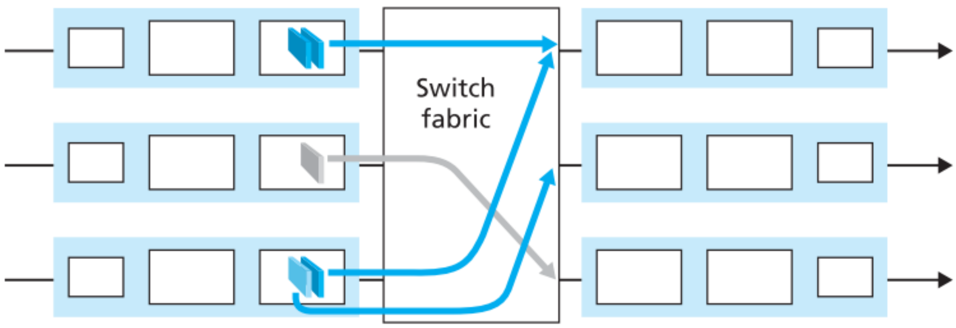
Final Exam

Computer Networks Fall 2018

Prof. Cheng-Fu Chou

<u>Question 1: ``Quickies''(20%)</u> Answer each of the following questions *briefly, i.e., in at most a few sentences.*

a) (5%) Give a pictorial example of head-of-the-line blocking in a router.



b) (5%) What is meant by network-assisted congestion control? Give an example of a congestion control protocol that takes this approach. Does TCP take this approach? Explain your answer.

Ans: In *network-assisted* congestion control, routers will explicitly signal the transport layer sender or receiver (by setting a bit in a passing data packet, or by an explicit message to the sender) to indicate congestion. ATM uses this approach. In *end-to-end* congestion control, the routers play no role -- the end-systems infer congestion from the observed network behavior. TCP uses lost packets to infer network congestion.

c) (5%) Which protocol – Go-Back-N or Selective-Repeat - makes more efficient use of network bandwidth? Why?

Answer: Selective repeat makes more efficient use of network bandwidth since it only retransmits those messages lost at the receiver (or prematurely timed out). In go-back-N, the sender retransmits the first lost (or prematurely timed out) message as well as all following messages (without regard to whether or not they have been received).

d) (5%) Consider the BGP protocol, an autonomous system (AS) A, and some destination network X. How does A control whether or not other autonomous systems route traffic destined to X through A?

Ans: By the paths that it advertises to its neighboring ASs. If A never advertises a path to X to its neighbors, the neighbors will never route to X via A.

Question 2: Transport Layer Potpourri (20%)

a) (5%) Suppose the sender and receiver in a pipelined reliable data transfer protocol have a window of size *N*. Suppose the sequence number of the segment at the base of the window at the receiver is *x*. What is the possible range of sequence numbers in the sender's window? Justify your answer.

Answer: If the receiver window base is *x*, then it has acked the previous N packets. However, those ACKs may not have been received by the sender. Hence the sender window could still be *[x-N,x-1]*. On the other hand, if all of the ACKs have been received at the sender, then the sender's window would be *[x,x+N-1]*. Hence the range is *[x-N,x+N-1]*.

b) (5%) Suppose that a reliable data transfer protocol such as TCP has an RTT estimate that is far too small. What is the consequence of having too small an estimate of the RTT? Now suppose the RTT estimate is far too large. What is the consequence?

Answer: if the RTT estimate is far too small, the retransmission timer will be too small and will timeout prematurely – resulting in the retransmission of packets that may not be lost. If the RTT estimate is far too large, the retransmission timer will be too large. In this case, the packet will not be transmitted until a long time after it has been lost – resulting in long error recovery delays.

c) (5%) What is the role of the *threshold* in TCP congestion control? How does it change over time?

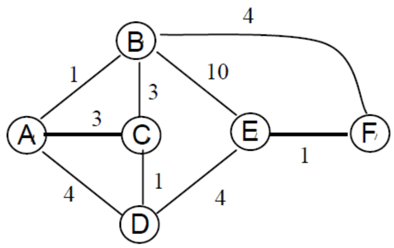
Answer: When the TCP window, W, is larger than the threshold, TCP is in congestion avoidance and grows the window slowly (at the rate of 1/W per packet acknowledged, or equivalently, at the rate of 1 packet per RTT). Note that on packet loss the threshold value is halved.

d) (5%) Consider two TCP senders. They are at different sending hosts and go to different destinations, but pass through a *common* bottleneck link (that is the only bottleneck link on either of their paths). What does mean to say that TCP provides fair sharing of bandwidth at the bottleneck link? Suppose the RTTs of the two connections are very different. Is TCP "fair" in this case? Justify your answer.

Answer: TCP is fair is that in the long term, each receiver will eventually see the same throughput at that link. As noted in the answer above, the TCP window size grows at a rate that is proportional to the RTT. If the RTT's are different, then the two TCP senders will grow their windows at different rates (even if they start with the same window size).

Question 2: Routing Algorithms (25%)

a) (10%) Consider the network shown below. Show the operation of Dijkstra's (Link State) algorithm for computing the least cost path from **F** to all destinations.



Step	N’	D(A), P(A)	D(B), P(B)	D(C), P(C)	D(D), P(D)	D(E), P(E)
0	F	∞	4, F	∞	∞	1, F
1	FE	∞	4, F	∞	5, E	
2	FEB	5, B		7, B	5, E	
3	FEBA			7, B	5, E	
4	FEBAD			6, D		
5	FEBADC					

or

Step	N’	D(A), P(A)	D(B), P(B)	D(C), P(C)	D(D), P(D)	D(E), P(E)
0	F	∞	4, F	∞	∞	1, F
1	FE	∞	4, F	∞	5, E	
2	FEB	5, B		7, B	5, E	
3	FEBD	5, B		6, D		
4	FEBDA			6,D		
5	FEBDAC					

b) (15%) Consider the three-node topology: the link costs are c(x,y)=5, c(y,z)=2, and c(z,y)=40. We focus here only on y’s and z’s distance table entries to destination x. Please explain the details of how the DV algorithm could reach a quiescent (or stable) state when the following cases happen.

a. (5%) When c(x,y) decreases from 5 to 1.

First, Y detects link-cost change, updates its DV, and informs its neighbors. Then, Z receives the update from Y, updates its table, computes new least cost to X, and sends its neighbors its DV. Finally, Y receives Z’s update, and updates its distance table. Y’s least costs do not change, so it does not send a message to Z again.

b. (5%) When c(x,y) increases from 5 to 50.

First, Y detects link-cost change, computes new least cost to X via Z (5 to 9), and sends its neighbors its DV. Then, Z receives the update from Y, updates its table, computes new least cost to X via Y (7 to 11), and sends its neighbors its DV. After several iterations, Y and Z's least cost to X eventually converge to 42 and 40.

c. (5%) How the poisoned reverse solves the above looping problem.

"Poisoned reverse" means that if Z routes through Y to get to X, Z tells Y its (Z's) distance to X is infinite. This solves the looping problem since Y won't route to X via Z (That is, $Y \rightarrow Z \rightarrow Y \rightarrow Z\ldots$ will not occur).

<u>Question 4: Reliable Broadcast Channel (20%)</u>

Consider a scenario in which a host, A, wants to simultaneously send messages to hosts B and C. A is connected to B and C via a broadcast channel --- a packet sent by A is carried by the channel to both B and C Suppose that broadcast channel connecting A, B, and C can independently lose and corrupt messages (and so, for example , a message sent from A might be correctly received by B, but not by C.) Design a stop-and-wait-like error-control protocol for reliably transferring a packet from A to B and C, such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. Give FSM descriptions of A and B. (Hint: The FSM for C should be essentially the same as for B.)

This problem is a variation on the simple stop and wait protocol (rdt3.0). Because the channel may lose messages and because the sender may resend a message that one of the receivers has already received (either because of a premature timeout or because the other receiver has yet to receive the data correctly), sequence numbers are needed. As in rdt3.0, a 0-bit sequence number will suffice here.

The sender and receiver FSM are shown in Figure 3. In this problem, the sender state indicates whether the sender has received an ACK from B (only), from C (only) or from neither C nor B. The receiver state indicates which sequence number the receiver is waiting for.

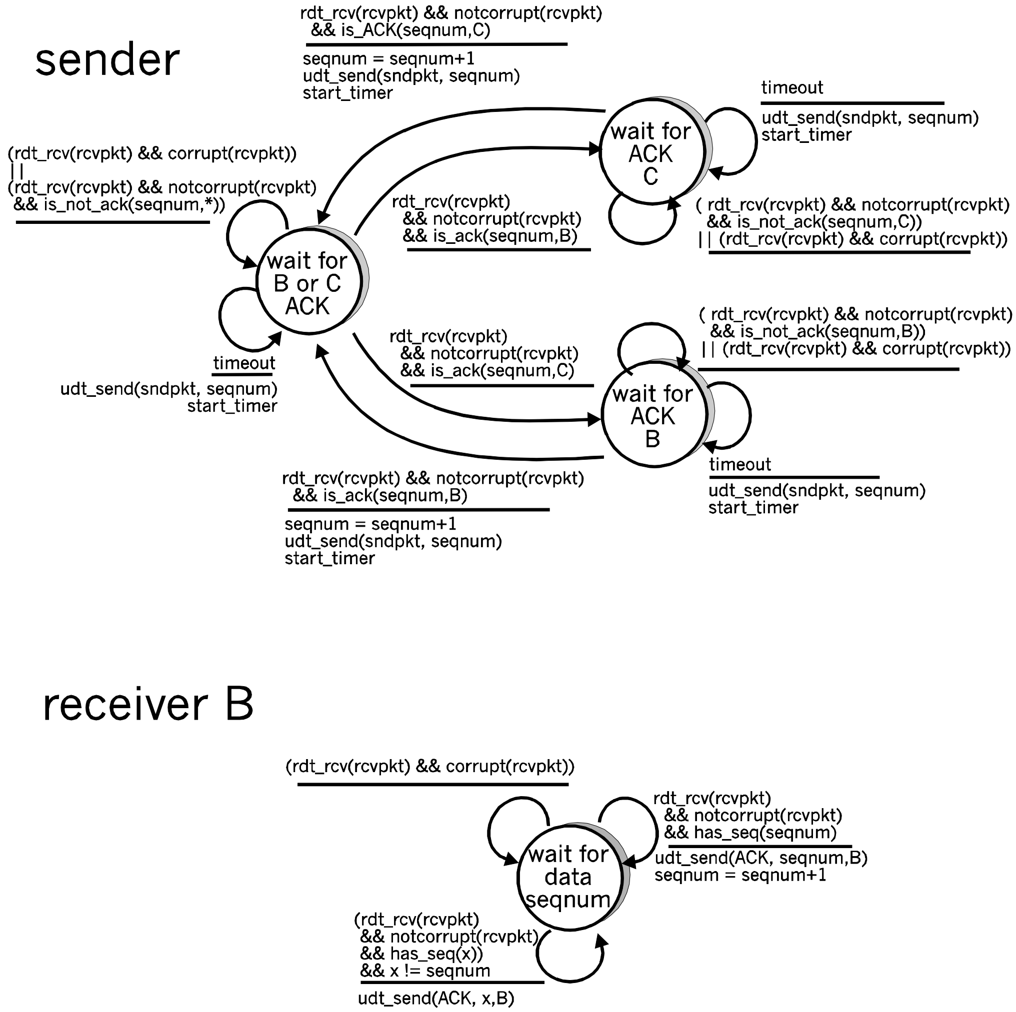


Figure 3. Sender and receiver FSMs.

Question 5: Homework Review (20%)

These problems are not the same as homework, which means that we have modified some description. Please be careful when answering.

(1) (10%) Consider a subnet with prefix 128.119.40.128/28.

a. (5%) Give the "range" of IP addresses (of form xxx.xxx.xxx.xxx) that can be assigned to this network.

   Answer

   IP address in range 128.119.40.128 to 128.119.40.143.

b. (5%) Suppose an ISP owns the block of addresses of the form 128.119.40.96/26. Suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

Answer:

128.119.40.96/28,

128.119.40.104/28,

128.119.40.112/28,

128.119.40.120/28.

Or

128.119.40.64/28,

128.119.40.80/28,

128.119.40.96/28,

128.119.40.112/28.

(2) Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 80; the second has sequence number 120.

a. (5%) How much data is in the first segment?
   Answer: 40
b. (5%) Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgement that Host B sends to Host A, what will be the acknowledgment number?
   Answer: 80