Daniel Meyer

0405182

CSIS 202-1008

**Homework Week 4**

**Review Questions:**

R1) a) A simple transport-layer protocol will open a socket with a source and destination port number for each of the application processes running. This would be the header for data chucks being sent to another host. This header would also be used when receiving data chucks as it would use this header information to find which socket to send the data to through demultiplexing.

1. Adding a source port number to the header would provide a “return address” to the destination process.
2. The protocol doesn’t “have to do anything” in the core of the computer network.

R2) a) A protocol delegates can use to deliver letters to family member from sending family members would be to open the envelope (packet) check for the unique name and send the letter to the corresponding recipient (socket) in the household.

b) IN my protocol the mail service has to open the envelope and examine the letter to find the recipient because there is no header information on the recipient.

R3) The source port number for data between Host B to Host A will be port number *y*and the destination port number will be port number *x*.

R4) A developer would choose UDP over TCP as UDP provides more control over the data being sent and when, a connection doesn’t have to be established, UDP is connectionless, and UDP uses only 8 bytes of overhead instead of TCP’s 20 bytes.

R5) Voice and video traffic is usually sent over TCP rather than UDP because TCP provides minimal packet loss and errors. Furthermore, TCP is more secure as opposed to UDP.

R6) It is possible for applications to have reliable data transfer through UDP if the developer builds reliability into the application itself, however it would take longer to develop.

R7) Host A and Host B will both have their UDP segment sent to the same socket in Host C, but Host C will know the segments came from two different hosts as their source port numbers will differ.

R8) The web server running on Host C using port 80 will have different sockets for the incoming data from Hosts A and B. This is through demultiplexing when two connections have the same source port numbers. It knows the difference between the two Hosts by their different source IP addresses instead.

R9) Sequence numbers were needed to handle scenarios where corrupted ACKs and NAKs were possible.

R10) A timer was needed for cases where retransmission was needed when a data packet was lost, an ACK or NAK was corrupt, or both were delayed.

R11) A timer would still be needed in the case of data packet loss or ACK/NAK loss, or both. A known RTT delay would simply mean a timer could be set more accurately.

R12) N/A – Companion Site 404

R13) N/A – Companion Site 404

**Practice Problems:**

P2) The source and port destination values of the segments from the server to the client’s processes are 7532 and 26145 for Host C and 26145 for Host A for the source port and 80 for both destination ports. The IP addresses in the datagrams are the server’s IP address.

P12) The protocol would still work however, there would be a large number of duplicate packets as up to 2n number of packets would be sent.

P15) (30.000012) / (1.2\*10^-5) \* .98 = 2,450,999.98 bits for the windows size in order to have 98% channel utilization.