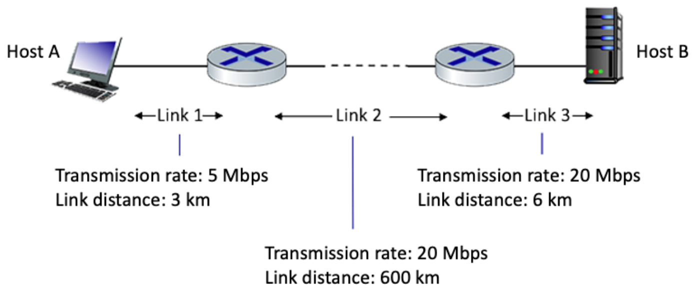
Student ID: ______________________ Section Number: __________
Student name: ____________________

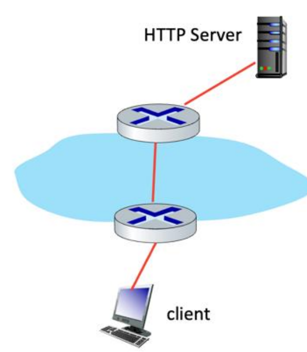
ENSF 462 – Final Practice

1. As shown in the figure below, Host A sends a packet of size 3 Kbits to Host B via a path with three links, each with the specified transmission rate and link length. What is the end-to-end delay? Consider store-and-forward packet transmission with zero queueing delay and processing delay.



$$D_{end-to-end} = D_{tran}^{link\ 1} + D_{prop}^{link\ 1} + D_{tran}^{link\ 2} + D_{prop}^{link\ 2} + D_{tran}^{link\ 3} + D_{prop}^{link\ 3}$$
$$= \frac{3*10^3}{5*10^6} + \frac{3000}{3*10^8} + \frac{3*10^3}{20*10^6} + \frac{6*10^5}{3*10^8} + \frac{3*10^3}{20*10^6} + \frac{6000}{3*10^8}\ secs = 2.93\ msecs$$

2. Consider the scenario shown in the figure below, assuming that the client and server have not communicated before, that the client knows the server's IP address, and that HTTP 1.1 is being used. Now assume that the client has a browser cache (but there is no web cache). The browser cache is initially empty and all GET request messages have an If-Modified-Since field. The HTTP client:
   - makes a first GET request to the server for a base page that it has never before requested; the base page contains a reference to an embedded jpeg object.
   - makes a second GET request for that embedded object. The jpeg file has a transmission time of 1/4 RTT.
   - then makes a third GET request (a very short time later, which means that the object is not modified) for the same embedded jpeg object (with that third GET request having an If-Modified-Since field).



   Let's focus on the second HTTP GET request. How many RTTs are needed from when the browser issues the second HTTP GET request, until the reply for that second GET request is received?

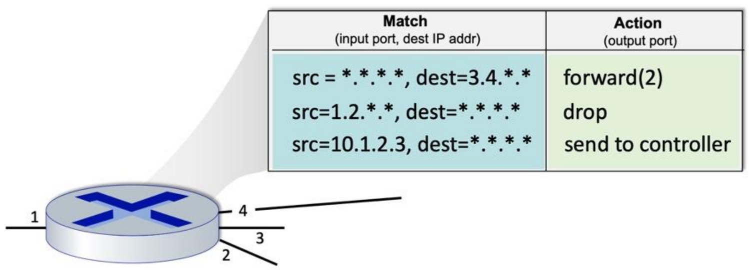
Student ID: ______________________ Section Number: __________
Student name: ____________________

Since HTTP 1.1 is used, so no TCP connection setup required for the second GET message. End-to-end delay = 1 RTT + transmission time = 1.25 RTT.

Let's focus on just this third HTTP GET request. How many RTTs are needed from when the browser issues the third HTTP GET request, until the reply for that third GET request is received?

Since the client already have the object, the reply for the third GET request will not contain the object. Therefore, the end-to-end delay is 1 RTT.

3. Consider the figure below that shows the generalized forwarding table in a router. Now consider an arriving datagram with the IP source and destination address fields indicated below. For each source/destination IP address pair, indicate which rule is matched. **Note:** assume that a rule that is earlier in the table takes priority over a rule that is later in the table and that a datagram that matches none of the table entries is dropped.



a. Source: 1.2.56.32 Destination:128.116.40.186
Rule 2, with action drop

b. Source: 65.92.15.27 Destination: 3.4.65.76
Rule 1, with action forward(2)

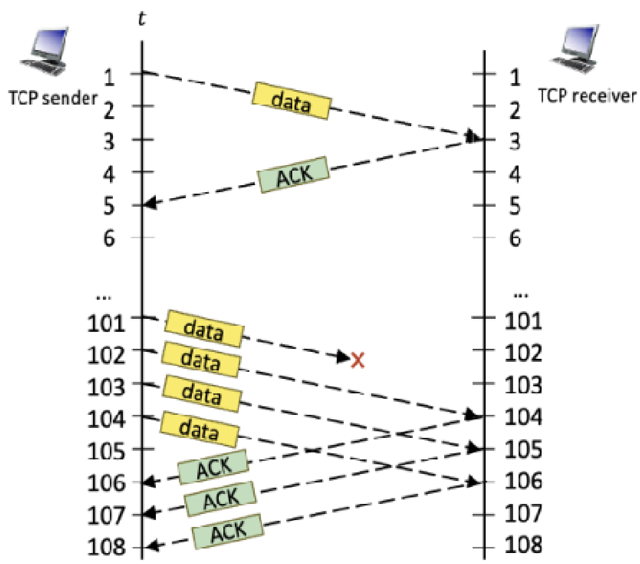
c. Source: 10.1.2.3 Destination: 7.8.9.2
Rule 3, with action send to controller

d. Source: 10.1.34.56 Destination: 54.72.29.90
No match to any rule. Will be dropped

4. Suppose that TCP connection begins at time t = 1 (no messages sent before t = 1) and the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 2 time units, and so the first segment arrives

at the receiver at t = 3. For TCP congestion control, AIMD mechanism is used for congestion avoidance after the TCP slow start.

Answer the following questions. Suppose that multiple packets can be sent at the same time. Mention clearly sequence numbers of data and ACK packets sent. You also need specify the actions related to the congestion window, including the congestion window size `cwnd`, whether the window base is moved, and if so, specify how much the window base is moved. Suppose that application layer generates data continuously, that means there are always packets to send if congestion window allows.



1) What is the sender action at t = 5 upon receipt of the ACK? (b)
   (a) cwnd = 2, move the window base forward by 2, and send new segments, as available and as allowed by the congestion window
   (b) cwnd = 2, move the window base forward by 1, and send new segments, as available and as allowed by the congestion window
   (c) Keep the congestion window size the same but send new segments, as available and as allowed by the congestion window.
   (d) Do nothing.
   (e) Send an ACK to the ACK.

2) Suppose that all the data and ACK packets sent before time t = 101 are correctly received (no unACKed data, no on-the-way ACK). At time t = 101, the data sequence number is 1,000, the TCP connection is in the slow start phase, and `cwnd` = 4. What is the sender

action at t = 108 upon receipt of the ACK? You can assume for this question that no timeouts have occurred. (b)

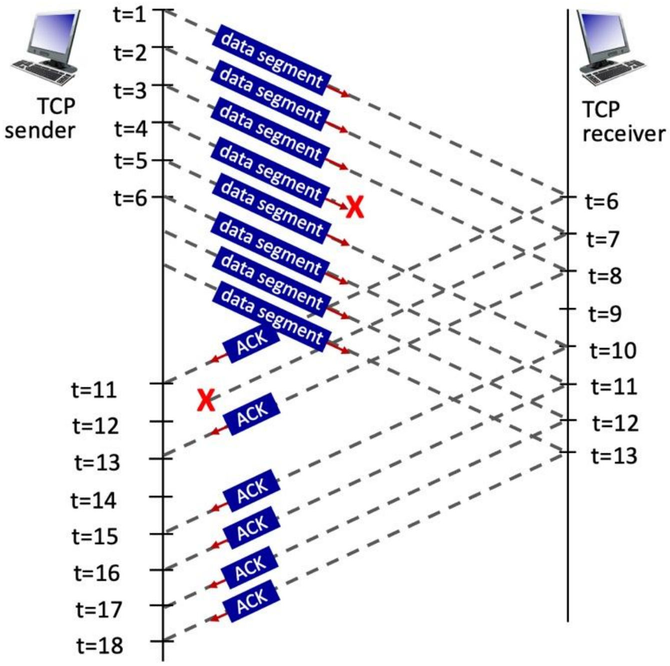
(a) cwnd = 1, do not move the window base, and retransmit the segment with sequence number 1000
(b) cwnd = 2, do not move the window base, and retransmit the segment with sequence number 1000
(c) cwnd = 3, do not move the window base, and retransmit the segment with sequence number 1000
(d) Inform the upper layer that the connection is terminated, and close the socket.
(e) Do nothing except increment the number of duplicate ACKs received by 1.

3) Based on the assumption in previous question 2), suppose that at t = 108 a timeout event occurs. What does the sender do? (a)

(a) cwnd = 1, do not move the window base, and retransmit the segment with sequence number 1000
(b) cwnd = 2, do not move the window base, and retransmit the segment with sequence number 1000
(c) cwnd = 3, do not move the window base, and retransmit the segment with sequence number 1000
(d) Inform the upper layer that the connection is terminated, and close the socket.
(e) Do nothing except increment the number of duplicate ACKs received by 1.

5. Suppose the initial value of the sequence number is 0 and every segment sent to the receiver each contains 100 bytes. The delay between the sender and receiver is 5 time units, and so the first segment arrives at the receiver at t = 6. The segment sent at t=4 is lost, as is the ACK segment sent at t=7. For TCP congestion control, AIMD mechanism is used for congestion avoidance after the TCP slow start. Suppose that `cwnd` = 8 and `ssthresh` = 18 at t =1.

Student ID: ________________________ Section Number: __________
Student name: ______________________



Answer the following questions. Suppose that multiple packets can be sent at the same time. Mention clearly sequence numbers of data and ACK packets sent. You also need specify the actions related to the congestion window, including the congestion window size `cwnd`, whether the window base is moved, and if so, specify how much the window base is moved. Suppose that application layer generates data continuously, that means there are always packets to send if congestion window allows.

a. What is the sender action at t = 11 upon receipt of the ACK?

cwnd = 9, move the window base forward by 1. Since only 7 packets are un-ACKed, the sender will send 2 new segments with seq# = 800 and seq# = 900

b. What is the sender action at t = 13 upon receipt of the ACK?

cwnd = 10, move the window base forward by 2. Since only 7 packets are un-ACKed, the sender will send 3 new segments with seq# = 1000 seq# = 1100, and seq# = 1200

c. What does the sender do at t=16? You can assume for this question that no timeouts have occurred.

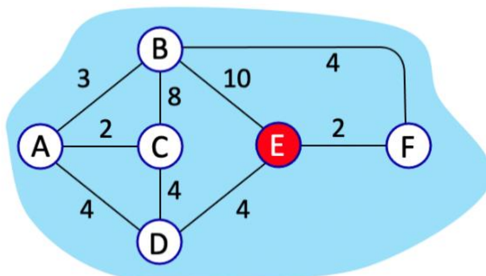
t = 16: cwnd = 10, do not move the window base. Do not send new segment. Increase the number of duplicate ACKs received by 1

d. What does the sender do at t=17? You can assume for this question that no timeouts have occurred.

Student ID: ________________________ Section Number: __________
Student name: ______________________

Detect three duplicate ACKs . cwnd = 5 , do not move the window base, and retransmit the segment with sequence number 300

6. Consider the network shown below, and Dijkstra's link-state algorithm to find the least cost path from source node E to all other destinations.



a. Complete the table below showing the link state algorithm's execution by providing the table entries with their $D(\cdot), p(\cdot)$ values for four first steps (i.e., initialization plus three iterations of main loop in this algorithm).

step	N’	D(A), p(A)	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(F), p(F)
0	E	∞	10, E	∞	4, E	2, E
1	E, F	∞	6, F	∞	4, E	
2	E, F, D	8, D	6, F	8, D		
3	E, F, D, B	8, D		8, D		
4	E, F, D, B, A			8, D		
5	E, F, D, B, A, C					

b. Now suppose that source node E has a packet to send to destination node A. What is the first node to which E will forward this packet on its path to A?

Node D

7. Suppose that TCP's current estimated values for the round trip time (estimatedRTT) and deviation in the RTT (DevRTT) are 300 msec and 13 msec, respectively. Suppose that the next two measured RTTs are 330 msec and 240 msec respectively. Compute TCP's new value of DevRTT, estimatedRTT, and the TCP timeout value after the last measured RTT values is obtained. Use the values of α = 0.125, and β =0.25. Round your answers to two decimal places after leading zeros.

DevRTT1 = (1-0.25)*13 + 0.25*(330-300) = 17.25
estimatedRTT1 = (1-0.125)*300 + 0.125*330 = 303.75

DevRTT1 = (1-0.25)*17.25 + 0.25*(303.75-240) = 28.88
estimatedRTT1 = (1-0.125)* 303.75+ 0.125*240 = 295.78

TCP timeout = 295.78 + 4*28.88 = 411.3