CSE 473 – Introduction to Computer Networks *Roch Guérin*

Midterm Exam

Your Name: *10/23/2017*

Five (5) Problems for a total of 100 points

1) **[25 points]** A New York hospital has deployed a system that allows doctors to view patient records and X-rays remotely using a web browser. Patient records and X-Rays are stored in an off-site data center reachable from the hospital over a dedicated 1 Gbps (10^9 bits/sec) link with a roundtrip propagation of 2ms (0.002 sec). Because connectivity is over a dedicated link, the hospital and the data center have configured their TCP stacks to bypass slow-start entirely and start immediately with a full *cwnd*.

   a. **[5 points]** Assuming that both *cwnd* and *rcvWindow* are set to 128 kbytes (2^{17} bytes), *i.e.,* they are scaled by a factor 2, what is the maximum achievable sustained transfer rate over a TCP connection between the hospital and the data center? Explain your answer.

   *The transmission time of a full cwnd at 1Gbps is 2^{17} bytes*8/10^9 = 1.05ms, which is smaller than the 2ms roundtrip time between the hospital in the data center. Hence, the maximum transfer rate of a TCP connection is limited to one cwnd per RTT or 2^{17} bytes*8/0.002 = 524.3 Mbps, i.e., about half the link capacity.*

   b. **[5 points]** Assume next as well as in subsequent questions that *cwnd* and *rcvWindow* have instead been scaled by a factor 8 and have, therefore, a maximum value of 512 kbytes (2^{19} bytes). What is now the maximum transfer rate of a TCP connection? Explain your answer.

   If cwnd is 512 kbytes, the transmission time of a full cwnd on a 1 Gbps link is four times that of a 128 kbytes cwnd or about 4.2ms. Because this exceeds the link's RTT of 2ms, the transmission rate will be limited to the link speed of 1 Gbps.

   c. **[10 points]** Access to patient records is done over `http 1.0`, *i.e.,* non-persistent `http`, but download of objects (X-rays) embedded in a patient's record is done using parallel TCP connections, one per object. Consider a patient record that is 256 kbytes (2^{18} bytes) with two (2) embedded X-rays, each 2 Mbytes (2^{21} bytes) in size. How long will it take for a physician to download the patient record and its two (2) X-rays? Explain your answer.

   *Download of the patient record requires two RTTs plus the transmission time of the record, i.e., 4ms + 2^{18} *8/10^9 = 6.1ms. This is immediately followed by opening two new TCP connections, one for each X-ray. The setups of those two connections happen in parallel and take 2ms. Once the connections are setup, they both start requesting data for their respective X-rays, but will be sharing the link bandwidth. So after the TCP connections are established, the download of the X-rays takes one additional RTT plus the time it takes to transmit two X-rays or 2^{22} bytes of data over a shared 1 Gbps link, i.e., overall 4ms + 2^{22}*8/10^9 =37.55ms to download both X-rays. Hence the total download time for the patient's record and the two X-rays is 43.65ms.*

   d. **[5 points]** How would the answer change if the hospital and its data center had instead used `http 1.1`, *i.e.,* persistent `http`? Explain your answer.

   If `http 1.1` *had instead been used, we would have avoided the additional RTT required to setup separate TCP connections to download the two X-rays, which would have subtracted 2ms from the download time, i.e., for a total download time of 41.65ms.*

2) **[20 points]**. Private network A uses the 10.1.0.0/16 private address space and connects to the Internet using a PAT/NAT router (Ra) with public IP address 65.5.4.15. Private network B also uses the 10.1.0.0/16 private address space and connects to the Internet using a PAT/NAT router (Rb) with public IP address 142.3.7.45. Assume a host with address 10.1.1.1 in private network A has four parallel TCP connections to port 80 on a host with the same private address 10.1.1.1 in private network B. The four connections have source port numbers 50551 to 50554.

a. **[3 points]** What is the IP address carried in the destination address field of packets leaving host 10.1.1.1 in private network A on those 4 connections, and similarly what is this destination address when those packets leave router Ra? Justify your answer.

In both cases, those packets' destination address is 142.3.7.45, the public address of Rb, and consequently the public address for any host in private network B.

b. **[5 points]** Describe possible entries present in Ra and Rb as a result of those four connections, and state explicitly how many entries there are in each router.

If none of the four port numbers used by host 10.1.1.1 in private network A have been previously assigned, Ra would have four entries mapping those four port numbers 50551 to 50554 to destination address 10.1.1.1 and the same destination port numbers. For example router Ra would have an entry mapping any packet with destination port 50551 to destination address 10.1.1.1 and the same destination port number. For any of those port numbers already in use, Ra would use alternate port numbers, and swap them to replace the original (source) port numbers in outgoing packets.

In Rb, there would only be one entry mapping (incoming) destination port 80 to port 80 on host 10.1.1.1. Note that because port 80 is a well-known port (for web service), it must be preserved when leaving the private network, i.e., router Rb will typically have a static mapping for (incoming) port 80 directed to port 80 on host (web server) 10.1.1.1.

c. **[5 points]** Using notation consistent with your answer in the previous question and focusing on the connection with source port number 50551 from host 10.1.1.1 in private network A, give the source and destination addresses and port numbers for packets that (i) leave host 10.1.1.1 in private network A, (ii) leave router Ra, (iii) reach host 10.1.1.1 in private network B; and conversely for packets in the reverse direction from host 10.1.1.1 in private network B to host 10.1.1.1 in private network A (for a total of six (6) distinct packet headers). Use the format <destAddr, destPort; srcAddr, srcPort> in your answer.

Packet headers are as follows:

Out of host 10.1.1.1 in private network A: <142.3.7.45,80;10.1.1.1,50551>
Out of Ra leaving private network A: <142.3.7.45,80;65.5.4.15,50551> (assumes port 50551 was unused)
Arriving at host 10.1.1.1 in private network B: <10.1.1.1,80;65.5.4.15,50551>

Out of host 10.1.1.1 in private network B: <65.5.4.15,50551;10.1.1.1,80>
Out of Rb leaving private network B: <65.5.4.15,50551;142.3.7.45,80>
Arriving at host 10.1.1.1 in private network A: <10.1.1.1,50551;142.3.7.45,80>

d. **[2 points]** Assume next that host 10.1.1.2 in private network A also opens four connections to port 80 on host 10.1.1.1 in private network B. What is the IP address carried in the destination address field of packets arriving at router Rb on those four connections? Justify your answer.

The destination IP address of those packets is again 142.3.7.45, which is the public address for any host in network B.

e. **[5 points]** How many new entries, if any, would these four connections create on Ra and Rb? Justify your answer.

These four new connections would create four new entries on Ra. They each would map a new outgoing destination port number to one of the four pairs <10.1.1.2,5055i>, i=1,2,3,4.

There would be no new entry on Rb.

3) **[25 points]** Consider a circular DHT with 1024 nodes/servers and hash values in the range $[0,2^{128}-1]$. A key with hash value *h* is assigned to the node whose ID is *closest* to *h*, where closest is defined as "immediate successor" under modulo operation. Assume that the first node to join the DHT is assigned ID 0, and initially owns the entire range of hash values. Subsequent nodes that join are assigned the lower half of the hash range of the node they contact to join the DHT, with their ID being the upper bound of that range. For example, if the DHT had only two nodes with IDs 0 and 2^{127}, respectively, the node with ID 0 would be the owner of any key with hash values in the range $[2^{127}+1, 2^{128} = 0]$, and node 2^{127} would own $[1, 2^{127}]$.

a. **[10 points]** Consider the first four nodes to join after node 0, and assume that they all join by contacting node 0. What are the IDs and hash value ranges for those four nodes after they have joined, and what is the remaining hash value range for node 0 after the four nodes have joined?

Consistent with the above description, when the first node joins, node 0 assigns it ID 2^{127} and range $[1,2^{127}]$ with node 0 still owning range $[2^{127}+1,0]$.
When the second node joins by contacting node 0, node 0 assigns it ID $2^{127}+2^{126}$ and range $[2^{127}+1, 2^{127}+2^{126}]$ with node 0 still owning range $[2^{127}+2^{126}+1,0]$.
When the third node joins by contacting node 0, node 0 assigns it ID $2^{127}+2^{126}+2^{125}$ and range $[2^{127}+2^{126}+1,2^{127}+2^{126}+2^{125}]$ with node 0 still owning range $[2^{127}+2^{126}+2^{125}+1,0]$.
When the fourth node joins by contacting node 0, node 0 assigns it ID $2^{127}+2^{126}+2^{125}+2^{124}$ and range $[2^{127}+2^{126}+2^{125}+1,2^{127}+2^{126}+2^{125}+2^{124}]$ with node 0 still owning range $[2^{127}+2^{126}+2^{125}+2^{124}+1,0]$.

b. **[10 points]** Assume next that the 1024 (2^{10}) nodes of the DHT have been added so that they are evenly distributed across the range of hash values, *i.e.,* they each have a range of $2^{128}/2^{10} = 2^{118}$ hash values, with corresponding IDs of 0, 2^{118}, $2\text{x}2^{118}$, $3\text{x}2^{118}$ … Each node is configured with one shortcut that connects them to the node furthest away from them on the ring, *e.g.,* node 0 has a shortcut to node with ID $512\text{x}2^{118} = 2^9\text{x}2^{118} = 2^{127}$. Consider now that node 0 receives a request for a key with hash value $2^{126}+2^{125}$. Through how many nodes will the request goes through before the answer is sent back to node 0? Justify your answer.

Node 0 has a shortcut to node $2^9x2^{118} = 2^{127}$. The hash value range for this node is $[2^{127}-2^{118}+1,2^{127}]$ and comparing the lower bound of that interval to the key $2^{126}+2^{125}$ shows that the key is below this lower bound, i.e., $2^{126}+2^{125} - (2^{127}-2^{118}+1) <0$. Hence, the request will need to go hop-by-hop until it reaches node $384\text{x}2^{118}$ ($2^{126}+2^{125} - 384\text{x}2^{118} = 0$) for a total of 384 hops before a response is sent back to node 0. Note that the response is sent back directly to node 0, since its IP address is carried in the request packet and, therefore, available to node 384 to send the response back.

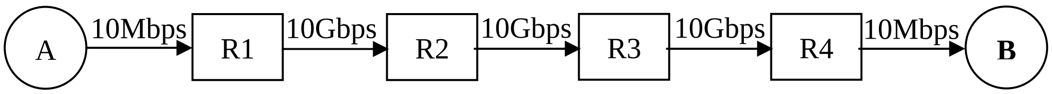
c. **[5 points]** Nodes maintain caches to previous answers so that on average a node can answer a query for a key it does not own with a 10% probability. Consider a request that arrives at node 0 for a key that is owned by node $5\text{x}2^{118}$. What is the expected number of hops the query will go through? Justify your answer

Because the target node for the query is in the first half of the DHT ring, node 0 will not use its shortcut to node 2^{127}, and the query will proceed hop-by-hop along the ring. At each hop, including at node 0, there is a 10% probability that the node will have cached the response so that the average number of hops H for the request is given by

$$H = 0*p+p(1-p)+2p(1-p)^2+3p(1-p)^3+4p(1-p)^4+5(1-p)^5$$

Where p=0.1, which gives H = 3.686 hops.

4) **[10 points]** Two audio clients, A and B, are using 16 kbps codecs, and are connected over a packet network as shown below, *i.e.,* four routers R1 to R4 connected by 10 Gbps links and access links that run at 10 Mbps. The end-to-end one way propagation delay is 15ms.



Assuming that packets travelling from either A to B or B to A see on average queue sizes of about *512* kBytes (a kBytes = 1024 bytes) at each router, and that the audio application they use tolerates a one way average delay of approximately 100ms, what is the maximum packet payload size that the codecs can use? (Ignore the transmission times of the audio packet itself).

Ignoring the transmission times of the audio packet itself, the end to end delay D is of the form:

$D = P + 15ms + 3*512*1024*8/10^{10} + 512*1024*8/10^{7} \approx P + 435.69ms$

Where P is the packetization delay, i.e., $S/16*10^{3}$*, where S is the payload size in bits. Given the 100ms target we have, there is unfortunately no way we can meet it, so that the maximum packet payload is 0…*

If we had instead used 100 Mbps access links, then we would have an end to end delay of the form

$D = P + 15ms + 3*512*1024*8/10^{10} + 512*1024*8/10^{8} \approx P + 58.2ms$

This left us a margin of 41.8ms for the packetization delay. Assuming a 16 kbps codec, this translates into 668.78 bits or just over 83 bytes, which would then have been the maximum possible payload.

5) **[20 points]** Consider two TCP connections that share a common bottleneck link of speed 100 Mbps. Both use TCP Reno with a standard *rcvWindow* size of 64 kBytes and have disabled delayed ACKs. The MSS on both paths is such that the maximum payload size of a TCP packet is 1 kBytes, *i.e.,* 1024 bytes. Connection 1 has an RTT value of 5ms, while connection 2 has an RTT of 10ms.

a. **[10 points]** Assume that connection 1 was initially the only one active and that it had reached a *cwnd* value of 64 kbytes, which therefore saturates the bottleneck link capacity. Explain why this is the case **[5 points],** and consequently how much data is approximately stored in the router's buffer **[5 points]**. Justify your answer. (Ignore the impact of IP and TCP headers).

*The maximum transfer rate that connection 1 can realize is the maximum of the link speed or one cwnd every RTT. This translates into a transfer rate of 64*1024*8/0.005 = 104,857,600 bits/sec, or just over the link speed of 100 Mbps (actually a bit more if we were to count the IP and TCP headers). Hence connection 1 alone can saturate the 100 Mbps link. Once connection 1 cwnd reaches its maximum value, subsequent transmissions are triggered by the reception of an ACK and are therefore paced by the bandwidth of the bottleneck link. The cwnd value that corresponds to saturating the 100 Mbps link is given by cwnd*8/0.005 =* 10^8*, or cwnd = 62,500 bytes or about 2 kbytes less than the maximum value of 64 kbytes for cwnd. Hence, connection 1 was allowed to send an extra 2 kbytes as cwnd1 was increasing beyond the value it needed to saturate the link, and those 2 kbytes are what is stored in the router buffer.*

b. **[5 points]** Assume next that connection 2 becomes active and that it starts with a value of *ssthresh* = 32kbytes, and that both connection 1 and connection 2 experience a loss right after connection 2 enters the congestion avoidance phase. What are the values of *cwnd1* and *cwnd2* after their respective connections exit the fast recovery phase following the loss and re-enter congestion avoidance? Justify your answer.

After the loss and the completion of the fast recovery phase, connection 1 will halve its cwnd value and switch to a value of cwnd1 = 32 kbytes. Conversely, connection 2 will have exited slow-start as soon as cwnd2 = ssthresh = 32 kbytes. Hence, after the loss and the completion of its fast recovery phase, connection 2 will also halve its cwnd value and switch to a value of cwnd2 = 16 kbytes.

c. **[5 points]** Once connection 1 and connection 2 reach steady state, what will approximately be their respective average transfer rates? Justify your answer.

Because of their difference in RTTs, connection 1 will grow its cwnd twice as fast as connection 2, so that at steady state they will split the link bandwidth approximately 2/3 for connection 1 versus 1/3 for connection 2, i.e., 66 Mbps vs. 33 Mbps, respectively. Note that a 10ms RTT would allow connection 2 to reach a transfer rate of about 52Mbps if alone on the shared bottleneck link, i.e., larger than the value it ultimately settles on when sharing the link with connection 1.