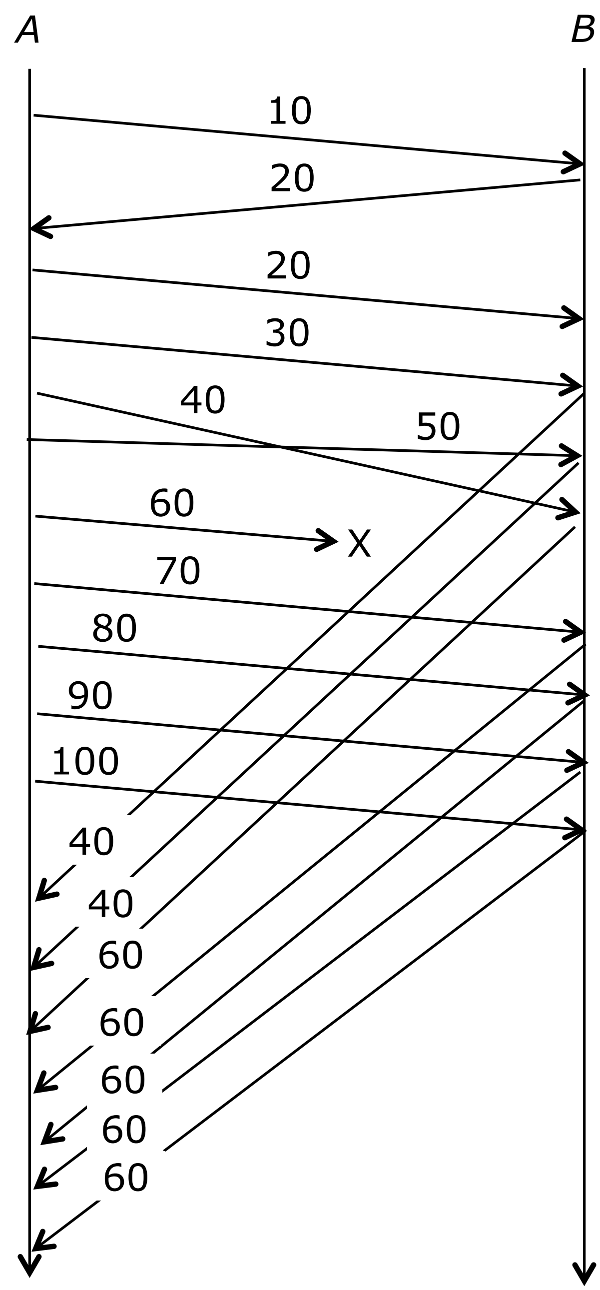
CSE 473 – Introduction to Computer Networks	*Instructor: John DeHart*

Midterm Exam

Your Name: *10/14/2015*

1) (10 points). The diagram at right shows a TCP segment being sent from host *A* to host *B* and an ACK being returned. The numbers on the arrows are the sequence numbers of the data segments and the ACK numbers. Suppose that after receiving the ACK with ack number 20, *A* sends packets with sequence numbers 20, 30, 40, 50, 60, 70, 80, 90 and 100. Some time later, it receives ACKs with sequence numbers 40, 40, 60, 60, 60, 60, 60. (Assume that *A* sends no additional data segments in the meantime.) Complete the diagram in a way that is consistent with the given information and what you know about the way TCP behaves.



40 and 50 cross

No ack for pkt with seq=20, cumulative (delayed ack) for pkt with seq=30

Seq=60 is lost

Specifically says "some time later it receives ACKs" so the 40 and 60 ACKs should all arrive after the sending of pkt with seq=100

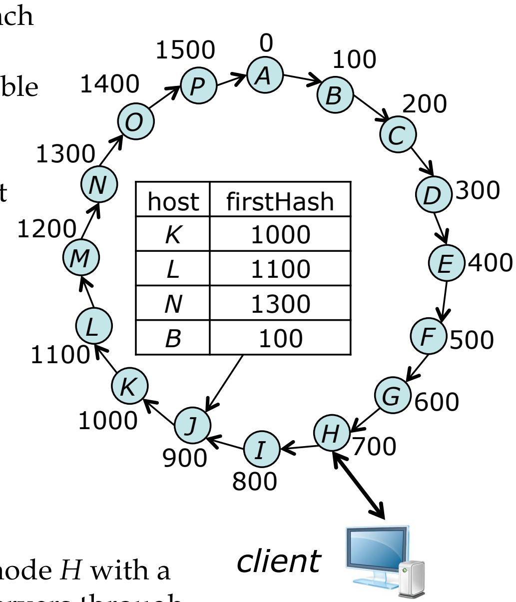
What sequence number would you expect to see in the next packet sent by *A*?

60

What ACK number would you expect in the next ACK? You may assume that all packets sent by *A* carry 10 bytes of data.

110

2) (12 points). The diagram at right shows a DHT with 16 nodes. Each node is labeled with the first value in its range values (so for example, *B* is responsible for hash values 100-199. The routing table for node *J* is shown in the figure. Note that *J* has routes to the node that is 1 hop away, the one that is 2 hops away, the one that is 4 hops away and the one that is 8 hops away. Assume that all nodes have routing tables that are configured similarly.



Fill in the routing table for node H:

host	firstHash
I	800
J	900
L	1100
P	1500

Suppose the client shown in the diagram sends a get request to node *H* with a key string of "flapjack", and that *hash*("flapjack")=513. List the servers through which this request would pass, assuming that the key string does not appear in any node's cache.

It would go through nodes H, P, D and F before returning to H and then the client.

What servers would the request pass through if the key string appears in node *M*'s cache?

It would go through the same nodes H, P, D and F before returning to H and then the client.

What servers would it pass through if the key string appears in node *D's* cache?

H, P and D before returning to H and then the client.

Suppose that "flapjack" is requested frequently. Specifically, each DHT node receives a get request for "flapjack" about once per second. If the system is operated without caches, how many requests per second must the "responsible server" process?

16 requests per second. The system as a whole receives 16 requests per second for flapjack. The responsible server will have to process all of them.

Again, each DHT node receives a get request for "flapjack" about once per second. If caching is enabled, and each cache entry expires 60 seconds after being placed in the cache, approximately how often does the responsible server receive a get request from another server? (Hint: how many other servers send directly to the "responsible server"?)

There are 4 servers that send packets directly to the responsible server. These 4 servers will each send the responsible server a new request every 60 seconds, just after their cache entry expires. So the responsible server will receive a request about once every 15 seconds from another server (in addition to the one per second that it receives directly from clients).

3) (15 points). Consider a pipelined, reliable transport protocol that uses go-back-*N* with cumulative acknowledgments. Assume that timeouts trigger retransmissions (duplicate ACKs do not) and that the receiver does not maintain any receive buffer. If the one-way delay between the sender and receiver is 50 ms and every packet is 10,000 bits long, how big must the window (N) be to allow the sender to send at a steady rate of 1 Gb/s under ideal conditions?

RTT=.1 second, so a 1 Gb/s link sends 100M bits per RTT or 10K packets per RTT. So the window size must be at least 10,000 pkts to support a 1 Gb/s rate after we have sent the first packet. So the total window size would need to be 10001.

For all of the following questions, assume the bottleneck link rate is 1Gb/s.

Suppose that approximately one packet in 100,000 is lost. If the sender uses a timeout of 500 ms and a window size of 20,000 packets, how often does sender experience a timeout? How many packets will it retransmit when a time out occurs?

) The sender can still only send 10K packets per RTT. After each loss, it takes half a second for the sender to detect the loss and all packets sent in that half second are effectively wasted (since the receiver discards them in go-back-N). But the window size limits the number of packets sent following the lost packet to 20K. So immediately after each loss, the sender sends 20K packets in 0.2 seconds, pauses for 0.3 seconds then re-sends the first 20K packets before sending another 60K, at which point it loses another packet. So, the sender experiences a timeout every 1.3 seconds.

Assume we have a loss: t = 0, total pkts sent this cycle = 0

Send 20K pkts: t = 0.2 sec, total pkts sent this cycle = 20K

Pause: t = 0.5 sec, TIMEOUT, total pkts sent this cycle = 20K

Resend original 20K pkts: t = 0.7 sec, total pkts sent this cycle = 40K

Send 60K pkts: t = 1.3 s, total pkts sent this cycle = 100K

Loss: t = 1.3 sec (total ***new*** *packets sent since last loss = 80K, resends = 20K, total pkts sent = 100K)*

Repeat

Assume that after the connection starts at time 0, the 100,000-th packet (call it *p*) is lost. At what time was *p* sent by the sender?

At time 1 second.

At what time does the sender re-transmit *p*?

1.5 seconds

What happens to the packets sent between the time *p* is sent the first time and the time it is retransmitted?

They are discarded by the receiver and retransmitted by the sender

Estimate the throughput for this connection, assuming one packet in 100,000 is lost.

The receiver gets 80K new packets every 1.3 seconds, so the throughput is (8/13) Gb/s which is about 620 Mb/s. Or you could try to solve, $T_{succ} = (1 + (a-1)q/(1-q)) * t_{pkt}$*. And use* T_{succ} *to find throughput. TCP throughput or RDT 3.0 throughput equations do not work here.*

4) (12 points). A user in St. Louis, connected to the internet via a 20 Mb/s (b=bits) connection retrieves a 250 KB (B=bytes) web page from a server in Seattle, where the page references 4 images of 1 MB each. Assume that the one way propagation delay is 25 ms.

   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 include transmission delay on the user's access link, but you may ignore delays at other network links)?

   *5 * 2 * RTT + transmission time*

   *5*2*50 ms + (0.250*8 + 4*8*1)Mb/(20 Mb/s) = 500 ms + 1.7 sec = 2.2 seconds*

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

   *3 * RTT + transmission time*

   3 50 ms + 1.7 sec = 1.85 seconds*

   Suppose that the path from the server to the user passes through a 1 Gb/s link at a router *R*, and that the rate at which packets arrive at router *R* that must be sent on this link is 450,000 packets per second. If the average packet length is 2,000 bits, what is the average queueing delay at this link? You may use the infinite queue approximation.

   *W = E[N] * E[L]/c, E[N] = I/(1-I), I = (450,000 pkts/sec) * (2000 bits/pkt) / (1Gb/s)*

   I = (900 Mb/s)/(1 Gb/s) = 0.9, so average queue length E[N] = I/(1-I) = .9/.1 = 9 packets

   time to send a packet = (2000 bits)/(1Gb/s) = (2x10^3 bits)/(1x10^9 bit/sec) = 2x10^{-6} sec = 2us

   *average delay = 9*2 μs = 18 μs*

5) (6 points) Label each of the following protocols by layer: Application, Transport or Network

   IP : *Network*

   DHCP : *Application*

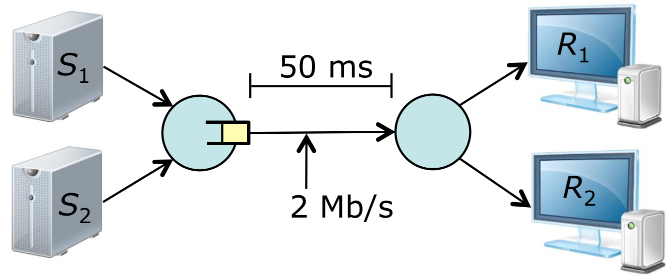
   UDP : *Transport*

   DNS : *Application*

   ICMP : *Network*

   TCP : *Transport*

6) (15 points) The diagram at right shows two TCP senders at left and the corresponding receivers at right. Both senders use TCP Reno. Assume that the MSS is 1 KB, that the one-way propagation delay for both connections is 50 ms and that the link joining the two routers has a bandwidth of 2 Mb/s. Let $cwnd_1$ and $cwnd_2$ be the values of the senders' congestion windows and assume that $cwnd_1 = cwnd_2$. What is the smallest value of $cwnd_i$ for which the link joining the two routers stays busy all the time?



We need 200 Kbits per RTT to keep the link busy, or 100 Kbits per sender. That means 12.5 KB

Assume that the link buffer overflows whenever $cwnd_1+cwnd_2 \geq 36$ KB and that at time 0, $cwnd_1=12$ KB and $cwnd_2=24$ KB. Approximately, what are the values of $cwnd_1$ and $cwnd_2$ one RTT later? Assume that all packet losses are detected by a triple duplicate ack.

It will detect a loss and immediately retransmit the lost packet and transition to fast recovery setting

$ssthresh_1$ = 6KB $ssthresh_2$ = 12KB $cwnd_1$ = 9KB $cwnd_2$ = 15KB *(Acceptable answer)*

About one RTT later it will receive the ACK for the retransmitted packet and transition back to congestion avoidance setting

$cwnd_1$ = 6 KB and $cwnd_2$ = 12 KB (Acceptable answer)

Either here or next part need to go back to congestion avoidance.

How many RTTs pass before $cwnd_1+cwnd_2=36$ again? What are the values of $cwnd_1$ and $cwnd_2$ at this point?

After 9 more RTTs, we have $cwnd_1$=15 KB and $cwnd_2$=21 KB.

(Or 10 more RTTs if the previous answer was the fast recovery answer)

Approximately, how many RTTs pass (in total) before $cwnd_2 - cwnd_1 < 2$ KB?

We have one more RTT in which we do our fast recovery, so after a total of 11 RTTs, the difference is 3 KB (7.5KB and 10.5KB) and it remains 3 KB for another 9 RTTs when the buffer fills again, at which point it becomes 1.5 KB. So approximately 20-22 RTTs pass before the difference in the congestion windows drops below 2 KB.