First Midterm for ECE374
03/24/11
Solution!!

Note: *In all written assignments, please show as much of your work as you can. Even if you get a wrong answer, you can get partial credit if you show your work.* If you make a mistake, it will also help the grader show you where you made a mistake.

Problem 1:
In modern packet-switched networks the source hosts segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packet back into the original message. We refer to this process as message segmentation. Figure 1 below illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is $8*10^6$ bits long that is to be sent from source to destination. Suppose each link has a maximum capacity of 2Mbps. Ignore propagation, queuing, and processing delays.

a. Consider sending the message from source to destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?
b. Now suppose that the message is segmented into 4,000 packets, with each packet being 2,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source to the first switch. At what time will the second packet be fully received at the first switch?
c. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.
d. Discuss the drawbacks of message segmentation.

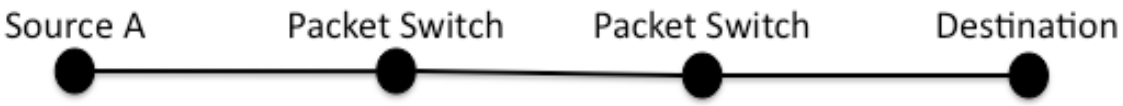


Figure 1 Network for problem 1

Solution:

a) Time to send message from source host to first packet switch = $\frac{8\times10^6}{2\times10^6}\text{sec} = 4\text{sec}$. With store-and-forward switching, the total time to move message from source host to destination host = $4\text{sec}\times 3\ hops = 12\text{sec}$

b) Time to send 1st packet from source host to first packet switch = . $\frac{2\times10^3}{2\times10^6}\text{sec} = 1\,m\text{sec}$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times1\,m\text{sec} = 2\,m\text{sec}$

c) Time at which 1st packet is received at the destination host = . $1\,m\text{sec}\times 3\,hops = 3\,m\text{sec}$. After this, every 1msec one packet will be received; thus time at which last (4000th) packet is received = $3\,m\text{sec} + 3999 * 1m\text{sec} = 4.002\,\text{sec}$. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).

d) Drawbacks:
   i. Packets have to be put in sequence at the destination.
   ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

Problem 2:

Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that *n* DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT_1, …, RTT_n. Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object.

a. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

Now suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

b. Non-persistent HTTP with no parallel TCP connections?
c. Non-persistent HTTP with browser configured for 5 parallel connections?
d. Persistent HTTP?
e. Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in the department? Explain.

Solution:

a. The total amount of time to get the IP address is $RTT_1 + RTT_2 + \cdots + RTT_n$. Once the IP address is known, RTT_O elapses to set up the TCP connection and another RTT_O elapses to request and receive the small object. The total response time is $2RTT_o + RTT_1 + RTT_2 + \cdots + RTT_n$

b. $RTT_1 + \cdots + RTT_n + 2RTT_o + 8 \cdot 2RTT_o = 18RTT_o + RTT_1 + \cdots + RTT_n$.

c. $RTT_1 + \cdots + RTT_n + 2RTT_o + 2 \cdot 2RTT_o = 6RTT_o + RTT_1 + \cdots + RTT_n$

d. $RTT_1 + \cdots + RTT_n + 2RTT_o + RTT_o = 3RTT_o + RTT_1 + \cdots + RTT_n$.

Problem 3:
Consider sending a large file from a host to another over a TCP connection that has no loss.

a. Suppose TCP uses AIMD for its congestion control without slow start. Assuming *cwnd* increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for *cwnd* to increase from 5 MSS to 11 MSS (assuming no loss events)?
b. What is the average throughput (in terms of MSS and RTT) for this connection up through time = 6 RTT?

Solution:

a. It takes 1 RTT to increase CongWin to 6 MSS; 2 RTTs to increase to 7 MSS; 3 RTTs to increase to 8 MSS; 4 RTTs to increase to 9 MSS; 5 RTTs to increase to 10 MSS; and 6 RTTs to increase to 11 MSS.
b. In the first RTT 5 MSS was sent; in the second RTT 6 MSS was sent; in the third RTT 7 MSS was sent; in the forth RTT 8 MSS was sent; in the fifth RTT, 9 MSS was sent; and in the sixth RTT, 10 MSS was sent. Thus, up to time 6 RTT, 5+6+7+8+9+10 = 45 MSS were sent (and acknowledged). Thus, we can say that the average throughput up to time 6 RTT was (45 MSS)/(6 RTT) = 7.5 MSS/RTT.

Problem 4:
Consider distributing a file of F = 15 Gbits to N peers. The server has an upload rate of u_s = 30 Mbps, and each peer has a download rate of d_i = 2 Mbps and an upload rate of u. For N = 10, 100, and 1,000 and u = 300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combination of N and u for both client-server distribution and P2P distribution.

Solution:
For calculating the minimum distribution time for client-server distribution, we use the following formula:

$$D_{cs} = max\ \{NF/u_s,\ F/d_{min}\}$$

Similarly, for calculating the minimum distribution time for P2P distribution, we use the following formula:

$$D_{P2P} = max\{F/u,\ F/d_{min},\ NF/(u + \sum_{i=1}^{N} u_i)\}$$

Where, F = 15 Gbits = 15 * 1024 Mbits
u_s = 30 Mbps
$d_{min} = d_i$ = 2 Mbps

Note, 300Kbps = 300/1024 Mbps.

Client Server

		N		
		10	**100**	**1000**
	300 Kbps	7680	51200	512000
u	**700 Kbps**	7680	51200	512000
	2 Mbps	7680	51200	512000

Peer to Peer

		N		
		10	**100**	**1000**
	300 Kbps	7680	25904	47559
u	**700 Kbps**	7680	15616	21525
	2 Mbps	7680	7680	7680

Problem 5:

Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2nd segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.

a. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.
b. If the timeout values for all three protocols are much longer than 5 * RTT, then which protocol successfully delivers all five data segments in shortest time interval?

Solution:

a. GoBackN:
   A sends 9 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2, 3, 4, and 5. B sends 8 ACKs. They are 4 ACKS with sequence number 1, and 4 ACKS with sequence numbers 2, 3, 4, and 5.

   Selective Repeat:
   A sends 6 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later re-sent segments 2. B sends 5 ACKs. They are 4 ACKS with sequence number 1, 3, 4, 5. And there is one ACK with sequence number 2.

   TCP:
   A sends 6 segments in total. They are initially sent segments 1, 2, 3, 4, 5 and later

re-sent segments 2. B sends 5 ACKs. They are 4 ACKS with sequence number 2. There is one ACK with sequence numbers 6. Note that TCP always send an ACK with expected sequence number.

b. TCP. This is because TCP uses fast retransmit without waiting until time out.

Problem 6:

For this problem you should familiarize yourself with Figure 2 first. Now assume that in the network shown in Figure 2 two parallel TCP transmissions are performed. *TCP1* is a transmission between Source A and Sink A that uses *TCP Tahoe*. *TCP2* is a transmission between Source B and Sink B that uses *TCP Reno.* Initial *ssthresh* for both TCP transmissions is set to 32. In this specific scenario no additional delay through forwarding is introduced. Thus, the RTT is only composed of the sums of the delay indicated on each link, times two.

a. For the *TCP 1* transmission, draw the resulting congestion window, assuming that a packet loss (triple duplicate ACKs) is detected at time t=900ms.
b. For the *TCP 2* transmission, draw the resulting congestion window, assuming that a packet loss (triple duplicate ACKs) is detected at time t=650ms.
c. Describe the benefit of TCP Reno over TCP Tahoe.

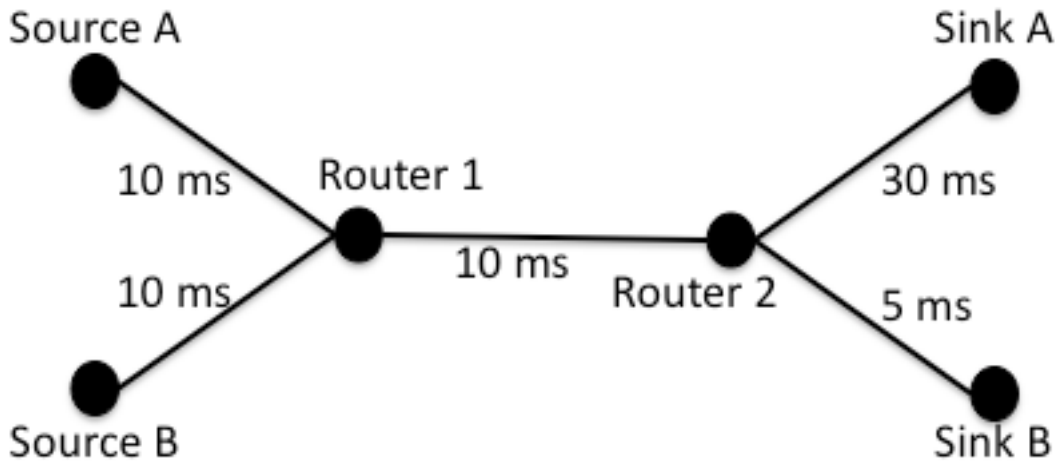


Figure 2 Network layout for problem 5.

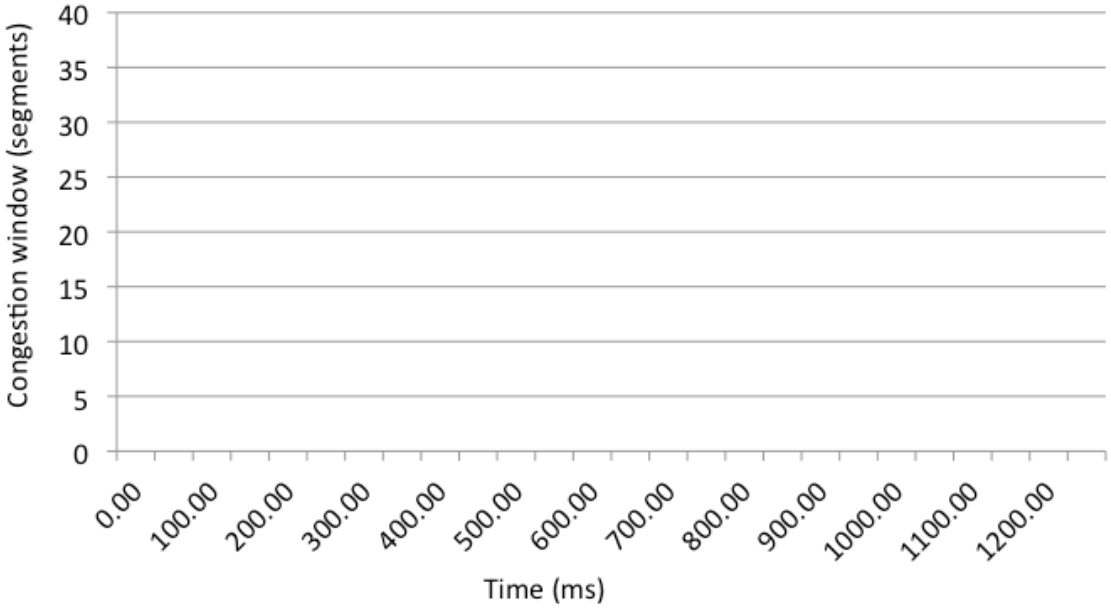
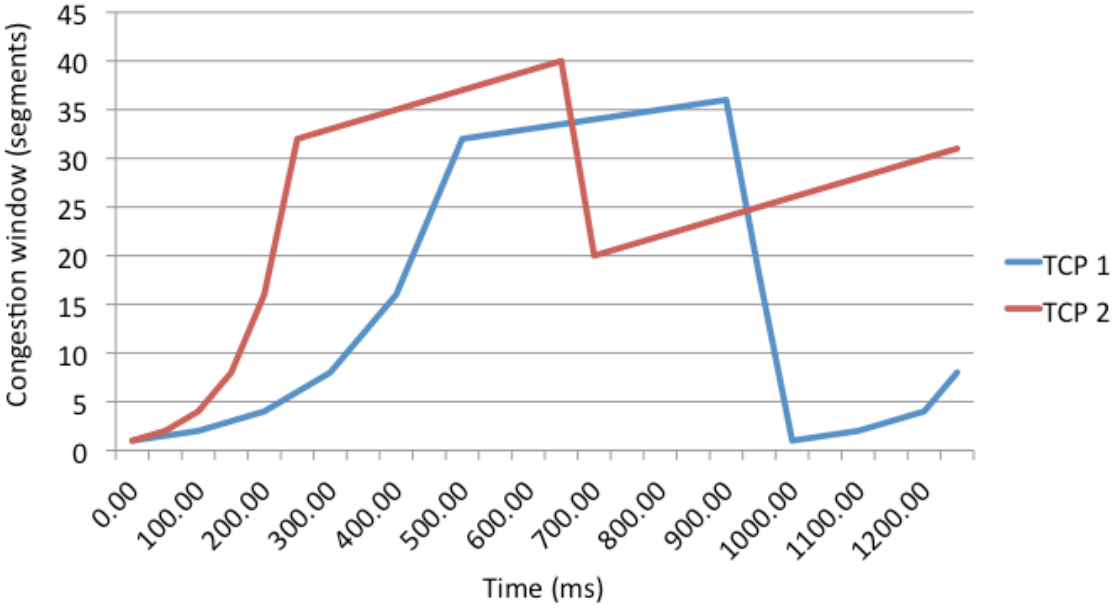


Figure 3 Solution

Solution:


Congestion window (segments)
45
40
35
30
25
20
15
10
5
0
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00
800.00
900.00
1000.00
1100.00
1200.00
Time (ms)
TCP 1
TCP 2