11000001
Checksum	00111110

b) Adding the two bytes gives 01000000; the one's complement gives 10111111.

	11011010
	01100101
Wraparound	100111111
	▶1
Sum	01000000
Checksum	10111111

c) First byte = 01010100; second byte = 01101101.

AP15
*It takes 12 microseconds (or 0.012 milliseconds) to send a packet, as 1500*8/109=12 microseconds. In order for the sender to be busy 98 percent of the time, we must have*
$util = 0.98 = (0.012n)/30.012$
or approximately 2451 packets.

AP23. In order to avoid the scenario of Figure 3.27, we want to avoid having the leading edge of the receiver's window (i.e., the one with the "highest" sequence number) wrap around in the sequence number space and overlap with the trailing edge (the one with the "lowest" sequence number in the sender's window). That is, the sequence number space must be large enough to fit the entire receiver window and the entire sender window without this overlap condition. So - we need to determine how large a range of sequence numbers can be covered at any given time by the receiver and sender windows.

Suppose that the lowest-sequence number that the receiver is waiting for is packet m. In this case, it's window is [m,m+w-1] and it has received (and ACKed) packet m-1 and the w-1 packets before that, where w is the size of the window. If none of those w ACKs have been yet received by the sender, then ACK messages with values of [m-w,m-1] may
still be propagating back. If no ACKs with these ACK numbers have been received by the sender, then the sender's window would be [m-w,m-1].

Thus, the lower edge of the sender's window is m-w, and the leading edge of the receivers window is m+w-1. In order for the leading edge of the receiver's window to not overlap with the trailing edge of the sender's window, the sequence number space must thus be big enough to accommodate 2w sequence numbers. That is, the sequence number space must be at least twice as large as the window size $k \geq 2w$.