CSE 123: Computer Networks

Homework 4 Solutions

Out: 5/25, Due: 6/1

Problems

1. **Scheduling and QoS**

Packet #	**Size**	**Flow**
1	70	1
2	210	1
3	100	1
4	260	2
5	20	2
6	240	3
7	90	3
8	180	3

Suppose a router has three input flows and one output. It receives the packets listed in the table above all at about the same time, in the order listed, during a period in which the output port is busy but all queues are otherwise empty. Give the order in which the packets are transmitted for

a. Fair Queueing.
   i. 1, 6, 4, 2, 5, 7, 3, 8

b. Weighted fair queuing with flow 2 having twice as much share as flow 1, and flow 3 having three times as much share as flow 1.
   i. Can build a table using the weighted size instead of the size of a packet where the weighted_size = size / weight_of_flow

Packet #	**Weighted Size**	**Flow**
1	70	1
2	210	1
3	100	1
4	130	2
5	10	2
6	80	3
7	30	3
8	60	3

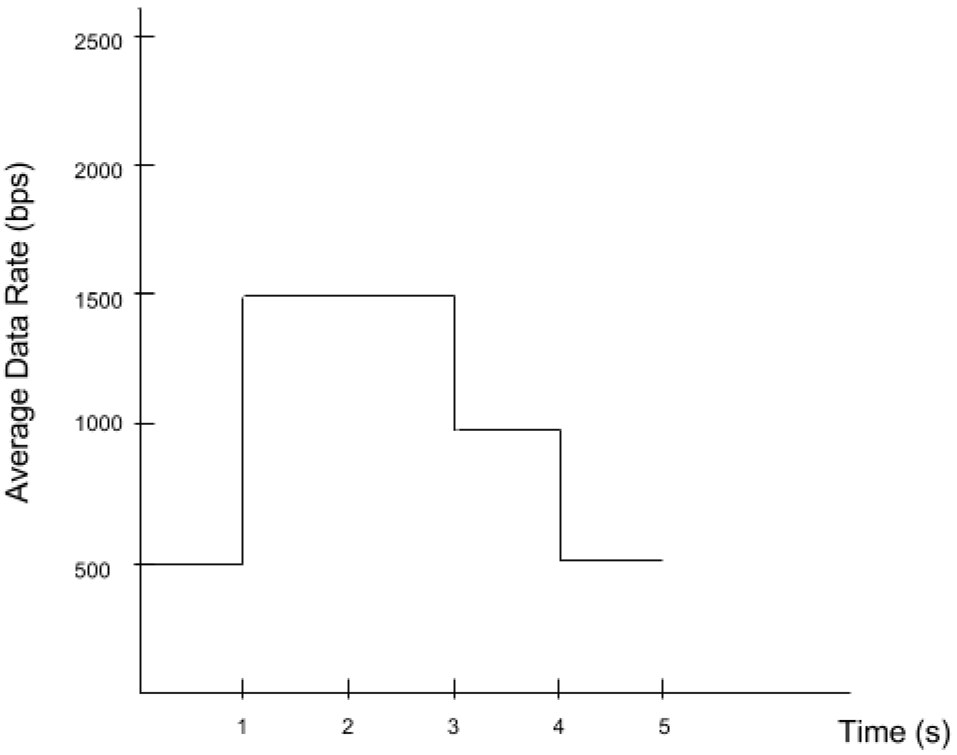
ii. 1, 6, 7, 4, 5, 8, 2, 3

Note: Resolve ties in the order of flow 1, flow 2, and flow 3

2. **Token Bucket Filter**

   Assume that a host is sending a router packets that it would like the router to forward. The router has implemented traffic policing using a token bucket filter as shown in the lecture slides. The token bucket has a fill rate of “r” bits per second (bps) and has a depth of “b” bits.

   For simplicity assume that data can be sent as individual bits rather than packets. Assume the average rate at which the host sends the router data follows the figure below and that the token bucket is initially full when time is 0.
   <u>Show your work for all parts of this problem.</u>



a. If r = 1000 bps and b = 500 bits, will there be any loss of data? If so, at what time will the data loss start occurring?
    i. In the first second, the bucket is filling faster than it's being drained, and the bucket is full, so it's still full when time = 1.
    ii. Between 1 and 2 seconds, the buckets is being drained 500 bps faster than it is filling, so the bucket is empty at 2 seconds.
    iii. Between 2 and 3 seconds, the bucket is still trying to be drained at 500 bps faster than it is filling with an empty bucket which means every other "bit" would be dropped.
    iv. After 3 seconds, the fill rate is always greater or equal to the drain rate, so there will be no more loss.
    v. So, yes there will be loss of data that starts when time is 2 seconds.

b. Keeping r = 1000 bps, what is the minimum bucket depth "b" needed for the host to successfully meet the demand of the figure above?
    i. The only time the bucket is drained faster than it is filling is between 1 and 3 seconds. The bucket drains 500 bps faster than it fills in this time window.
    ii. So, 500 bps for 2 seconds gives a bucket needed bucket depth of 1000 bits.

3. **Network Address Translation**

NAT, is the translation of an Internet Protocol address (IP address) used within one network to a different IP address known within another network. Recall, a NAT capable router essentially translates private address within a network to public addresses that can be used publicly on the internet.

In this question, we look at the inner workings of a simple NAT capable router. The router will have mappings between the private addresses within the network (here, the private addresses all fall within the 10/8 network) to the public address(es) that it uses. Let us assume that the router has a single public address 72.24.47.124 which it uses for all communication with hosts that are not part of the private network. The router multiplexes its public IP address(es) as needed and keeps track of the multiplexing in a NAT translation table.

Assume that the router multiplexes the public address using ports starting from 9000 and then incrementing by one. For example, if a host in the private network with address 10.0.0.5:6000 sends a message to 132.239.8.45:80 then the entry in the NAT table would be filled in as below. The next time the router will use 9001 as the port to establish a new connection and so on.

NAT Translation Table	
WAN Side Address	**LAN Side Address**
72.24.47.124:9000	10.0.0.5:6000
...	...

a. What would be the entries in the NAT Translation Table at the end of the following events

   i. 10.0.0.6:5000 sends a message to 74.125.239.33:80
   ii. 10.0.0.10:6000 sends a message to 204.79.197.200:80
   iii. 10.0.1.101:6001 sends a message to 206.190.36.45:80
   iv. 10.0.0.10:6000 sends another message to 204.79.197.200:80
   v. 10.0.1.101:6001 sends a message to 74.125.239.33:80
   vi. 10.0.0.7:7000 sends a message to 63.245.215.20:80
   vii. 204.79.197.200:80 sends a message to 10.0.0.10:6000

viii. 206.190.36.45:80 sends a message to 74.125.239.33:80

Note: The NAT Table should be like the one above with the entries generated from the events in (a.) filled in

NAT Translation Table	
WAN Side Address	**LAN Side Address**
72.24.47.124:9000	10.0.0.5:6000
72.24.47.124:9001	10.0.0.6:5000
72.24.47.124:9002	10.0.0.10:6000
72.24.47.124:9003	10.0.1.101:6001
72.24.47.124:9004	10.0.0.7:7000

Events iv, v, vii, and viii have no impact on the NAT Translation Table

b. For simplicity, let us assume that message format is MSG<Sender, Receiver>. In that case, if a host in the private network with address 10.0.0.5:5000 sends a message to 132.239.8.45:80 then the message recevied at the router and leaving at the router would look as follows

   Message Received from Host: MSG<10.0.0.5:5000, 132.239.8.45:80>
   Message Sent from Router: MSG<72.24.47.124:9000,132.239.8.45:80>

   Note: We will need to use the entries from the table we filled in (a.) to do this.

   List out the message received from the host at the router and the message sent from the router (like shown above) for the following messages:

   i. 10.0.0.6:5000 sends a message to 74.125.239.33:80

      Message Received from Host: MSG<10.0.0.6:5000, 74.125.239.33:80>

Message Sent from Router: MSG<72.24.47.124:9001, 74.125.239.33:80>

ii. 10.0.0.10:6000 sends a message to 204.79.197.200:80

Message Received from Host: MSG<10.0.0.10:6000, 74.125.239.33:80>
Message Sent from Router: MSG<72.24.47.124:9002, 74.125.239.33:80>

c. If the router gets a message MSG<74.125.239.33:80, 72.24.47.124:9001>, what would the message look like leaving the router?

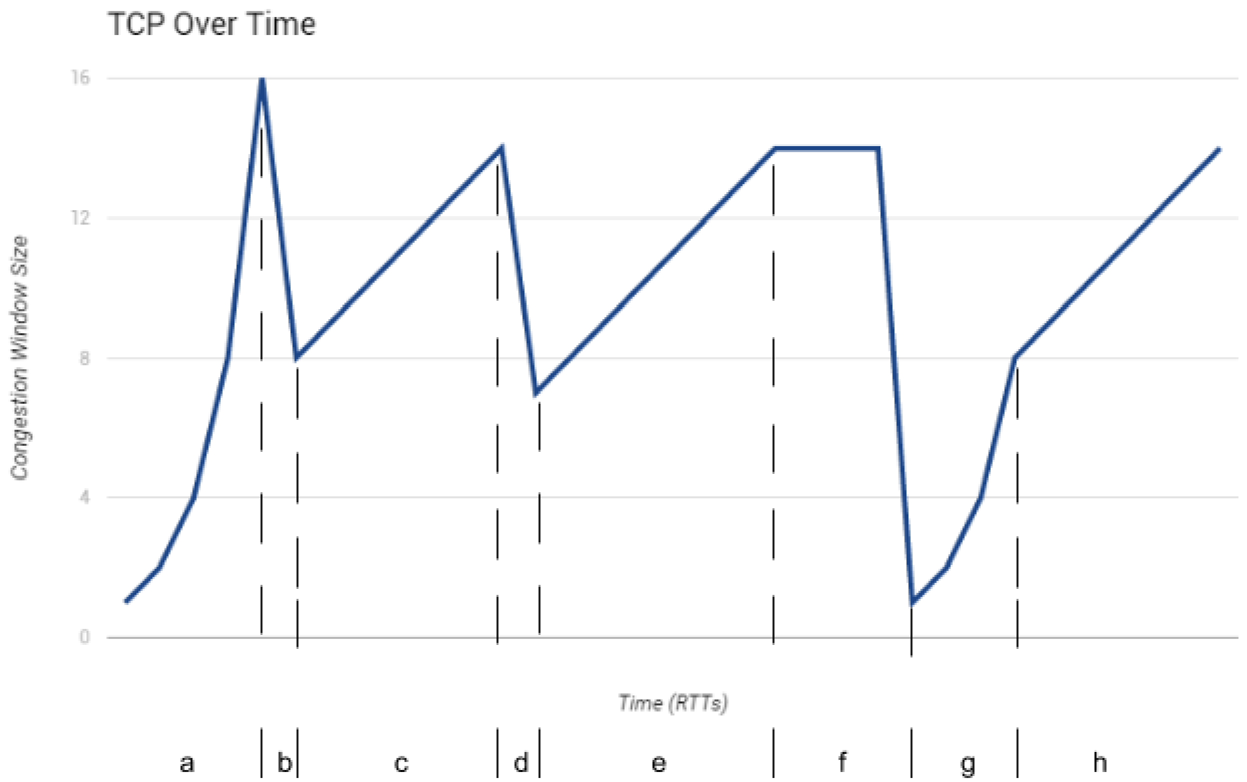
MSG<74.125.239.33:80, 10.0.0.6:5000>

d. Finally, if the router gets a message MSG<10.0.0.10:6000, 10.0.1.101:6001>, what what would the message look like leaving the router?

No change to the message because they are both private addresses.

4. TCP Behavior

The figure below shows the behavior of a TCP connection implementing the TCP Reno flavor of TCP including slow start, congestion avoidance, fast retransmit, and fast recovery. Every letter in the figure denotes a part of the graph divided by the vertical dashed lines that can be described by one of the phrases in the word list also given below. For every letter in the graph (a-h), give the phrase from the word list that best describes that part of the graph. The phrases in the word list can be used more than once and possibly not at all.



Word List

Quality of Service	Additive Increase	Maximum Segment Size
Fast Retransmit + Fast Recovery	Slow Start	Timeout

a. Slow Start
b. Fast Retransmit + Fast Recovery
c. Additive Increase
d. Fast Retransmit + Fast Recovery
e. Additive Increase
f. Timeout
g. Slow Start
h. Additive Increase

5. **TCP Slow Start Extension**

   Assume that TCP implements an extension that allows window sizes much larger than 64KB. Suppose that you are using this extended TCP over a 1-Gbps link with a constant one-way latency of 50 ms to transfer a 10MB file, and the TCP receive window is 16MB. If TCP sends 1KB packets (assuming no congestion and no lost packets):

   a. How many RTTs does it take until slow start opens the send window to 1MB?

      i. We start with a window size of 1 KB that doubles every RTT. The mathematical way to solve this is to realize that 1 MB is 1024 times the size of 1 KB. We can then take log base 2 of 1024 which is equal to 10.

      ii. So, it would take 10 RTTs until the send window becomes 1MB

   b. How many RTTs does it take to send the file?

      i. When our window size became 1MB it means we've sent 1KB + 2KB + 4KB + … + 512KB = 1MB - 1KB. The only bound we have on our window size is the receive window which is 16MB because the TCP algorithm won't set the send window larger than the advertised window which is never larger than the receive window. Until this limit is reached, the window size will keep doubling.

      ii. First let's calculate how much still needs to be sent: 10MB - (1MB - 1KB) = 9MB + 1KB. To keep things clean let's put the rest in a table.

RTT	New Send Window	Remaining Data
10	1MB	9MB + 1KB
11	2MB	8MB + 1KB
12	4MB	6MB + 1KB
13	8MB	2MB + 1KB
14	---	DONE

      iii. So, it would take 14 RTTs to send the whole file.

c. If the time to send the file is given by the number of required RTTs multiplied by the RTT, what is the effective throughput for the transfer? In other words, what percentage of the link bandwidth is utilized?

    i. The round trip time (RTT) is the 2-way delay which is 100 ms in this problem. Thus, the time it took to send the file was 14 * (100 ms) = 1.4 s.

    ii. To find the link utilization for that time, we compare the amount of data that was sent with the amount of data that could have been sent during that time. We sent a 10MB file. Using a 1Gbps link, in 1.4 seconds, 1.4 * 10^9 bits could have been sent. An exact answer for a percentage precise to 2 decimal places would then be (10 * 2^20 * 8) / (1.4 * 10^9) = 0.0599 = 5.99% of the link. An estimate assuming that 1MB = 8 * 10^6 bits would be about 5.71%.