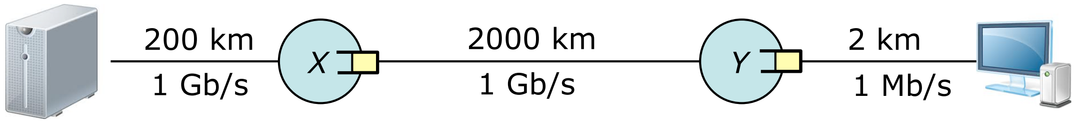
CSE 473 – Introduction to Computer Networks *Jon Turner*

Final Exam Solution

1. (8 points). The figure below shows a network path connecting a server to a client.



What is the propagation delay for a packet going from the server to the client (you may assume that the speed of light is 200,000 km/s)?

1 ms + 10 ms + .01 ms = 11.01 ms

What is the total transmission delay of a 10,000 bit packet on all of the links?

10 μs + 10 μs + 10 ms = 10.02 ms

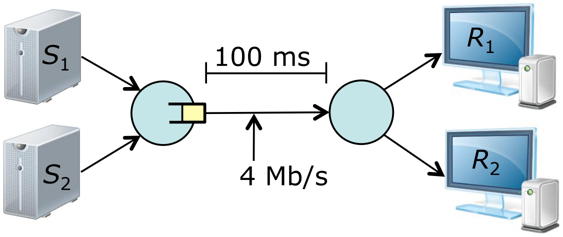
What is the average queueing delay at router *X*, assuming that the traffic intensity is 1.3, and the buffer can hold 10,000 packets and that the average packet size is 5,000 bits?

*In this case, the queue will be nearly full all the time, so it takes about 10,000*5,000 ns or 50 ms for a packet to get to the front of the queue.*

What is the average queueing delay at router *Y*, assuming that the traffic intensity is 0.8, and the buffer can old 100 packets and the average packet size is 5,000 bits?

*In this case, the average number of packets in the queue is 0.8/(1–0.8)=4, so the queueing delay is 4*5,000 μs or 20 ms.*

2. (15 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 4 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?



The RTT is 200 ms in this case, so the link rate is equivalent to 800 Kb per RTT or 100 KB.
So, $cwnd_1$=$cwnd_2$= 50 KB.

Assume that the link buffer overflows whenever $cwnd_1+cwnd_2 \geq 200$ KB and that at time 0, $cwnd_1$=40 KB and $cwnd_2$=160 KB. Approximately, what are the values of $cwnd_1$ and $cwnd_2$ one RTT later? Also, what are the values of ssthresh for each of the two connections? Assume that all losses are detected by triple duplicate ACKs.

Since we are using Tahoe, we go to slow start in this case. So, $cwnd_1$=$cwnd_2$= 1 KB, $ssthresh_1$=20 KB and $sshresh_2$= 80 KB.

After 7 more RTTs, approximately what are the values of $cwnd_1$ and $cwnd_2$?

22 KB and 80 KB

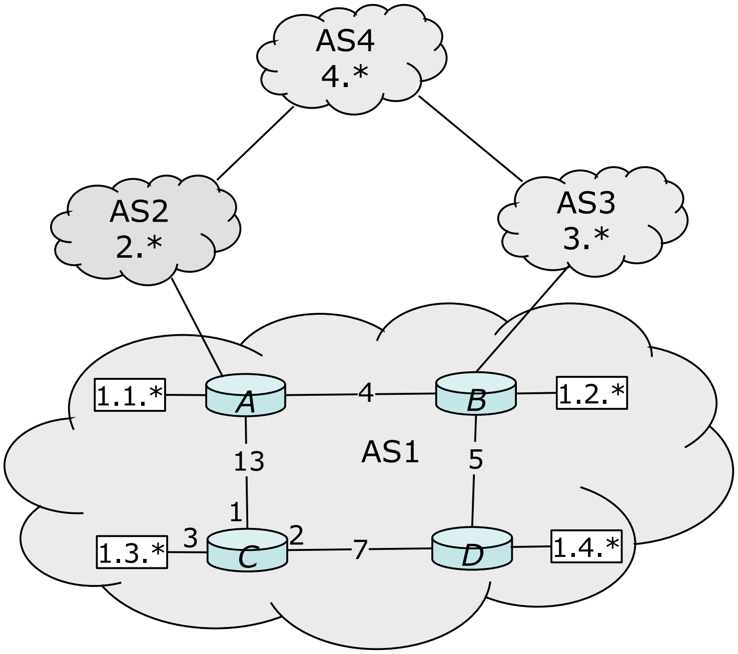
Approximately, how many more RTTs before $cwnd_1+cwnd_2 \geq 200$ KB again?
What is $cwnd_2-cwnd_1$ at this point?

(200–102)/2=49 RTTs *$cwnd_2-cwnd_1$=58 KB*

Approximately, how many more RTTs pass before $cwnd_1+cwnd_2 \geq 200$ KB and $cwnd_2-cwnd_1 \leq 10$ KB?

The difference goes down by roughly a factor of 2 after every cycle of the congestion control algorithm.
So, 3 more cycles will be required to get the difference below 10 KB.
So it will take about 3(1+7+49)=171 RTTs.*

3. (10 points) The figure below shows a portion of the internet with four autonomous systems, one of which is shown with four routers, each with its own /16 subnet. Note that each of the other ASs has a /8 subnet. The blank routing table at right is for router *C*. Complete the table, providing an entry for each of the seven subnets. Assume that AS1 uses OSPF as its intra-AS routing algorithm, and BGP as its inter-AS routing algorithm. Use the standard tie-breaking rules and assume there are no policy constraints that must be satisfied.



prefix	output link
1.1.*	1
1.2.*	2
1.3.*	3
1.4.*	2
2.*	1
3.*	2
4.*	2

Suppose the link joining router *B* with AS3 fails, which entries would change, and how would they change?

The output link fields of the last two entries would both change to 1.

4. (8 points) Consider a VoIP system that uses a fixed playout delay of 200 ms. If a receiver receives a sequence of four packets with timestamps 17.02, 17.04, 17.06 and 17.08 (where the timestamp is expressed in seconds relative to some arbitrary starting time), when will the receiver play out these packets (you may assume that the receiver's clock is synchronized with the sender's).

   At times 17.22, 17.24, 17.26 and 17.28.

   Now, suppose that we are using an adaptive playout delay, instead of a fixed playout delay. After a period of silence, the receiver starts receiving a sequence of voice packets with timestamps 30.01, 30.03, 30.05 and 30.07. Let d_i=50 ms be the current value for the average measured delay across the network, let v_i=5 ms be the current value of the measured delay variation. When computing the playout time, multiply the delay variation by the parameter K=5. At what times are the four packets played out?

   The playout delay is 50+25=75 ms, so the playout times are 30.085, 30.105, 30.125 and 30.145.

   Now suppose the VoIP system does loss concealment by separating "odd" voice samples from "even" voice samples and sending them in different packets. In this scheme, samples from missing packets can be interpolated using the samples from the previous and following packets. How does this affect the required playout delay if voice packets are still sent at the rate of 50 packets per second, during a talkspurt?

   To compute the samples from a lost packet, we need to wait for the next packet to arrive. This effectively increases maximum delay that packets can experience, adding 20 ms to the playout delay.

5. (10 points) Consider a token bucket rate controller, used to control a reserved rate flow. Assume that the token bucket has a capacity of 10 tokens and a token fill rate of 100 tokens per second, and that every packet consumes one token. If no token is available for an arriving packet, it is marked for possible discarding.

Suppose that at time 0, the token bucket is empty and the next token "arrives" at time 10 ms. If packets arrive at times 11, 13, 17, 19, 23 and 29, which packets (if any) are marked. (You may identify the marked packets by their arrival times.)

Packets 13, 17, 19 and 29 will all be marked.

How many tokens are in the token bucket at time 55 ms if no additional packets arrive?

Tokens are added to the bucket at times 30, 40 and 50, so the token bucket will have 3 tokens at time 55.

How many tokens are in the token bucket at time 195 ms?

10.

What is the largest number of packets that can be sent between time 201 ms and time 299 ms without any of the packets being marked?

We get at most 9 new tokens during this period so at most 19 packets can be sent without being marked.

If this largest possible number is sent between times 201 and 299, what is the largest number of packets that can be sent between times 301 ms and 399 without any of the packets being marked?

New tokens are added 300, 310,..., 390, so 10 more can be sent without being marked.

6. (12 points) Consider a user who is downloading a large file from a remote web server to her laptop in a coffee shop with WIFI access. The shop's DSL link has a download rate of 5 Mb/s and the wireless network supports a peak rate of 50 Mb/s. Assume there are no other limits on the user's download rate (no one else in the coffee shop is online and the server and internet path are lightly loaded).

   What is the maximum throughput the user can expect in this case?

   5 Mb/s

   Approximately how often does TCP have to retransmit a packet in this scenario? Assume that the maximum segment size is 10,000 bits and that the round-trip time is 122 ms.

   To answer this, we first need to compute the loss probability for the TCP connection.

   $L=((1.22*10{,}000)/(.122*5{,}000{,}000))^2=(1/50)^2=.02^2=.0004$

   so TCP must retransmit approximately one packet in 2500 and since it will send an average of 500 packets per second, this translates to one packet every 5 seconds.

   Now, assume that the wireless network experiences interference whenever the coffee grinder is running. During this time, it fails to deliver approximately 4% of the packets sent to the user. Approximately what throughput can you expect in this situation?

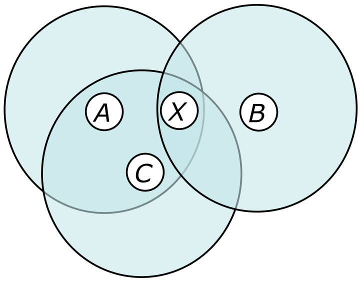
   This loss rate is 100 times larger than the loss rate in the first scenario, and since the throughput is inversely proportional to the square root of the loss rate, we expect the throughput to go down by a factor of 10 to 500 Kb/s. Or, we can apply the throughput equation directly.

   *T=(1.22*10,000)/(.122*sqrt(.04))=100,000/.2=500 Kb/s*

   Suppose the throughput went down by a factor of 2, relative to the previous situation. Approximately what would the packet loss rate have to be, in order to explain this?

   Since loss rate is inversely proportional to the square of the throughput, the loss rate must have gone up by factor of 4 to 16%.

7. (10 points) The diagram at right shows a WIFI network with an access point, *X* and three hosts, *A*, *B* and *C*. The large circles indicate the *coverage areas* of the three hosts. The coverage area for *X* is not shown, but you may assume that it includes all three hosts. Assume RTS/CTS are not used.



Suppose *X* is transmitting a packet at time 0 and finishes sending it at time 100 μs. Also,

- *A* gets a packet to send at time 50 that takes 100 μs to send and is assigned a backoff timer of 70.
- *B* gets a packet at time 70 that takes 200 μs and is assigned a backoff timer of 200.
- *C* gets a packet at time 120 that takes 150 μs and is assigned a backoff timer of 150.

For each of the three hosts, what time do they start sending their packets? You may ignore the inter-frame spacing and the time required for acks.

A starts sending at time 170
B starts sending at time 300
C defers while A is sending, so it starts at time 120+100+150=370

Of the three packets sent, which are successfully delivered on the first attempt?

Only the packet from A is delivered.

For each packet that is not successfully delivered on the first attempt, approximately when does the sending host learn that the packet was lost and must be sent again?

Hosts learn of lost packets from the absence of ACKs. Here, B would expect an ACK at 500 and so would learn of the lost packet when the ACK fails to arrive at time 500. Similarly, C would learn of its lost packet at time 370+150=520.

Now, suppose RTS/CTS is enabled. In this case, approximately when does each host send its data packet? You may assume that the time needed to send RTS, CTS and ACK packets is negligible.

A sends at 170.
B sends at 550.
C sends at 370.

8. (10 points) Recall that an RSA encryption key is a pair of numbers (n,e) and the corresponding decryption key is another pair (n,d). Which of the following could be used as RSA key pairs (ignoring the fact that they are too small)? For those that are valid key pairs, show that they meet all the requirements for a key pair, for those that are not, explain why they are not.

   - (31,5), (31,11)

     This is not a valid key pair as n=31 is a prime, and for a valid pair, it must be the product of two primes.

   - (77,7), (77,43)

     n=pq, where p=7, q=11, (p–1)(q–1)=60, 7 and 60 are relatively prime and (ed–1)=300 is a multiple of 60, so this is a valid key pair.

   - (55,7), (55,41)

     This is not a valid key pair, since (p–1)(q–1)=50 and (ed–1)=286 is not a multiple of 50.

   Consider the pair (91,5), (91,29). Assume the first is the encrypted key and the second is the decryption key. What is the encrypted value of the number 10? (*Hint*: 10^2 mod 91=9)

   *(10^2 mod 91)= 9, so (10^4 mod 91)= 81 and (10^5 mod 91)=(810 mod 91)=810–8*91=810–728=82*

9. (10 points) The diagram below represents a portion of the payload of an IP datagram carrying data for an SSL session. Each line represents a distinct SSL record.

type	version	length	data	MAC
0	3.2	35	kasfjla;kja;ljf	2343
0	3.2	41	aiusrooqirgn/nc;p	9783
0	3.2	25	qp98hj;vn	8520
1	3.2	36	z2jt3;r’wjf2h5	2985

Which of the fields in each record are encrypted by the sender, and which are sent as cleartext.

The type, version and length field are sent as clear text, while the data and MAC are encrypted.

Suppose an attacker was able to intercept and modify packets as they pass through the internet. If the attacker were to modify several bytes in the data field of the second record, what changes would the attacker have to make to the IP and TCP headers, in order to ensure that these records were actually delivered to the SSL software.

It would have to recompute the IP and TCP checksums.

How would the SSL software detect the modification to the second record?

The MAC for record 2 would not match the one computed for the modified record.

Suppose the attacker were to make a copy of the second record and insert it after the third record (so the modified packet had five records). Explain how the SSL software at the receiver would detect this modification?

The computed MAC at the receiver would not match for this record, since it uses a different sequence number that the sender did for this record.

Suppose the attacker were able to discover the key used for the MAC, but not the encryption key. Could the attacker use this to change the type field of the third record, without it being detected by the SSL software at the receiver. If so, explain how this is done. If not, explain why not.

This could not be done, because the MAC field must be encrypted. So, even if the intruder could compute a new MAC field, it could not produce the encrypted version. Actually, since the MAC is computed over the unecrypted record, the attacker could not even compute the required MAC value, let alone encrypt it.