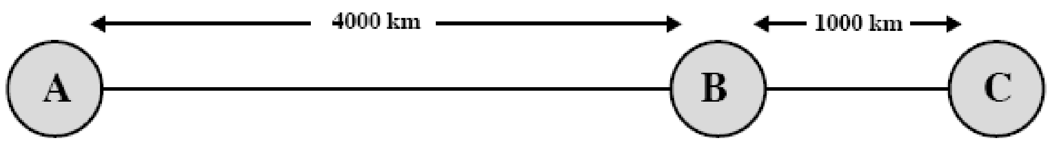
CMPT 371 ASSIGNMENT 3 SOLUTION

Please note that not all parts of all problems will be graded. Randomly chosen portions of the assignment totaling 100 points will be graded. Complete solutions to all parts of the assignment will be provided

a) (40 points) Frames are generated at node A and sent to node C through node B. Determine the minimum transmission rate required between nodes B and C so that the buffers of node B are not flooded, based on the following:
   - The data rate between A and B is 100 kbps
   - The propagation delay is 5 microseconds/km for both lines
   - There are full duplex lines between the nodes
   - All data frames are 1000 bits long: ACK frames are separate frames of negligible length
   - Between A and B a sliding window protocol with window size of 3 is used
   - Between B and C stop and wait is used
   - There are no errors.

   Hint: In order not to flood the buffers of B, the average number of frames entering and leaving B must be the same over a long interval.



Find the average number of frames entering B. The utilization of the A to B link will give us this information
The utilization for a sliding window protocol with a window size of W=3 is

$U = T_f / T_t = W/(2a+1)$ ***for*** $W<2a+1$

$U= 1$ ***for*** $W >= 2a+1$ ***(1 point for equation)***

The propagation time is T_{PROP} ***=5 microseconds/km * 4000 km = 0.02 s***
(1 for identifying value as Tprop 1 for answer)

The transmission time for a frame is T_{FRAME}***=1000bits/100000bps = 0.01 s***
(1 point each for identifying values of frame length and data rate, 1 point for selecting correct relation 1 point for answer)

$a = T_{PROP}/T_{FRAME} = 2$ ***so*** $2a+1=5$ ***and*** $U=3/5=0.6$
(2 points for giving/using equation for a, 1 point for correct value of a, 1 point for determining the utilization)

Since the utilization is 0.6 the number of frames entering B in time T is 0.6 times the number of frames transmitted in time T assuming continuous transmission.
T_{FRAME} ***is .01 s so***

On average 0.6*100=60 frames per second or 0.6* 6000=360 frames per minute are entering B.

The B to C link is using stop and wait flow control. For this link we wish to find the data rate R
The propagation time is T_{PROP} ***=5 microseconds/km * 1000 km = 0.005 s***

(1 point)
The transmission time for a frame is T_{FRAME} =1000bits/R
(1 point)

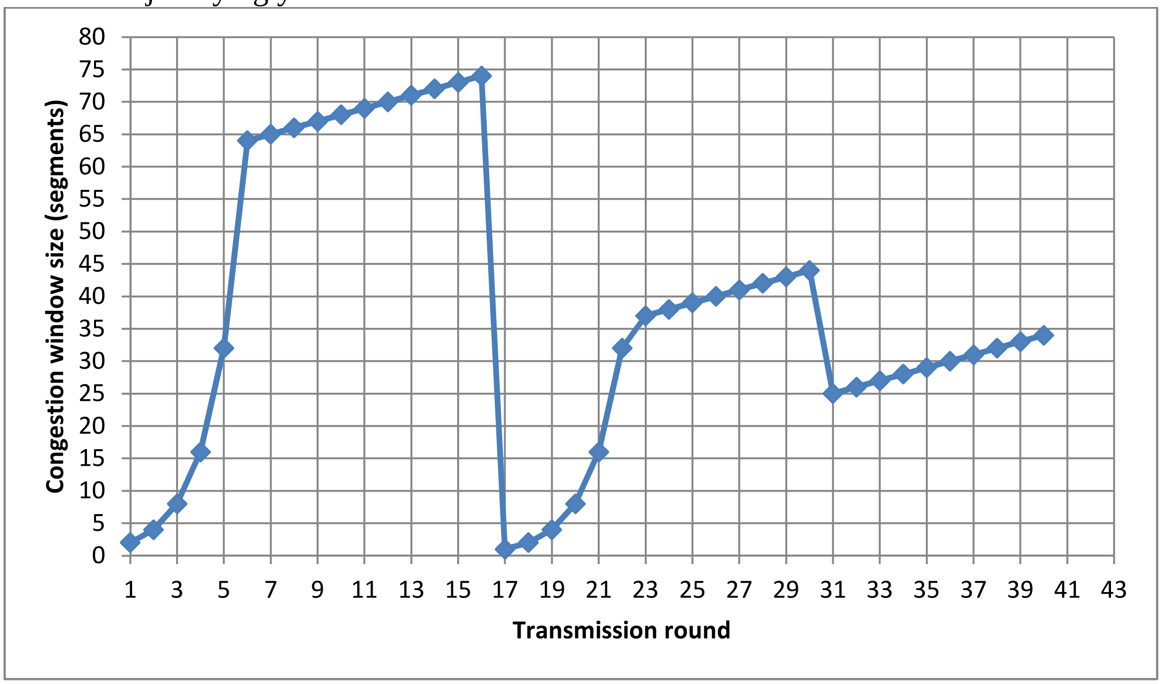
$a = T_{PROP}/T_{FRAME} = 5*10^{-6}R$ *(1 point)*

and $1+2a= 1+10^{-5}R$ *(1 point)*

The average rate that data leaves B is the Utilization of the B-C link multiplied by the data rate of the link. (1 point)
This should be equal to the average rate at which data is arriving at B, which is 60 frames per second or $6*10^4$*bps*
The utilization of the B-C link is(1 point)
$U=1/(1+2a)=1/(1+10^{-5}R)$
(1 point)

So
$UR = R/(1+10^{-5}R) = 6 * 10^4$ *(1 point)*
$R = 6*10^4 + 0.6R$
$R = 6 * 10^4 / 0.4 = 1.5 * 10^5$ *bps = 150 Kbps (1 point)*

2. Consider the figure shown below. Assuming TCP (Reno) is using congestion control (slow start and collision avoidance modes as discussed in class and in your text) answer the following questions. Assume the system has been running before this sample congestion window length data shown in the plot below was collected. For each question you should provide a short discussion justifying your answer.



a) [5 points] Identify the periods of time when TCP collision avoidance is operating
When collision avoidance is operating the length of the congestion window will increase by one segment after each round. Therefore, CA is operating in rounds 6-15 rounds 23-29 and rounds 31-39

b) [5 points] Identify the periods of time when TCP slow start is operating
When slow start is operating the congestion window will either increase by a factor of 2 or increase to the value of ssthresh. The smaller of the two increases will be made. Therefore, slow start is operating in rounds 1-5 and rounds 17-22. After round 21 the length of the congestion window increases to the value of ssthresh (half the value at which the previous loss event occurred).

c) [3 points] After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
The loss is detected by a timeout. When a timeout occurs in collision avoidance mode the value of ssthresh is cut in half and slow start mode is reinitialized (the length of the congestion window is set to 1). Slow start begins again. This does not occur for TCP Reno when a triple duplicate ACK occurs indicating a loss.

d) [3 points] After the 30st transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
The loss is detected by a triple duplicate ACK. In TCP Reno, when a triple duplicate ACK occurs in collision avoidance mode fast recovery is initiated. The value of ssthresh is cut in half and the length of the congestion window is set to the newly reduced value of ssthresh+3. Collision avoidance begins again.

e) [8 points] What is the value of ssthresh at the 4th transmission round? At the 12th transmission round? At the 20th round? At the 26th round? At the 32nd round?

For rounds 0-16 the value of ssthresh is 64. We can tell because when the length of the congestion window reaches 64 congestion avoidance mode is entered. Congestion avoidance mode is entered from slow start mode only when a loss event occurs, or the receive window size is reached, or ssthresh is reached. We cannot have reached the receive window size because the size of the congestion window continues to increase. There is no reduction in congestion window size indicating a loss event. Therefore we must have reached ssthresh.
After the loss event at round 16 ssthresh will be set to half the congestion window size when the loss event occurred. Therefore for rounds 17-30 ssthresh = 74/2 = 37. At round 30 the congestion window length is 44. The loss event at round 30 will cause TCP to enter fast recovery. The value ssthresh to be set to a value of 22+3=25 for the remaining illustrated rounds. So the ssthresh values requested are

round	ssthresh
4	64
12	64
20	37
26	37
32	25

f) [4 points] During what transmission round is the 70th segment sent? The 320th segment?
Each round the number of segments sent is indicated on the plot. Add these numbers until we find which round the desired number of segments is reached
2+4+8+16+32 < 70 2+4+8+16+32+64 > 70 segment 70 is sent in round 6
2+4+8+16+32+64+65+66 < 320
2+4+8+16+32+64+65+66+67 > 320 segment 320 is sent in round 9

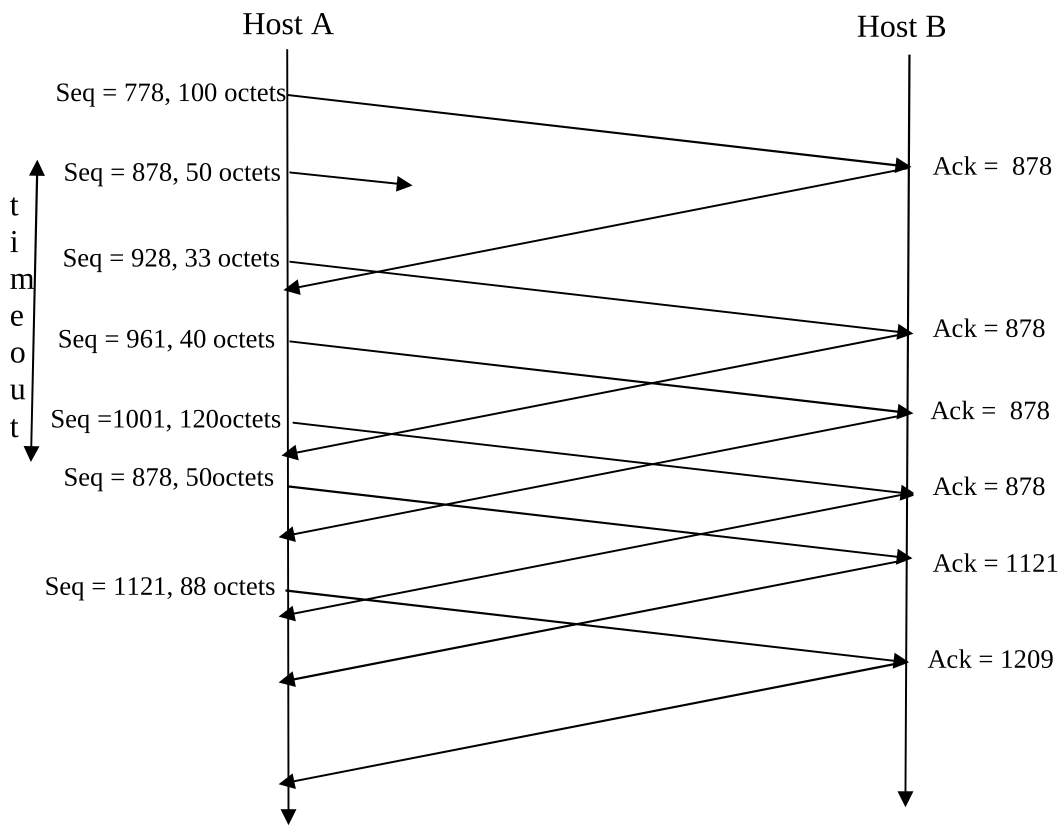
g) [4 points] Assuming a packet loss is detected after the 40th round by the receipt of a triple duplicate ACK. What will be the resulting values of the length of the congestion window and of ssthresh?
At round 40 the value of ssthresh is 34, 34/2 = 17 will be the new ssthresh
The new length of the congestion window will be 17+3=20

h) [6 points] Suppose TCP Tahoe is used (instead of TCP Reno). Assume that triple duplicate ACKs are received during the 27th round. What are the ssthresh and the congestion window size at the 32nd round?
At the 27th round the reception of 3 duplicate ACKs will cause TCP Tahoe to set the ssthresh to ½ of the current congestion window size. The congestion window will be set to 1. Then move to SS mode. So the values of congestion window length and ssthresh are
27th round ssthresh 41/2=20
28th round ssthresh 20 congestion window length 1
29th round ssthresh 20 congestion window length 2
30th round ssthresh 20 congestion window length 4
31st round ssthresh 20 congestion window length 8
32nd round ssthresh 20 congestion window length 16
33rd round ssthresh 20 congestion window length 20

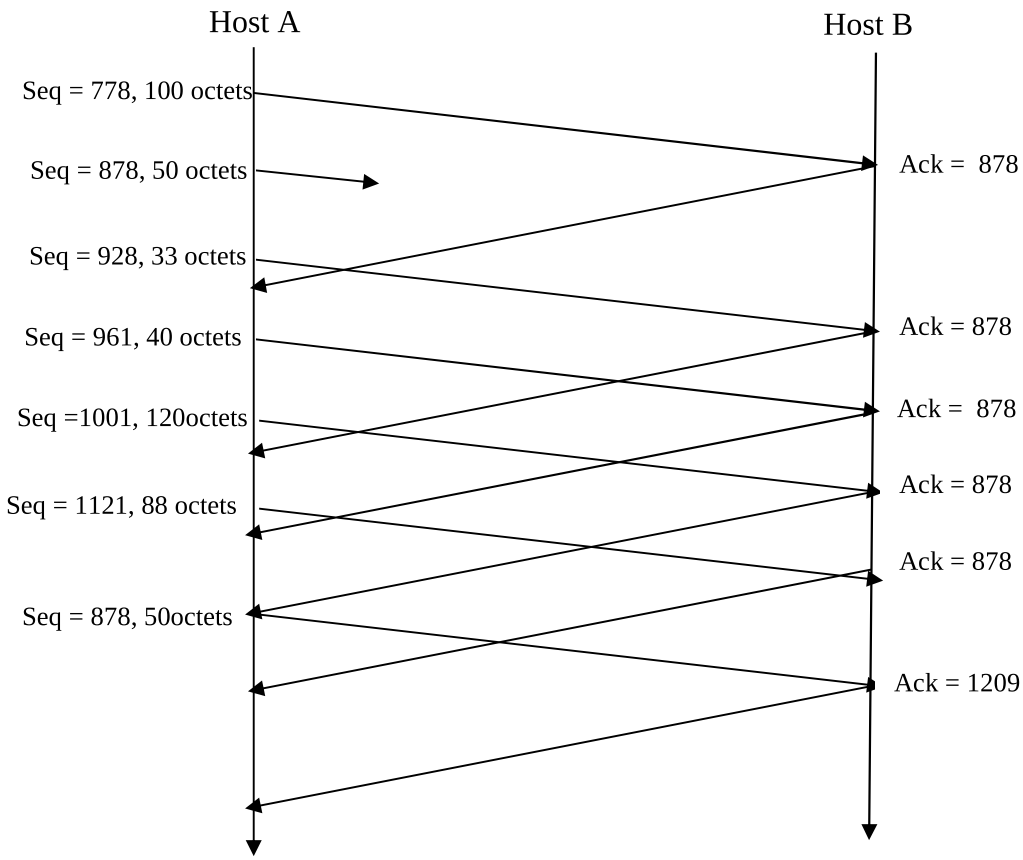
i) [2 points] For the case in part h), how many segments are sent in rounds 27 to 32
1+2+4+8+16 = 31

3. Consider two hosts A and B who are communicating over an existing TCP connection. Host B sends a cumulative ACK whenever it receives data from host A. Host A sends a cumulative ACK whenever it receives data from host B. Up to 1000 octets can be sent before an ACK is required. (sliding window buffer of 1000 octets)
First consider data flowing from host A to host B. Host B has already received all octets up to and including octet 777 from host A. Host A has sent all octets up to and including octet 777. Host A has not yet sent any octets with octet numbers larger than 777. Host A then sends a series of segments with the following numbers of octets, segment #1 has 100 octets, segment #2 has 50 octets, segment #3 has 33 octets, segment #4 has 40 octets, segment #5 has 120 octets and segment #6 has 88 octets.
   a) [3 points] What is the sequence number in each of the first 3 segments (segments #1, #2 and #3)?
      Sequence number segment #1 778
      Sequence number segment #2 878
      Sequence number segment #3 928
   b) [3 points] Assume the segments arrive in the order they were sent and that all ACKs reach host A. What is the acknowledgment number in each of the first 3 acknowledgements (ACKs sent when segments #1, #2 and #3 arrive at host B)?
      Acknowledgement number segment #1 878
      Acknowledgement number segment #2 928
      Acknowledgement number segment #3 961
   c) [3 points] Assume the segments arrive in the order segment #2, segment #3, segment #1 and that all ACKs reach host A. What is the acknowledgment number in each of the first 3 acknowledgements (ACKs sent when segments #2, #3 and #1 arrive at host B)?
      Acknowledgement number segment #1 778
      Acknowledgement number segment #2 778
      Acknowledgement number segment #3 961
   d) [6 points] Assume the segments arrive in the order segment #1, segment #2, and segment #3. Also assume that the ACKs for segments 1 and 3 reach host A and the ACK for segment 2 does not reach host A. What is the acknowledgment number in each of the first 3 acknowledgements. Is segment 2 retransmitted? Why or why not?
      Acknowledgement number segment #1 778
      Acknowledgement number segment #3 961
      Segment 2 is not retransmitted. Cumulative ACKs are used. This means that if we receive an ACK that indicates a particular acknowledgement number al octets up to (but not including) that octet have been successfully received by the destination. Even if an earlier ACK has not been received the source knows that the data that ACK would have acknowledged has in fact been received.

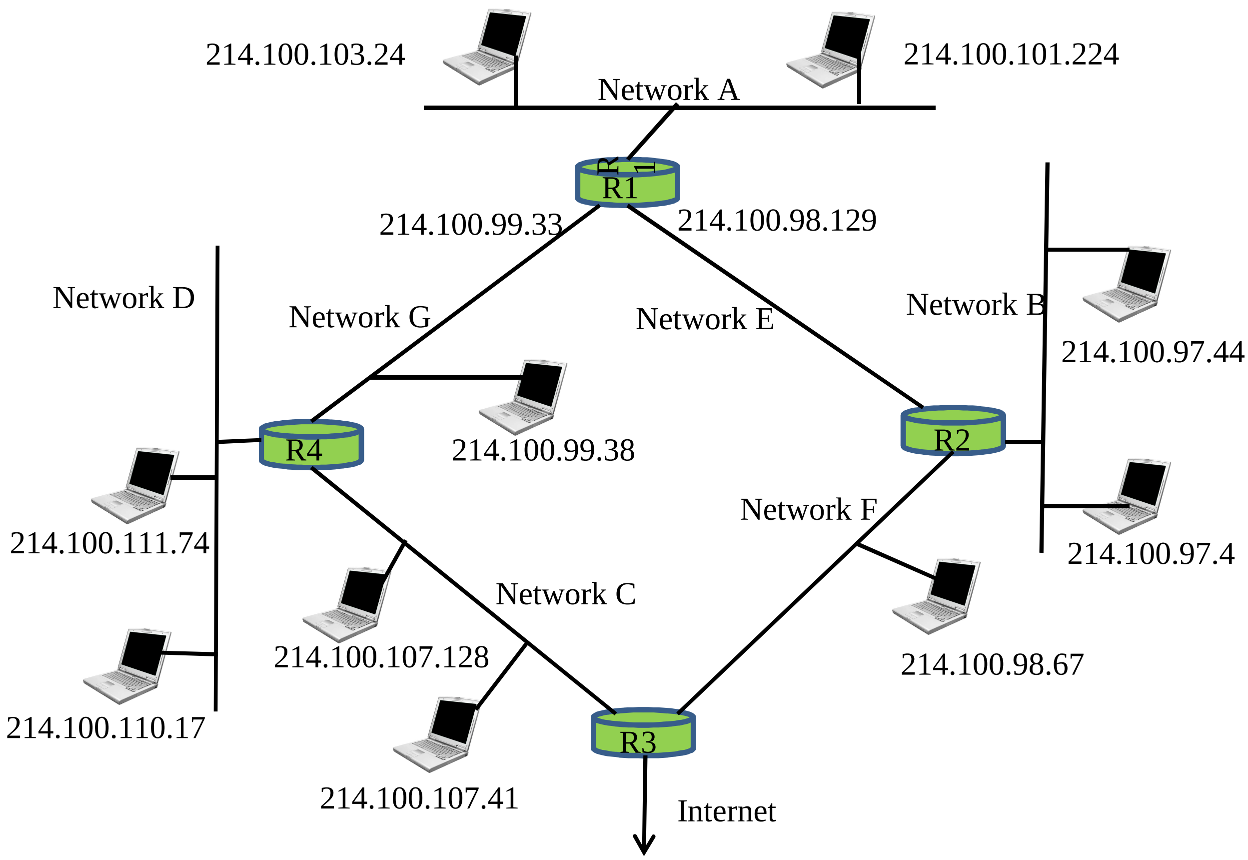
e) [12 points] Suppose segments that are not lost arrive at host B in the order that host A sent them. Also suppose host A uses a retransmission timeout of RTO seconds. Host A should receive an ACK in the first RTO seconds after it transmitted the segment. If host A does not receive the ACK before RTO seconds have elapsed host A will retransmit the segment. Assume the second segment is lost. All other segments, all retransmitted segments and all ACKs arrive and are not corrupted. The value of RTO (the length of the timeout) is such that the timeout occurs after host A transmits segment 5 but before host A transmits segment 6. Draw a timing diagram, showing the segments and acknowledgments sent. For each segment in your figure provide the sequence number and the number of octets of data in the segment; for each ACK show the acknowledgement number. Continue your diagram until all six segments have been acknowledged successfully.



f) [7 points] Suppose segments that are not lost arrive at host B in the order that host A sent them. Also suppose host A and host B both use fast retransmit. Assume the second segment is also lost. All other segments, retransmitted segments, and ACKs arrive and are not corrupted. Draw a timing diagram, showing the segments and acknowledgments sent. For each segment in your figure provide the sequence number and the number of bytes of data; for each ACK show the acknowledgement number.



4. Consider the network topology shown below. Each network has one or two interfaces that connect to routers. On each network the router with the smallest router number (R1 <R2<R3<R4) will have the lowest possible address for a host in the address block allocated for that network. If there are two routers attached to the network, the address of the second router (with the router with the larger router number) will have the second lowest possible address in the block of addresses allocated for the network. Each router has more than one interface. Assign each router interface an Ethernet interface number (eth1, eth2, …) using the following constraint. Start at 12 O'clock move clockwise, the first interface encountered is eth1, the second eth2 and the third is eth3.



a) [18 points] Assign network addresses to each of the seven subnets, with the following constraints: All addresses must be allocated from 214.100.96/20: Subnet A should have enough addresses to support 512 interfaces: Subnet B should have enough addresses to support 120 interfaces: Subnet C should have enough addresses to support 500 interfaces: Subnet D should have enough addresses to support 333 interfaces: Subnets E should have enough addresses to support 2 interfaces: Subnets F, and G should each be able to support 3 interfaces: Express the address blocks in the form xxx.xxx.xxx.xxx/yy where xxx.xxx.xxx.xxx is the network address of the block of addresses and yy indicates the prefix (the number of binary digits in the network address). Briefly explain how you determined the network address and the length of the prefixes.

Network A	**214.100.100.0 / 22**	**1022 interfaces**	**512 OK**
Network B	**214.100.97.0 / 25**	**126 interfaces**	**120 OK**
Network C	**214.100.106.0 / 23**	**510 interfaces**	**500 OK**

Network D	**214.100.110.0/ / 23**	**510 interfaces**	**333 OK**
Network E	**214.100.98.128 / 30**	**2 interfaces**	**2 OK**
Network F	**214.100.98.64 / 29**	**6 interfaces**	**3 OK**
Network G	**214.100.99.32 / 29**	**6 interfaces**	**3 OK**

A) To determine the network address for Network A, know the network will use 1024 addresses so 2^{10} addresses 32-10 = 22, so prefix length is 22. The address block must start on a boundary that is a multiple of 1024 addresses, either 214.100.96.0, 214.100.100.0, 214.100.104.0, or 214.100.108.0. The addresses on the illustrated network are 214.100.101.224, and 214.100.103.24. Both these addresses fall into the 1024 length block starting at 214.100.100.0. So our network address is 214.100.100.0 / 22

B) To determine the network address for Network B, know the network will use 128 addresses so 2^7 addresses 32-7 = 25, so prefix length is 25, Address block must start on a boundary that is a multiple of 128 addresses, that is 214.100.96.0, 214.100.96.128, 214.100.97.0, 214.100.97.128, … 214.100.127.0.0, 214.100.127.128.0. The addresses on the illustrated network are 214.100.97.44, and 214.100.97.4.. Both these addresses fall into the 128 length block starting at 214.100.97.0. So our network address is 214.100.97.0 / 25

C) To determine the network address for Network C, know the network will use 512 addresses so 2^9 addresses 32-9 = 23, so prefix length is 23. The address block must start on a boundary that is a multiple of 512 addresses, either 214.100.96.0, 214.100.98.0, 214.100.100.0 … 214.100.110.0. The addresses on the illustrated network are 214.100.107.128 and 214.100.107.41. Both these addresses fall into the 512 length block starting at 214.100.106.0. So our network address is 214.100.106.0 / 23

D) To determine the network address for Network D, know the network will use 512 addresses so 2^9 addresses 32-9 = 23, so prefix length is 23. The address block must start on a boundary that is a multiple of 512 addresses, either 214.100.96.0, 214.100.98.0, 214.100.110.0 … 214.100.110.0. The addresses on the illustrated network are 214.100.110.17 and 214.100.111.74. Both these addresses fall into the 512 length block starting at 214.100.110.0. So our network address is 214.100.110.0 / 23

E) To determine the network address for Network E, know the network will use 4 addresses so 2^2 addresses 32-2 = 30, so prefix length is 30. The address block must start on a boundary that is a multiple of 4 addresses, either 214.100.96.0, 214.100.96.4, 214.100.96.8 … 214.100.111.252. The address on the illustrated network is 214.100.98.129. This address falls into the 4 length block starting at 214.100.98.128. So our network address is 214.100.98.128 / 30

F) To determine the network address for Network F, know the network will use 8 addresses so 2^3 addresses 32-3 = 29, so prefix length is 29. The address block must start on a boundary that is a multiple of 8 addresses, either 214.100.96.0, 214.100.96.8, 214.100.96.16 … 214.100.111.240. The address on the illustrated network is 214.100.98.67. This address falls into the 8 length block starting at 214.100.98.64. So our network address is 214.100.98.64 / 29

G) To determine the network address for Network F, know the network will use 8 addresses so 2^3 addresses 32-3 = 29, so prefix length is 29. The address block must

start on a boundary that is a multiple of 8 addresses, either 214.100.96.0, 214.100.96.8, 214.100.96.16 … 214.100.111.240. The address on the illustrated network is 214.100.99.38. This address falls into the 8 length block starting at 214.100.99.32. So our network address is 214.100.99.32 / 29

b) Using your answer to part (a) provide the routing tables for each of the three routers in the form shown below. Assume that the route taken will pass through the smallest possible number of routers. Also assume that if there are two different paths through the same number of routers the chosen path will send the packet clockwise through the networks. Be sure that your routing table will forward packets with any legal global IP address. Be sure the entries in your routing table are ordered so the longest match is first and the shortest match is last (mask with most 1's first, mask with fewest 1's last).

Network address	Final address in block	Gateway address	interface
214.100.98.128	214.100.98.131	*	Eth2

•
•
•

R1

Network address	Final address in block	Gateway address	interface
214.100.98.128	214.100.98.131	*	Eth2
214.100.98.64	214.100.98.71	214.100.98.130	Eth2
214.100.99.32	214.100.99.39	*	Eth3
214.100.97.0	214.100.97.127	214.100.98.130	Eth2
214.100.106.0	214.100.107.255	214.100.99.34	Eth3
214.100.110.0	214.100.111.255	214.100.99.34	Eth3
214.100.100.0	214.100.103.255	*	Eth1
0.0.0.0		214.100.98.130	Eth2

R2

Network address	Final address in block	Gateway address	interface
214.100.98.128	214.100.98.131	*	Eth3
214.100.98.64	214.100.98.71	*	Eth2
214.100.99.32	214.100.99.39	214.100.98.129	Eth3
214.100.97.0	214.100.97.127	*	Eth1
214.100.106.0	214.100.107.255	214.100.98.66	Eth2
214.100.110.0	214.100.111.255	214.100.98.66	Eth2
214.100.100.0	214.100.103.255	214.100.98.129	Eth3
0.0.0.0		214.100.98.66	Eth2

R3

Network address	Final address in block	Gateway address	interface
214.100.98.128	214.100.98.131	214.100.98.65	Eth1
214.100.98.64	214.100.98.71	*	Eth1
214.100.99.32	214.100.99.39	214.100.106.2	Eth3
214.100.97.0	214.100.97.127	214.100.98.65	Eth1

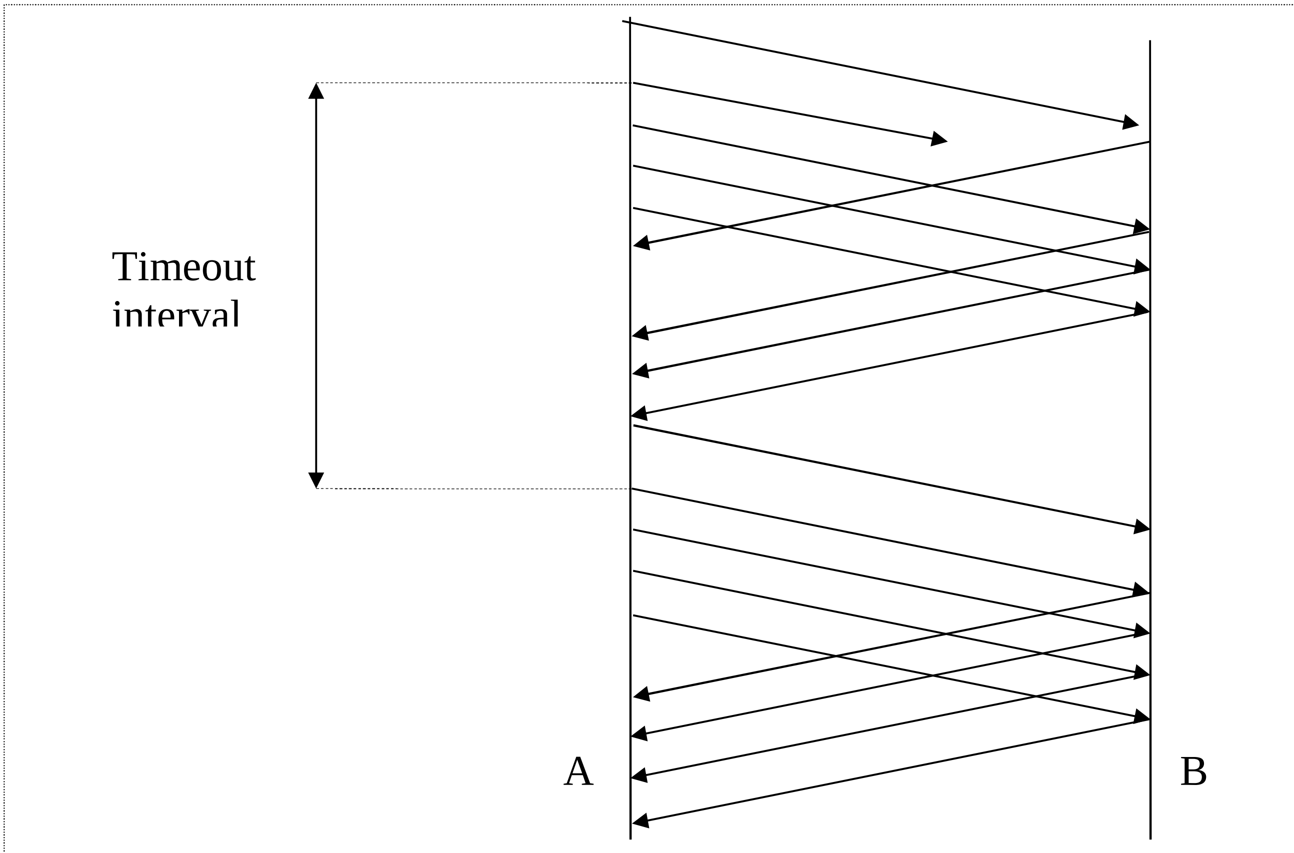
214.100.106.0	214.100.107.255	*	Eth3
214.100.110.0	214.100.111.255	214.100.106.2	Eth3
214.100.100.0	214.100.103.255	*	Eth3
0.0.0.0		*	Eth2

R4

Network address	Final address in block	Gateway address	interface
214.100.98.128	214.100.98.131	214.100.99.33	Eth1
214.100.98.64	214.100.98.71	214.100.106.1	Eth2
214.100.99.32	214.100.99.39	*	Eth1
214.100.97.0	214.100.97.127	214.100.99.33	Eth1
214.100.106.0	214.100.107.255	*	Eth2
214.100.110.0	214.100.111.255	*	Eth3
214.100.100.0	214.100.103.255	214.100.99.33	Eth1
0.0.0.0		214.100.106.1	Eth2

5. [40 points total, 36 for part a) 4 for part b)] P37 page 295 of text
   a) In addition to what is requesting in part a) of problem P37 please draw and explain a timing diagram for each protocol (GBN, SR, TCP). You diagram should begin with the sending of the first of the five consecutive segments containing data. You diagram should end after the sender has received acknowledgements of the data in all 5 segments. You should assume that all segments will be transmitted before the first ACK is received. Show the timeout interval on your diagram. You DO NOT need to label each data segment and ACK with the seq number and ack number. After your diagram explain step by step which segment is sent and which ACK is received from the point of view of the host A. Then answer the question in part a)

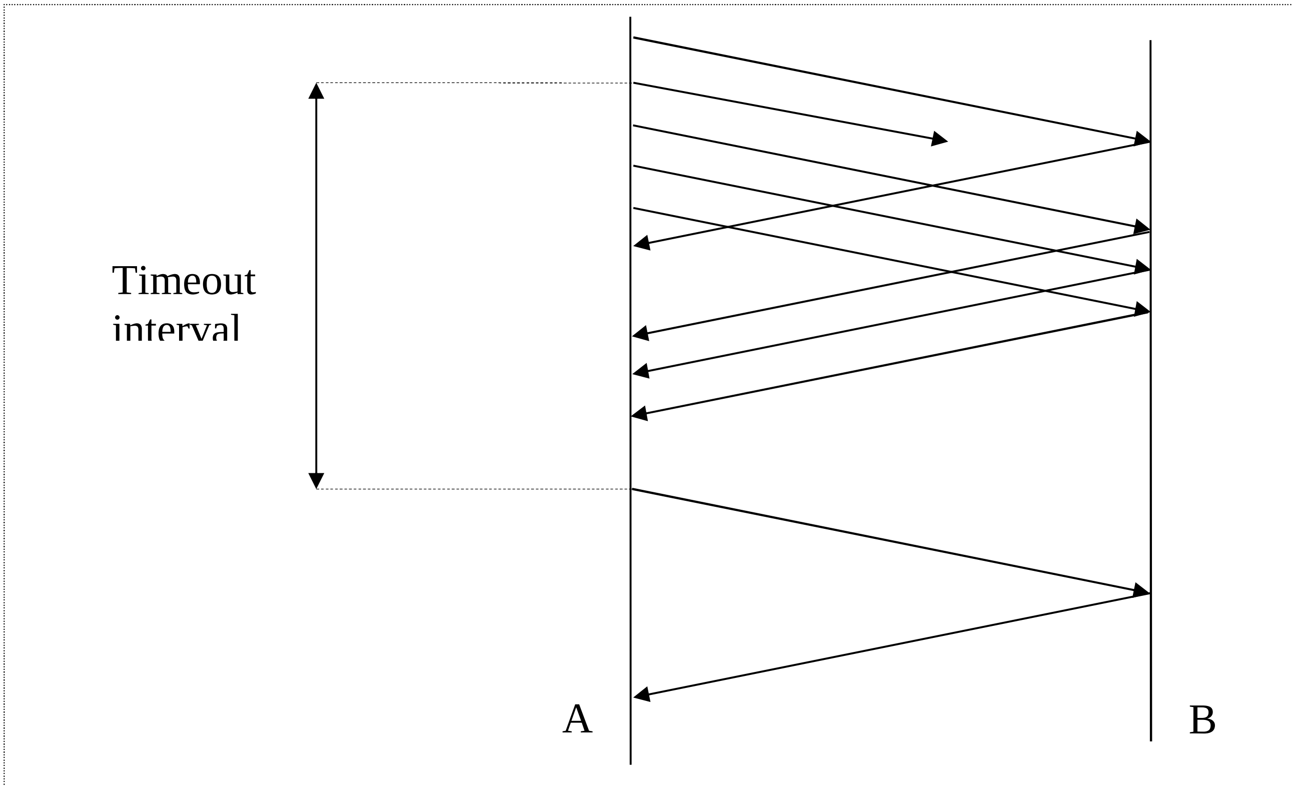
GoBackN: [12 points]



7

A sends data segment 1, then sends data segment 2, then sends data segment 3 then sends data segment 4 then data segment then sends data segment 5. A then receives an ACK for segment 2 (says A now expects to receive segment 2). Because segment 2 was lost A then receives three more ACKS for segment 1. After the timeout segment 2, 3, 4, and 5 are resent (in that order). Then ACK for segments 3, 4, 5, and 6 are then received
A sends a total of 9 segments
A receives a total of 9 ACKs

Selective Repeat: [12 points]

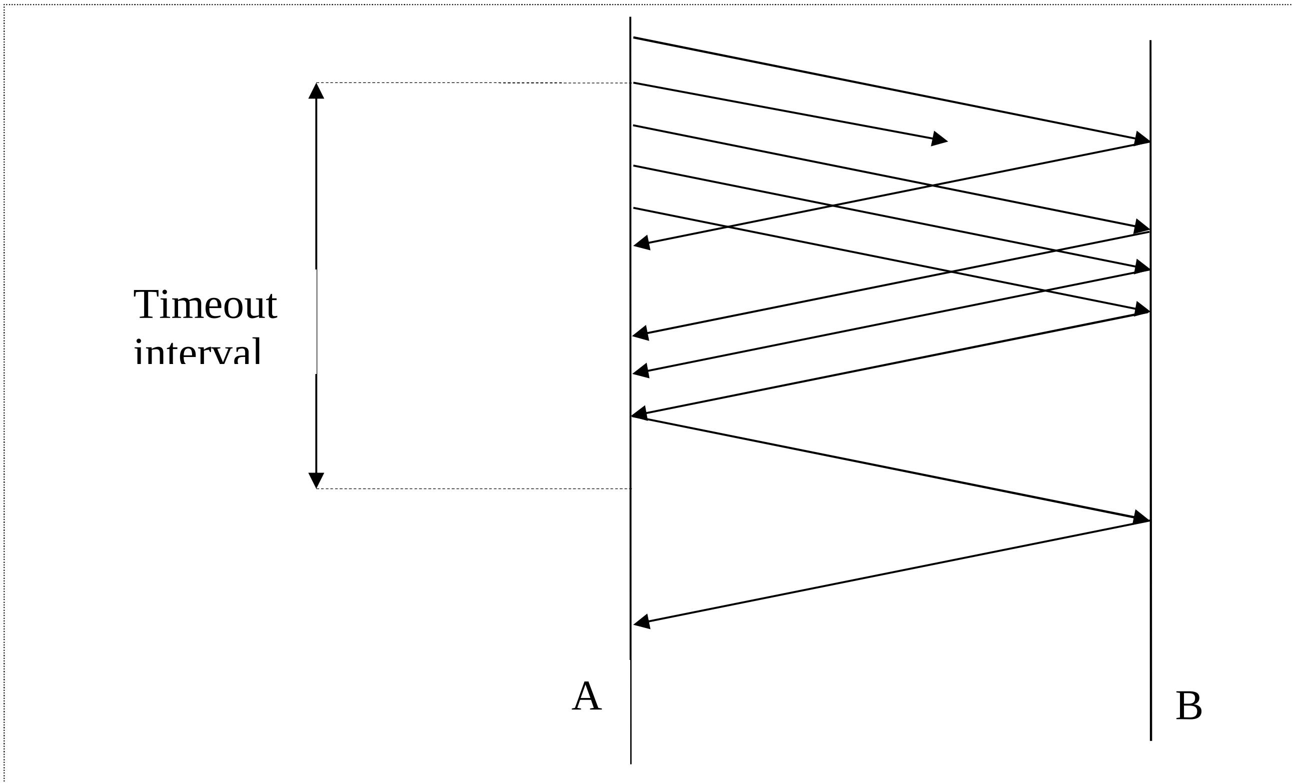


A sends data segment 1, then sends data segment 2, then sends data segment 3 then sends data segment 4 then data segment then sends data segment 5. A then receives an ACK for segment 1 (says A has received segment 1). Because segment 2 was lost A then receives three more ACKS for segments 3, 4 and 5.. After the timeout segment 2 is resent. Then ACK for segments 2 is received

A sends a total of 6 segments

A receives a total of 5 ACKs

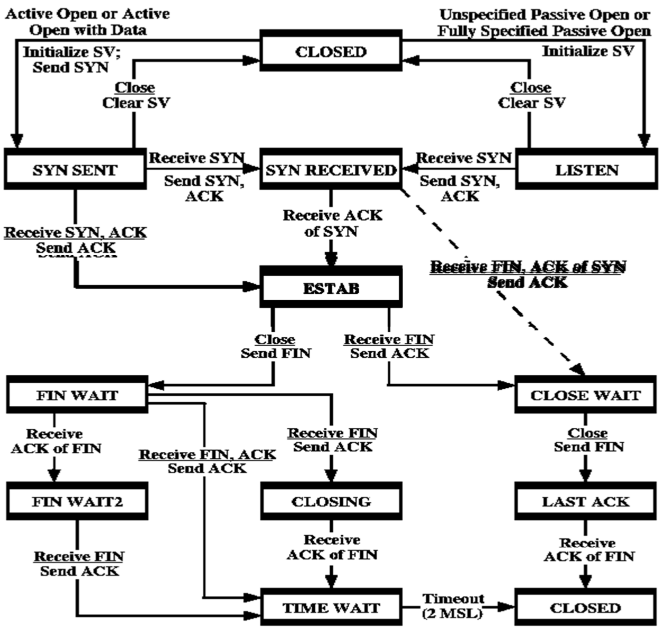
TCP: [12 points]



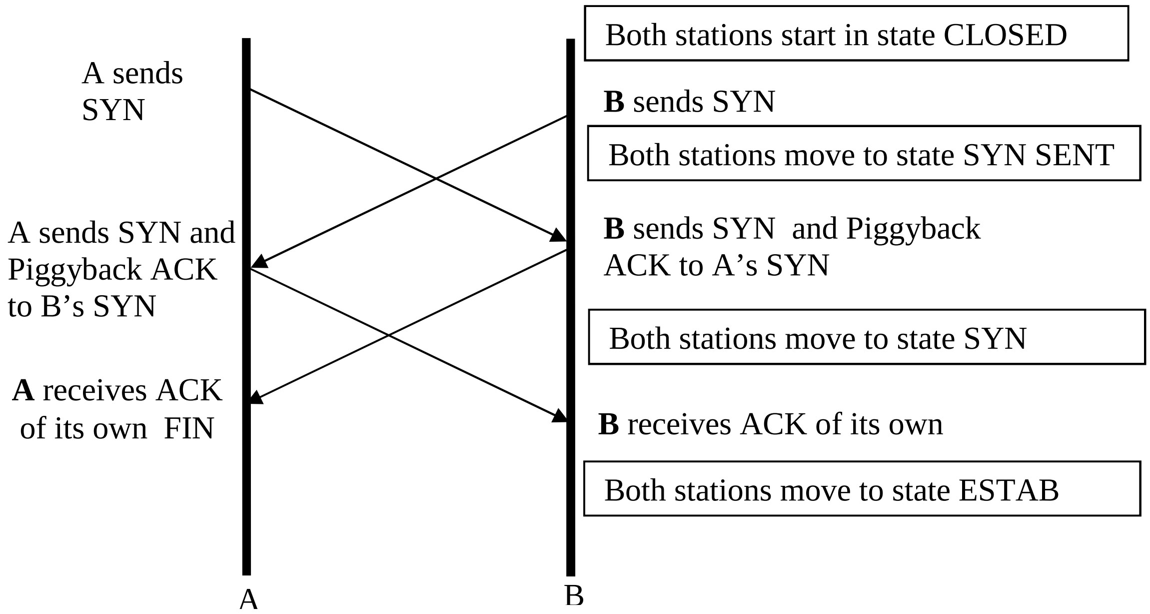
A sends data segment 1, then sends data segment 2, then sends data segment 3 then sends data segment 4 then data segment then sends data segment 5. A then receives an ACK for segment 2 (says A is expecting to receive segment 2 next). Because segment 2 was lost A then receives three more ACKS for segment 2.. When the third repeated ACK for segment 1 is received then TCP fast retransmit is activated and segment 2 is retransmitted. It is not necessary to wait until the timeout timer expires before resending segment 2. Finally A receives a final ACK for segment 6, saying it is ready to receive data in the next segment.
A sends a total of 6 segments
A receives a total of 5 ACKs
.

b) [4 points] TCP. This is because TCP uses fast retransmit without waiting until time out

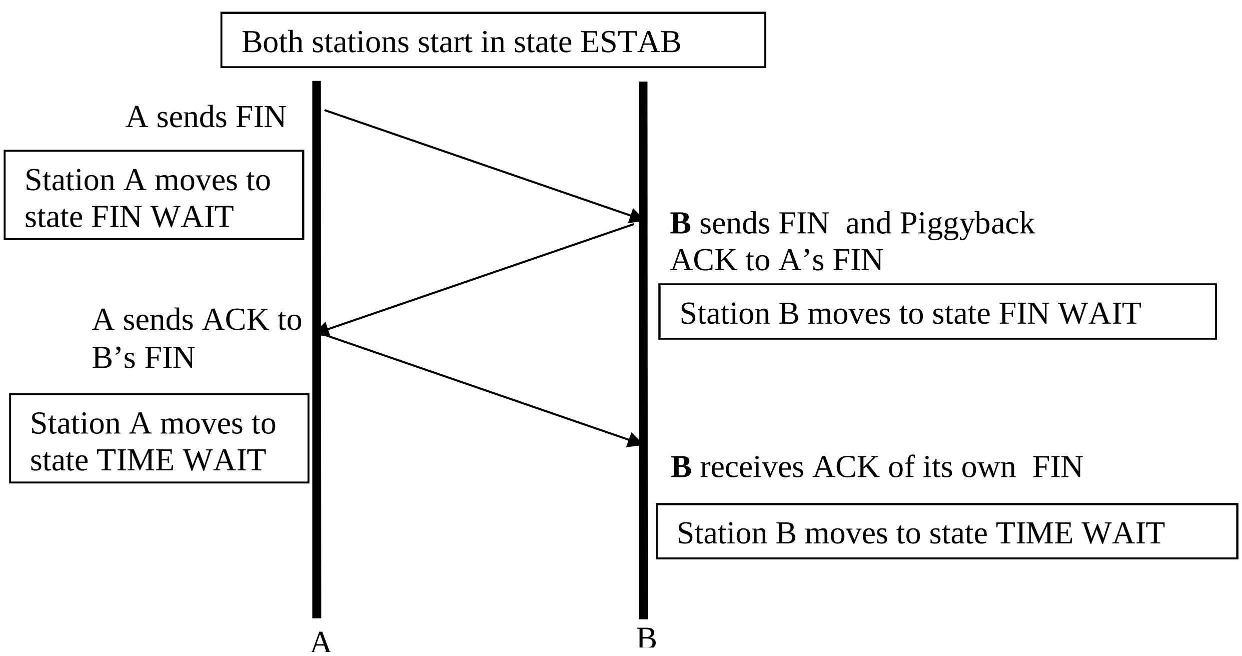
6. Refer to the TCP state machine in the class notes (given below)
   a. [4 points] The TCP state transition diagram allows for the case where the two stations issue a SYN segment at nearly the same time. This is called a simultaneous open. Draw the sequence of segment exchanges and the sequence of states that are transitioned through by the two stations for a simultaneous open.
   b. [12 points] Assume that a TCP connection between stations A and B has beer se. This connection was established using an active open. Explain how the co can be closed using an active close. To help you explain draw the series of pack ged during the active close for each of the three possible paths through the state machine (from stat ESTAB to state CLOSED).

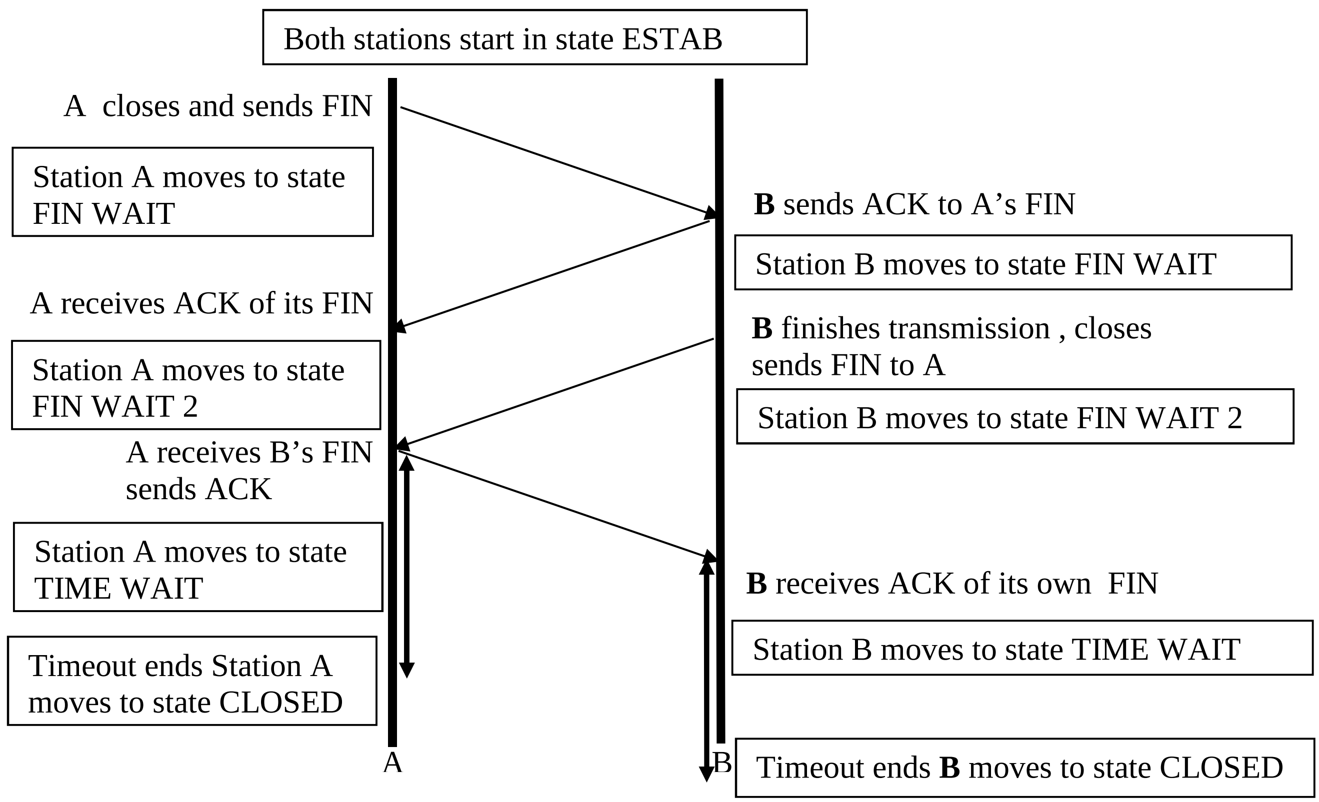


SV = state vector
MSL = maximum segment lifetime

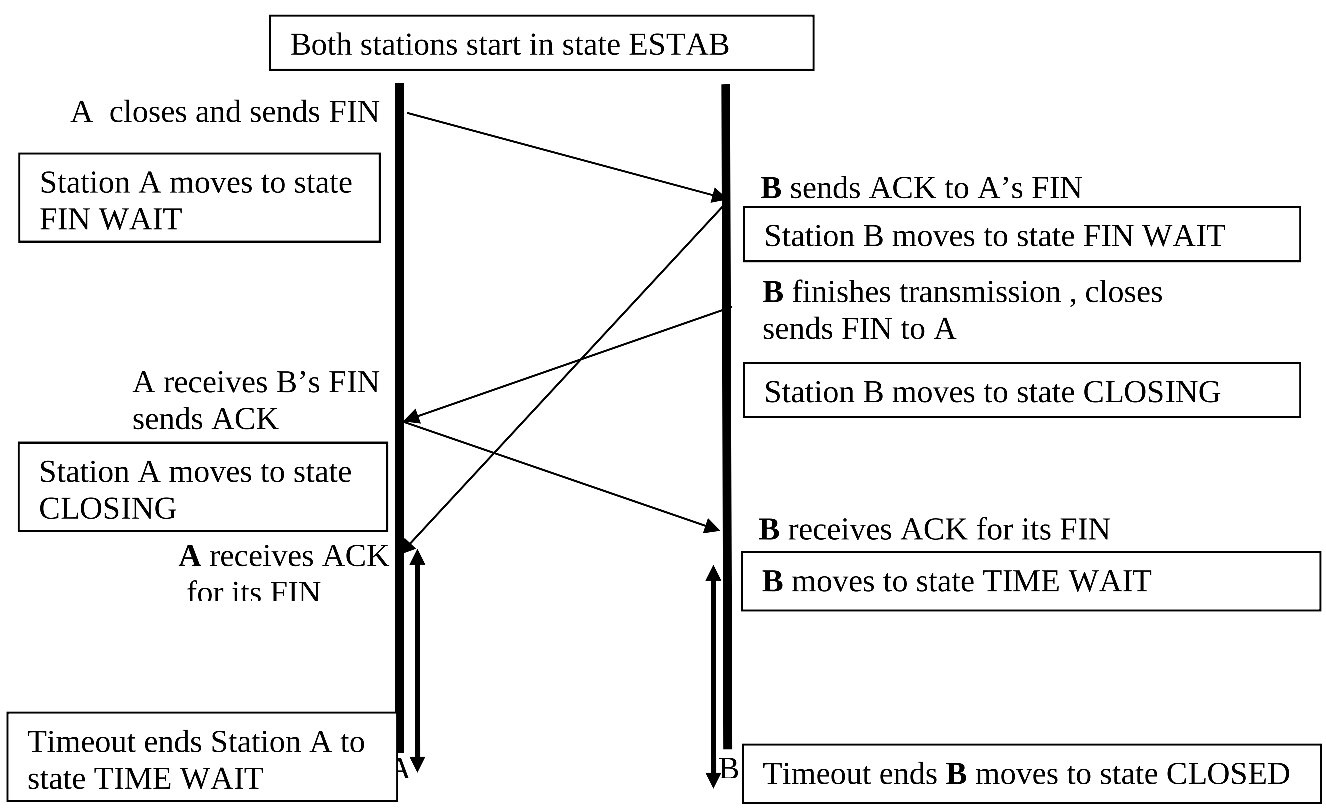

Both stations start in state CLOSED
A sends SYN
B sends SYN
Both stations move to state SYN SENT
A sends SYN and Piggyback ACK to B's SYN
B sends SYN and Piggyback ACK to A's SYN
Both stations move to state SYN
A receives ACK of its own FIN
B receives ACK of its own
Both stations move to state ESTAB
A
B

b)


Both stations start in state ESTAB
A sends FIN
Station A moves to state FIN WAIT
B sends FIN and Piggyback ACK to A's FIN
Station B moves to state FIN WAIT
A sends ACK to B's FIN
Station A moves to state TIME WAIT
B receives ACK of its own FIN
Station B moves to state TIME WAIT
A
B



Both stations start in state ESTAB
A closes and sends FIN
Station A moves to state FIN WAIT
B sends ACK to A's FIN
Station B moves to state FIN WAIT
A receives ACK of its FIN
B finishes transmission , closes sends FIN to A
Station A moves to state FIN WAIT 2
Station B moves to state FIN WAIT 2
A receives B's FIN sends ACK
Station A moves to state TIME WAIT
B receives ACK of its own FIN
Station B moves to state TIME WAIT
Timeout ends Station A moves to state CLOSED
A
B
Timeout ends B moves to state CLOSED



Both stations start in state ESTAB
A closes and sends FIN
Station A moves to state FIN WAIT
B sends ACK to A's FIN
Station B moves to state FIN WAIT
B finishes transmission , closes sends FIN to A
A receives B's FIN sends ACK
Station B moves to state CLOSING
Station A moves to state CLOSING
B receives ACK for its FIN
A receives ACK for its FIN
B moves to state TIME WAIT
Timeout ends Station A to state TIME WAIT
A
B
Timeout ends B moves to state CLOSED