CSE 677: Homework 2

(Total: 40 points)

Revised: Oct 26th

1. **(3 points)** No, the receiver cannot be absolutely certain that no bit errors have occurred. The checksum field is small (in fact considerably small) in comparison to the number of bits in the rest of the packet. Therefore multiple possibilities of the contents of packets will map to the same checksum. If the bit errors result in transforming the packet to another one with the same checksum, then such errors will go undetected.

2. **(3 points)** The possible sequence numbers are 501, 502, 503, 504.

3. **(3 points)** The possible ranges are [498,499,500], [499,500,501], [500,501,502] and [501,502,503].

4. **(3 points)** The possible ranges are [501,502,503], [502,503,504], [503,504,505] and [504,505,506].

5. **(3 points)** The possible ranges are [498,499,500], [499,500,501], [500,501,502] and [501,502,503].

6. **(3 points)** Shortcuts are useful for faster traversal on the DHT cycle. Without the shortcuts each query has to traverse along the cycle using only the next hop pointer. On average a query may end up traversing half of the cycle leading to huge delays. With shortcuts traversals of parts of the cycle can be avoided based on the key that is being searcher for.

   Suppose all the peers representing IDs from [0,63] are in a DHT. Suppose node 0 has a shortcut to node 32. For a query that maps to an ID of 33, rather than traversing around the cycle, the query can be forwarded to node 32, which will then be forwarded to 33 using the next pointer.

7. **(4 points)** Answer True or False to the following questions and briefly justify your answer:

   (a) True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at t_0. At t_1 ($t_1 \geq t_0$) the receiver ACKS 1, 2, 3. At $t_2 (t_2 \geq t_1)$ the sender times out and resends 1, 2, 3. At t_3 the receiver receives the duplicates and re-acknowledges 1, 2, 3. At t_4 the sender receives the ACKs that the receiver sent at t_1 and advances its window to 4, 5, 6. At t_5 the sender receives the ACKs 1, 2, 3 the receiver sent at t_2. These ACKs are outside its window.

   (b) True. By essentially the same scenario as in (a).

   (c) True. With a window size of 1, SR, GBN, and the alternating bit protocol are functionally equivalent. The window size of 1 precludes the possibility of out-of-order packets (within the window).

   (d) True. By the same reasoning as above. A cumulative ACK is just an ordinary ACK in this situation, since it can only refer to the single packet within the window.

8. **(4 points)**

$$\frac{95}{100} \times \frac{1500 \times 8}{200 \times 10^{-3}} bits/sec = 57 Kbps$$

9. **(3 points)**
   torrent: A group formed by the peers who are either downloading or uploading chunks of the same file. This also refers to the file with the .torrent extension that containts meta-information about the file.
   seed or seeder: A peer that is sending data to other clients.
   leecher: All users who are not seeders.

10. **(5 points)**

    (a) His first claim is possible, as long as there are enough peers staying in the torrent for a long enough time. Bob can always receive data through optimistic unchoking by other peers.

(b) His second claim is also true. He can run a client on each machine, and let each client do 'free-riding', and combine those collected chunks from different machines into a single file. He can even write a small scheduling program to let different machines only asking for different chunks of the file. This is actually a kind of Sybil attack in P2P networks.

11. **(6 points)**

(a) (a) -a: Show the state of all sockets, all routing table entries, or all interfaces, both physical and logical.
-P protocol: Limit display of statistics or state of all sockets to those applicable to protocol. The protocol can be one of ip, ipv6, icmp, icmpv6, igmp, udp, tcp, rawip.
-p: Show the net to media tables.
-p program: Show the PID and name of the program to which each socket belongs.

First, I ftp to the site *sunsite.unc.edu*. Then, I use nslookup to find that the IP address of sunsite.unc.edu is 152.46.7.80. And at last, I use "netstat -n | grep 152.46.7.80" to get the result below.
164.107.112.44.58515 152.46.7.80.21 5840 0 24840 0 ESTABLISHED

Local IP address is 164.107.112.44; local port number is 58515
Remote IP address is 152.46.7.80; remote port number is 21
Local window is 5840 bytes; local queue size is 0 byte
Remote window is 24840 bytes; remote queue size is 0 byte
The internal state "ESTABLISHED" means the connection has been established

(b) -U: Send UDP packets instead of ICMP (ICMP6) packets.

-v: Verbose output. List any ICMP (ICMP6) packets, other than replies from the target host.

-a: ping all of the addresses, both IPv4 and IPv6, of the multi-homed destination.

-s: Send one datagram per second and collect statistics.

-I interval: Turn on the statistics mode and specify the interval between successive transmissions. The default is one second.

There is another version of ping. The bellow answers are also accepted.

-U: Print full user-to-user latency (the old behavior). Normally ping prints network round trip time, which can be different f.e. due to DNS failures.

-s packetsize: Specifies the number of data bytes to be sent The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

-I interface address: Set source address to specified interface address. Argument may be numeric IP address or name of device. When pinging IPv6 link-local address this option is required.