CSE 473 – Introduction to Computer Networks *Roch Guérin*

Midterm Exam

Your Name: *10/14/2014*

1) (**5 points**). A user in St. Louis is connected to the Internet via a 4 Mbps ($4x10^6$ bits/sec) DSL link (download speed) and is retrieving a webpage from a server in China. The page is 125 kbytes and contains references to ten (10) images that are each 250 kbytes. The one-way propagation delay is 50 ms and the DSL link is the bandwidth bottleneck for this connection.

   (2 points) Approximately how long does it take for the page (including images) to appear on the user's screen, assuming she uses a single persistent HTTP connection to access the server (queueing and transmission delays on links other than the DSL access link are negligible, and you can ignore the impact of TCP's ramp-up)?

   (2 points) How long would it take if the user's browser used instead two persistent HTTP connections (the two connections are opened in parallel; one connection is used to request the page, and the two connections would then each request five images)?

   (1 point) Approximately how big (in bytes) should the buffer at the access router connected to the user's DSL link be to ensure there are no packet losses (assume that packets arrive from the Internet much faster than the 4 Mbps download speed of the DSL link)?

2) (**10 points**). A private network that uses the 10.1.0.0/16 private address space sits behind a NAT router (NAT1) that has been allocated the public IP address 45.6.7.1. Assume that a local host in that network with address 10.1.10.1 is accessing a remote web server with public address 194.34.45.67. Headers of local packets originating from host 10.1.10.1 and destined for the web server have the following format in the private network behind NAT1:

<src_addr>	<dst_ addr>	<src_port>	<dest_port>
10.1.10.1	194.34.45.67	4567	80

The NAT (NAT1) already has the following entries:

Internal: <local_addr><local_port>	**External**: <external_port>
<10.1.11.2><4567>	<4567>
<10.1.10.1><3333>	<3333>
<10.1.10.1><4444>	<5555>

(3 points) What is (i) a possible entry in the forwarding table of NAT1 to accommodate the connection from 10.1.10.1 to web server 194.34.45.67, and (ii) what would then be the headers of packets from this connection when forwarded by NAT1 into the public Internet?

(2 points) Assume now that host 10.1.10.1 decides to open a second parallel connection to the web server. Packet headers for this connection are of the following form in the private network behind NAT1:

<src_addr>	<dst_ addr>	<src_port>	<dest_port>
10.1.10.1	194.34.45.67	4568	80

Does NAT1 need to create a new entry for this connection, or can it reuse the previous one? Justify your answer.

Host 10.1.10.1 is also engaged in a videoconference with another host in a different private network behind another NAT (NAT2). The private address of that other host is 10.2.2.2. Packet headers for the connection from host 10.1.10.1. to host 10.2.2.2 have the following formats on the internal network of host 10.1.10.1 (internal network 1), in the public Internet, and in the internal network of host 10.2.2.2 (internal network 2):

Packet headers	<src_addr>	<dst_ addr>	<src_port>	<dest_port>
Internal Network 1	10.1.10.1	53.3.4.7	3456	2345
Public Internet	45.6.7.1	53.3.4.7	5889	2345
Internal Network 2	45.6.7.1	10.2.2.2	5889	1345

(3 points) Identify the entries for this connection in the forwarding table of the two NATs?

(2 points) How would the header of a packet for the return connection (from 10.2.2.2 to 10.1.10.1) look like when traversing the public Internet?

3) (**10 points**) Consider a circular DHT with 64 nodes numbered 0, 1,..., 63. Node *i* handles keys with hash values in the range $i \times 2^{28}$ to $(i+1) \times 2^{28}-1$, and has routes to nodes *i*+1, *i*+4, *i*+8 and *i*+32 (where addition is modulo 64).

(5 points) What is the maximum number of hops required to get from one server to another in this DHT?

(2 points) Suppose that node 32 receives a *get* request with a key string of "Led Zeppelin" that *hashes* to 8,321,499,137. How many hops will this request go through?

(3 points) Assume now that nodes can cache key-value pairs. Suppose node 4 cached the key-value pair for the key "Led Zeppelin" but is the only node to have done so, *i.e.,* the only other node that has the key-value pair is the node that has the hash value of the key in its range. Assuming that the next query for key "Led Zeppelin" arrives at a randomly selected node, what are the odds that node 4 will be handling it, *i.e.,* what are the odds that the next query for "Led Zeppelin" is handled by node 4?

4) (**10 points**) Two hosts, A and B, are connected by a direct point-to-point link of speed 100 Mbps (10^8 bits/sec), and rely on a basic go-back-N protocol to ensure reliable transmissions between them. The link can lose/corrupt packets, but not reorder them. A and B use 1,250 bytes packets, including header and payload, and as usual the protocol uses a single shared timer. The window size is **<u>61 packets</u>**.

(3 points) What is the maximum possible RTT between A and B that will allow communication at the maximum rate of 100 Mbps in the absence of packet errors or losses? For simplicity, assume that the RTT measures the time between the transmission of the **<u>last</u>** bit of a packet until the ACK for that packet is received.

(4 points) Assume from now on an RTT of 10 ms (5 ms each way) and a time-out value of 11 ms. Consider a scenario where at time $t = 0$, A starts transmitting packets 1 to 50, packet 1 is lost, so that A eventually retransmits packets 1 to 50. Packet 50 is then lost during this retransmission. When will packet 50 be eventually received at B (assume that no ACKs are lost)?

(3 points) Assume next that at time t the state at A is as follows: sendBase = 25, nextSeqNum = 63, timerExpiration = 5 ms, where sendBase is the sequence number of the oldest unacknowledged packet, nextSeqNum is the sequence number to be used in the next packet transmission, and timerExpiration tracks how much time is left before the time-out timer expires. An ACK with sequence number 32 is received at t + 1 ms, and at t + 7 ms the application at A writes five (5) packets worth of new payload. What is the state at A, *i.e.,* values of sendBase, nextSeqNum, and timerExpiration, at time t + 9 ms, and what packets are still in A's re-send buffer?

5) (**15 points**) Consider a TCP sender that sends MSS size packets with MSS = 1000 bytes. The sender needs to upload a file whose size corresponds to the aggregate payload of a total of 2048 MSS size packets. The sender starts packet transmissions at $t = 0$ (after the TCP handshakes completes) with *ssthresh* = *rcvWindow* = *64 MSS*. The sender is connected to the Internet through a 10 Gbps (10^{10} bits/sec) link, but the bottleneck link on the connection's path is 100 Mbps (10^8 bits/sec). The connection's RTT is mostly constant at 40 ms, *i.e., EstimatedRTT* ≈ 40 ms and *DevRTT* ≈ 0 ms. Assume that the TCP connection <u>does not use delayed ACKs</u>.

(5 points) How long approximately will it take the sender to upload half the file, *i.e.,* transmit 1024 MSS size packets, in the absence of packet losses?

(6 points) The sender uses TCP Reno. Assume that the connection experienced no losses until packet number 1984 that is lost at time *t*. At that time, *cwnd* had long reached it maximum value. How long approximately will it take from *t* onward before the sender knows that the entire file has been successfully uploaded to the receiver, *i.e.,* receives an ACK acknowledging the last packet? For simplicity, assume that the lost packet 1984 is the first to be transmitted in a new batch (window) of *cwnd* packets sent by the sender.

(4 points) Assume next that the sender is using TCP Tahoe instead of TCP Reno. How does this change the answer?

6) **15 points)** A sender uses TCP Reno (without delayed ACKs) with *cwnd* = *rcvWindow* = 64 kbytes (2^{16} bytes), an MSS of 1 kbytes (2^{10} bytes), an RTT of 104.858 ms, and is transmitting over a path with a bottleneck link of speed 10 Mbps (10^7 bits/sec). The router connected to that link has a very small buffer, so that it will lose packets as soon as the aggregate incoming transmission rate barely exceeds the link capacity.

(5 points) How many parallel connections can the sender safely open to increase its aggregate transmission rate without incurring losses at the bottleneck link? You can assume that the receiver has enough memory to allocate a rcvBuffer of 64 kbytes to each new connection.

(10 points) Assume that the sender decides to open 4 parallel connections. What will approximately be the steady-state aggregate transmission rate it manages to realize? Note that if connections experience losses, you need to identify what their overall rate would be, accounting for the up and down pattern of transmission rate the connection would experience