Homework 2, Victor Chu **Due: October 21, 2016**

Question 1

(a) response time = time for connections + transmission delay

RTT = 100 ms

time for connections = num of objects × RTT × 2 = 6 × 2 × 100 ms = 1200 ms

Since this is a Non-Persistent Connection, we have to establish a TCP connection for each object. We have to set up one TCP connection (1 RTT) for each time we send over an object. Since we have 6 objects, and we have to have 2 RTT for each object (one for TCP and one for sending over the object), we get 12 RTT for the time for connections.

length of link = 50 Mbps or 50,000 Kbps

webpage object = 7 Kbits

images = 5 × 2.5 Kbits = 12.5 Kbits

total length of pkt = webpage object + images = 19.5 Kbits

transmission delay = $\frac{19.5Kb}{50{,}000Kbps}$ = .00039 secs

response time = 1200 ms + .390 ms = 1200.390 ms

(b)

time for connections = 800 ms ((1 RTT for TCP + 1 RTT for sending) * 4 objects = 8 RTT ; the webpage is sent first, then the 5 images are sent in parallel)

transmission delay = $\frac{(7Kb+2.5Kb+2.5Kb+2.5Kb)}{50{,}000Kbps}$ = 0.00029 secs = .29 ms

response time = 600 ms + .29 ms = 800.29 ms

(c) time for connections = 7 × 100 ms = 700ms ((1 tcp connection + 6 objects) × RTT)

response time = 700 ms + .390ms = 700.390 ms

We set up one tcp connection because it is a persistent HTTP connection.

(d) time for connections = 300ms (1 tcp connection RTT + 6 objects sent (the 7 Kb first, then the 5 2.5 Kb all in one RTT)

new transmission delay = $\frac{(7+2.5*5)Kb}{50{,}000Kbps}$ (have to put on all packets) = 0.00039 secs = .39 ms

response time = 300 ms + .39 ms = 300.39 ms

(e) Non-Persistent HTTP connections are useful in the cases where parallel connections are allowed and pipelining is not allowed in Persistent connections. Moreover, it is preferred in cases where there are less embedded objects in the HTML code, since we only have to get one HTML object. Lastly, I believe that Non-Persistent connections are more secure than Persistent connections, so they are used in situations where you want to connect to a website more securely.

Question 2

(a)

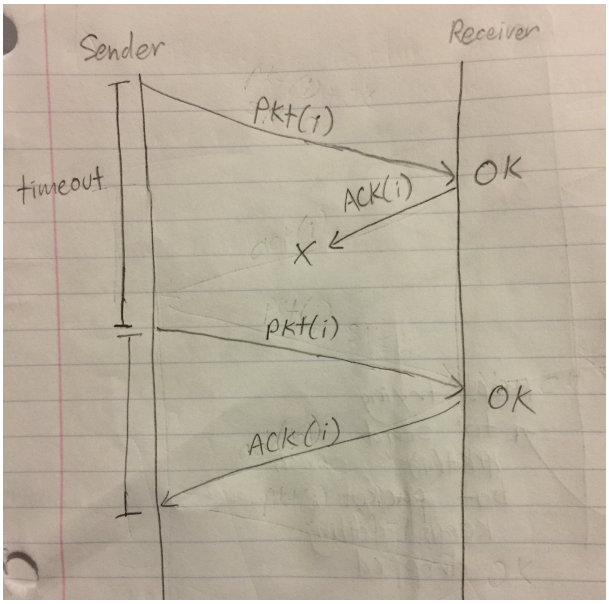
1. a.root-servers.net
2. a.edu-servers.net
3. ns3.wesleyan.edu

(b)

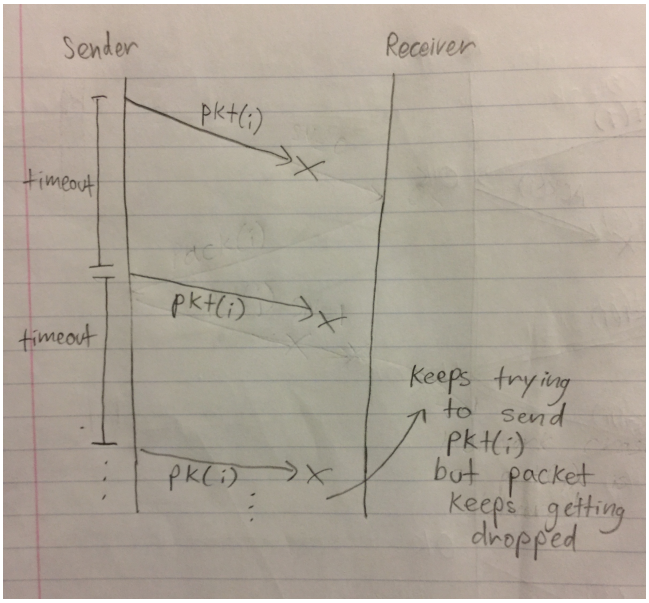
1. a.root-servers.net
2. f.edu-servers.net
3. usw2.akam.net
4. auth-ns2.csail.mit.edu

Question 3

(a)



(b)



Question 4

(a) True, it is possible because the receiver reacknowledges already received packets below the current window base. This is necessary in the case that the sender doesn't receive an ACK for an already received packet on the receiver's side or else the sender cannot move forward.

(b) False, it is not possible because GBN requires a cumulative acknowledgement from packet with sequence number n, which tells us that all packets with a sequence number up to and including n have been correctly received at the receiver, i.e. all packets in the current window send ACKs back to the sender. Only after this cumulative ACK can the window slide and move on to the next packets. Therefore, we cannot receive ACKs for packets that are outside the current window.

Question 5

(a) In the second segment sent from Host A to Host B, the sequence number is 289, the source port number is 503, and the destination port number is 80.

(b) If the first segment arrives before the second, then the acknowledgement number is 289, the source port number is 80, and the destination port number is 503.

(c) If the second segment arrives before the first, then the acknowledgement number is 249 because Host B is still waiting for bytes 249 - 288 to arrive.

(d)