

Midterm Exam Review

10/9/2014

1. (10 points). A user in Chicago, connected to the internet via a 100 Mb/s (b=bits) connection retrieves a 250 KB (B=bytes) web page from a server in London, where the page references three images of 500 KB each. Assume that the one way propagation delay is 75 ms and that the user's access link is the bandwidth bottleneck for this connection.

Approximately how long does it take for the page (including images) to appear on the user's screen, assuming non-persistent HTTP using a single connection at a time (for this part, you should ignore queueing delay and transmission delays at other links in the network)?

How long does it take if the connection uses persistent HTTP (single connection)?

Suppose that user's access router has a 4 MB buffer (B=byte) on the link from the router to the user. How much delay does this buffer add during periods when the buffer is full?

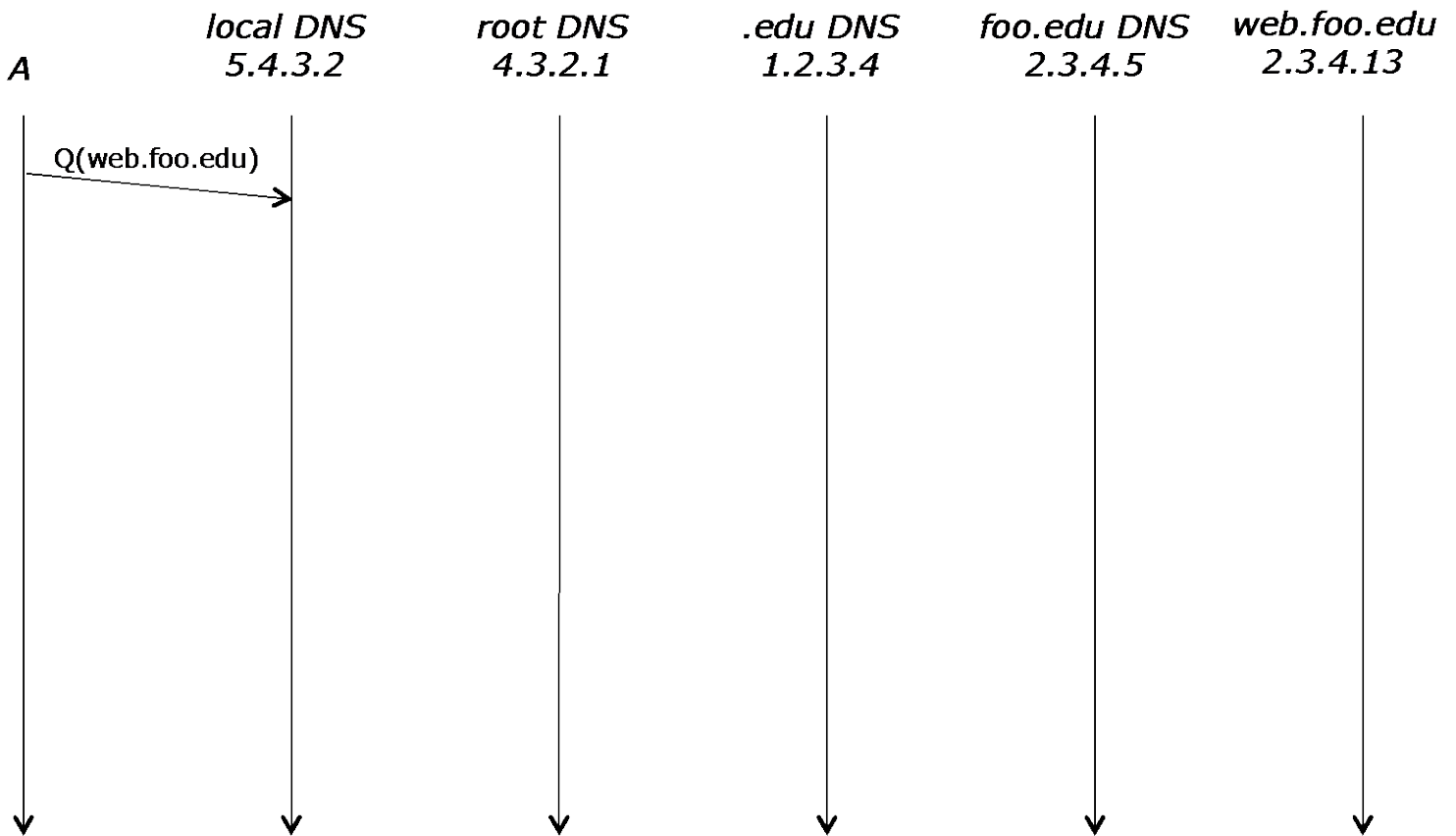
2. (10 points). Consider a circular DHT with 27 nodes numbered $0, 1, \dots, 26$, where the nodes cache key-values pairs for 60 seconds before discarding them. Assume that each node i handles keys with hash values in the range $100i$ to $100i+99$. Also, assume that node i has routes to nodes $i+1$, $i+3$ and $i+9$ (where the additions are modulo 27). What is the maximum number of hops required to get from one server to another in this DHT?

Suppose that some client sends a *get* request to node 20 with a key string of "candy corn", and that $hash(\text{"candy corn"})=1853$. If the request is handled by exactly 4 nodes in the DHT (including 20), which nodes are they?

Suppose node 13 has an entry for "candy corn" in its cache but no other nodes do. Now assume that every other node receives a *get* request for "candy corn". How many of these requests does 13 handle? (Ignore any new cache entries that might be created in the process of handling these requests.)

Repeat your answer to the last question assuming that node 2 has the cached entry for "candy corn". (That is, how many of the requests would 2 handle in this case?)

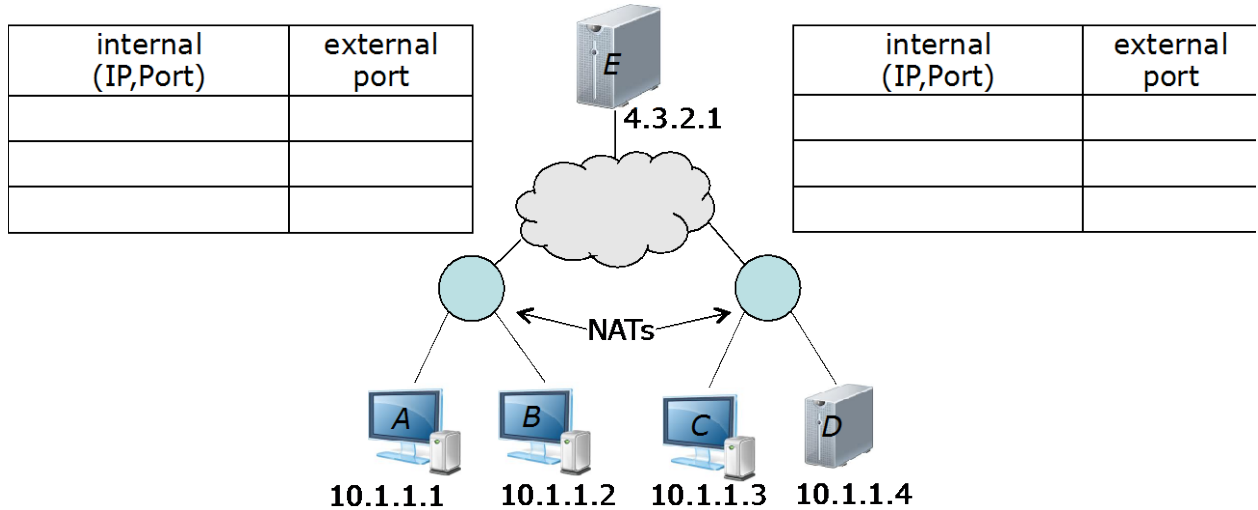
3. (10 points) The diagram below shows a DNS query from a host *A* to its local DNS server. The IP addresses of all hosts are shown in the diagram. The label "*Q(web.foo.edu)*" specifies the query string. Complete the diagram showing all packets sent to resolve the name and continuing through the opening of a TCP connection to the web site and the first GET request. All arrows that represent DNS queries should have a label of the form "*Q(a.b.edu)*" and replies should have a label of the form "*R(b.edu=2.3.7.11)*". TCP connection packets should be labeled with the appropriate flags and HTTP packets with the request type. Assume that the local DNS server performs recursive processing and has nothing in its cache, while the others perform iterative processing. You may assume that all queries and responses are for *A* records.



List all the mappings in the local DNS server's cache after the query has been processed.

List the mappings in the local server's cache if the *.edu* server did recursive processing rather than iterative.

4. (15 points). The diagram below shows two residential networks with routers that implement NAT and a remote server with a public internet address



The packet header diagrams at right are for a packet from a host in the left-hand network, going to the server. The first shows the header when the packet arrives at the router, the second shows it when the packet leaves the router. Add an entry to the left-hand NAT table that is consistent with these two packet headers. What is the public IP address of the left-hand router?

src adr	dest adr	src port	dest port
10.1.1.1	4.3.2.1	5555	3333
3.7.5.7	4.3.2.1	8888	3333

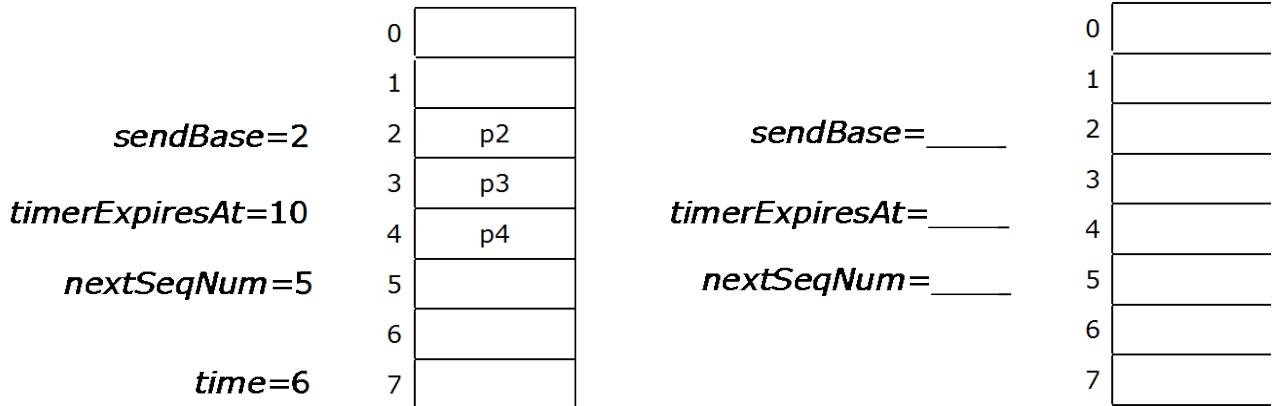
The three header diagrams at right are for a packet from a host in the right-hand network, going to a host in the left-hand network. Fill in the blank fields. Add entries to the two NAT tables that are consistent with this sequence of packet headers. What is the public IP address of the right-hand router?

src adr	dest adr	src port	dest port
10.1.1.4		1212	7878
5.3.5.2		5454	7878
5.3.5.2	10.1.1.2	5454	6565

In the diagrams at right, fill in the header fields that would be used by a response to the last packet, (the response goes from the left hand network to the right).

src adr	dest adr	src port	dest port

5. (10 points) The diagram at left below shows the state of the sending side of a communication session using a go-back- N protocol with cumulative acknowledgements, a window size of 4 and a retransmission timeout of 25. The array represents the “re-send” buffer and entry in the buffer represents a packet with its sequence number. As usual for go-back- N , we are using a single shared timer. Its expiration time is shown in the diagram. Assume that the channel may lose packets but never re-orders them.



Suppose that at time 7, the application passes us a new payload to be sent and at time 12, we receive an ack with sequence number 3.

List all the packets sent by the sender between times 6 and 13, including repeats (if any).

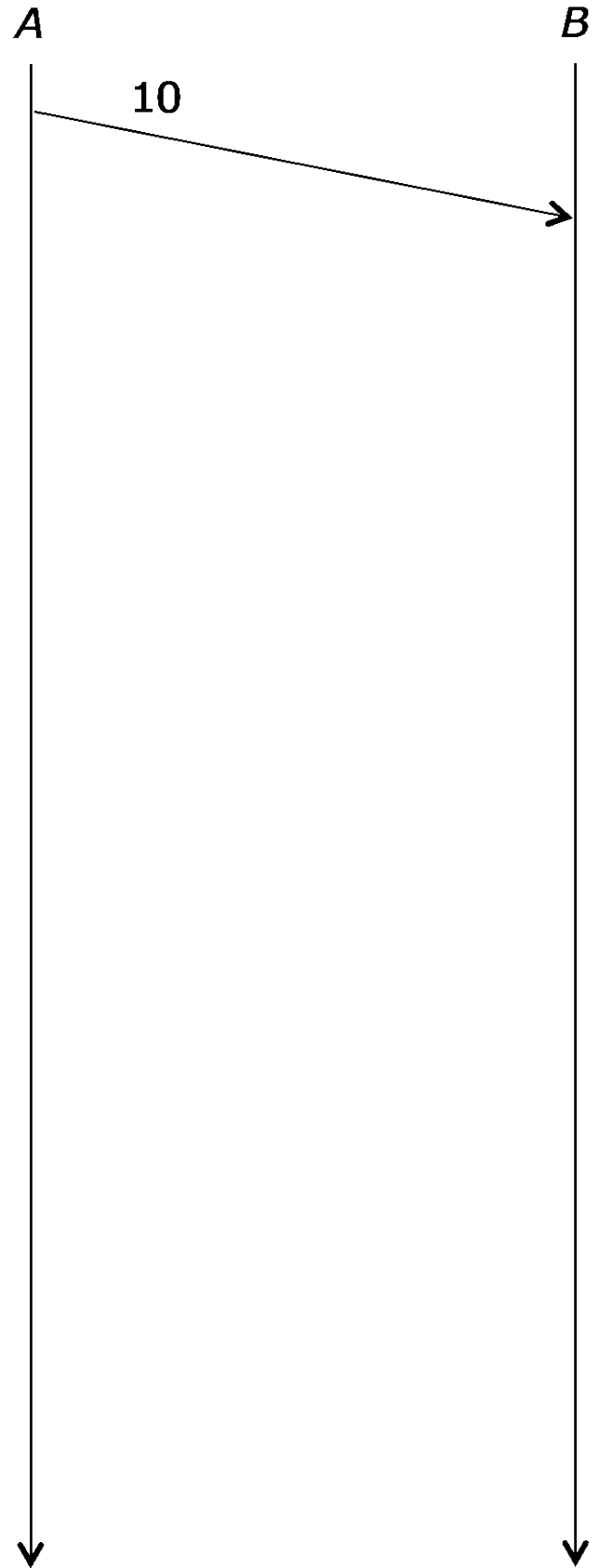
Show the state of the sender at time 13, in the right-hand diagram.

At time 13, how many additional packets can be sent before an ack is received?

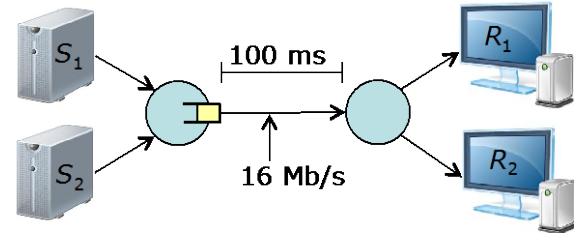
6. (10 points) Consider a TCP sender that sends 10 bytes per packet. Suppose that the sender transmits six packets one right after another (including the one shown at right). Show the packets and the corresponding acks that TCP would return in a typical situation. Label the data packets with the sequence number and the ack packets with the ack number.

If the second of the acks is lost, will the sender retransmit any packets? Why or why not?

Continuing the example, suppose that the next two packets sent arrive out of order and that the sender transmits four more packets after these and that the second packet in this group of four is lost. Extend the diagram to show these packets and the resulting acknowledgements. You need not show the retransmission of the lost packet.



7. (10 points) The diagram at right shows two TCP senders at left and the corresponding receivers at right. Both senders use TCP *Tahoe*. Assume that the MSS is 1 KB, that the one-way propagation delay for both connections is 100 ms and that the link joining the two routers has a bandwidth of 16 Mb/s. Let $cwnd_1$ and $cwnd_2$ be the values of the senders' congestion windows. What is the smallest value of $cwnd_1 + cwnd_2$ for which the link joining the two routers stays busy all the time?



Assume that the link buffer overflows whenever $cwnd_1 + cwnd_2 \geq 600$ KB and that at time 0, $cwnd_1 = 500$ KB and $cwnd_2 = 100$ KB. Approximately, what are the values of $cwnd_1$ and $cwnd_2$ one RTT later? Assume that all losses are detected by triple duplicate ACKs.

At this point, what are the values of $ssthresh_1$ and $ssthresh_2$?

After 20 more RTTs, approximately what are the values of $cwnd_1$ and $cwnd_2$?

Approximately, how many more RTTs before $cwnd_1 + cwnd_2 \geq 600$ KB again?