Brandon Vicinus

Assignment 3

CNT4004

1. 0x3346 = 0011 0011 0100 0110   
   0x7766 = 0111 0111 0110 0110   
   0x71AB = 0111 0001 1010 1011  
     
   total = 1 0001 1100 0101 0111  
   after wrap around = 0001 1100 0101 1000

after 1’s compliment = 1110 0011 1010 0111  
  
B. No, there is no way for the UDP server to know if they are absolutely certain they did not receive a corrupt package by using checksum. Lets say 0x3346 changed to 0x3347 and 0x71AB changed to 0x71AA. There would be 1 extra ‘1’ due to the change from 6 to 7. There would also be 1 less ‘1’ from the B changing to an A. This would add 1 and subtract 1 from the check sum at the same time. This would not be detectable.

1. See attached paper.
2. The incorrect receiver shown in Fig. 3.75 can enter a deadlocked state with the correct sender in Fig 3.11. If the receiver is waiting for a 0 call from below, and the sender is sending out a 1, they will both enter a deadlocked state. This is because the receiver will continue to send NAKs if it is receiving packets with a 1 sequence number. In return, the sender will continue to keep sending its packets with a sequence number of 1 because the receiver is returning with NAKs. To fix this, the Fig. 3.12 has an additional self-loop on both states of the receiver if it receives a packet that is not corrupt but not the sequence number they were expecting.
3. Part A and B: see attached paper.   
   C.) Having a short time-out time is greedy for the user. It makes sure that the users packets are getting out to the receiver by flooding the network with duplicate packets. But the users data will definitely get to the receiver. If the time-out time is very long, the user will suffer because they are waiting extra time just to be sure that their packet is lost and not just ‘stuck in traffic’ or on an alternate longer route. The network will benefit from users having longer wait times because there will be less duplicate packets sent when they do not need to be. Therefore, other users of the network will benefit more if the overall population has longer time-out times.
4. See attached paper.
5. See attached paper.
6. When decoding a layer 4 header, being TCP, UDP or some others such as MTCP (Multipath TCP), or SCTP (Stream Control Transmission Protocol), one obvious way to decipher what kind of protocol header you are looking at is the size of the header. A typical TCP header is 128 bits while a UDP header is only 32 bits.
7. A.) UDP has more control of what data is sent in a segment compared to TCP because UDP simply packages the data passed to it and immediately passes the segment onto the network layer after adding a small amount of port numbers and other information. TCP has a concept of ‘congestion control’ so data’s exact sending process is subjective to the network congestion.   
   B.) UDP has more control of when a segment because there is no ‘connection’ between client and server. UDP is said to be ‘connectionless’ meaning UDP simply pushes the segment onto the network and that’s it. A segment is sent when UDP says so. TCP, on the other hand, uses secure connections and requires acknowledgements to be sent confirming packet delivery. These connections take time, causing delays. Additionally, if a packet is not delivered, the client will have to resend their data, postponing other data from being sent.
8. There are two ways to terminate a TCP connection. The first way is to send a reset signal to the client. This will terminate any connection immediately. These types of resets are from an external source, perhaps a hacker or a firewall. Think about China’s firewall preventing a Chinese resident from accessing Facebook. The Chinese firewall will send a reset message to the client and end their connection. The second way to end a TCP connection is for the client to end their connection by sending a FIN, or finish, command to the receiver. The receiver will send an ACK to the client to confirm their termination.
9. 

