EE 122 Final Exam - Solution
Date: December 14, 2002

Name:
SID:
ee122 login:
Day/Time of section you attend:

Problem	Points
1	/10
2	/10
3	/10
4	/10
5	/10
6	/10
7	/10
8	/10
9	/10
10	/10
Total	/100

1. Question 1 (10 pt)

(a) (5 pt) Explain briefly how the Reservation Protocol (RSVP) works. In particular, what are the main messages and what do end-hosts and routers do upon receiving these messages?

(b) (5 pt) How does RSVP handle IP route changes? Give a simple example to illustrate this point.

Solution:

a) RSVP is a signaling protocol that, in the context of Intserv, is used for admission control and resource reservation. There are two main control messages in RSVP:
   - Path message – this message is sent by the sender to the receiver and is used to (1) carry the sender's traffic specifications, and (2) set up the path. Upon receiving this message each node stores the previous node along the path.
   - Resv message – this message is sent by the receiver to sender and is used to perform admission control and resource reservation. Upon receiving such a message a node checks whether it has enough resources to satisfy the receiver requirements. If yes, it reserves resources; if not, it marks the message to indicate that the reservation is denied. In both cases the node forwards the message to the previous node (set up by the Path message) towards the source.

b) RSVP is a soft-state protocol, i.e., routers maintain state only as long as it is periodically refreshed; when the state timeouts it is simply removed. To refresh state on routers along the path, the sender periodically sends Path messages to the receiver and the receiver periodically sends Resv messages to the sender. RSVP handles IP route changes as follows. When the route changes, the next Path message is forwarded along the new route and a new reverse path is set up. Then, when the next Resv message is sent by the receiver, the message travels the new route and reserves resources on that route. Eventually, the resources on the old route are released when they timeout. For an example, see the example in page 10 of the lecture "Integrated Services".

2. Question 2 (10 pt)

Answer each of the points below as briefly as possible:

(a) (5 pt) Give two advantages of RED over Drop Tail. Explain.
(b) (5 pt) Give one advantage of RED over Round-Robin and one advantage of Round-Robin over RED. Explain.

Solution:

(a) RED avoids (1) TCP flow synchronization and (2) large bursts of losses. The main reason behind these advantages is the fact that RED starts to drop packets randomly before the buffer is full. This allows the router to spread losses over a larger time interval, and signal the congestion to end-hosts earlier.

(b) RED can be implemented more efficiently since, unlike Round-Robin, it does not require to maintain per flow buffers and perform per-flow scheduling; RED maintains only one queue and performs FIFO scheduling.

Round-Robin can provide per-flow isolation. While with RED, a non-adaptive flow such as an UDP flow can hurt a TCP flow if it sends traffic at more than its fair share, with Round-Robin that flow can hurt only itself.

3. Question 3 (10 pt)

(a) (5 pt) Describe the Assured service model.
(b) (5 pt) Describe how RIO (Red with In and Out) buffer management scheme works, and how it can be used to implement the Assured service model.

Solution:

(a) The assured service provides a traffic profile (usually characterized by token-bucket) to each customer. The customer is promised that as long as it sends assured traffic without exceeding its profile, it will experience a lower packet loss than using the best-effort service. The traffic profile is defined independent of the destinations.

(b) RIO is an extended version of RED that uses two thresholds. When the queue size exceeds the first threshold, RIO starts to drop the best-effort packets. RIO starts to drop assured packets only when the queue size of the assured traffic exceeds the second threshold. Since the second threshold is always larger than the first one, RIO will start to drop assured packets only as a last resort, i.e., if dropping all the best-effort packets will not eliminate the congestion.

4. Question 4 (10 pt)

(a) (5 pt) Explain how the fair max-min share is computed.
(b) (5 pt) Consider 10 flows with the arrival rates of 1, 2, …, 10 Mbps that traverse a link of 45Mbps. Compute the fair share on this link. What is the fair share if the link capacity is 60 Mbps?

Solution:

(a) Let C be the capacity of the congested link, S be set of all flows, n be the number of flows in S, and r_i be the arrival rate of flow i. Then we can use the following iterative algorithm to compute the max-min fair share:
   (1) Compute f=C/n
   (2) Let S(f) be the sub-set of flows from S consisting of all flows with arrival rates that are less or equal to f. If S(f) not empty:

$$C = C - \sum_{i \in S(f)} r_i;$$
$$n = n - size(S(f));$$
$$S = S - S(f);$$

goto step 1

Otherwise, stop, and f is the max-min fair share; terminate

(b) For C=45Mbps we have:

   (1) Iteration 1: f = 45Mbps/10 = 4.5Mbps; S(f) = {1, 2, 3, 4}. Thus, C = 45Mbps – 10Mbps = 35Mbps, n = 6, and S = {5, 6, 7, 8, 9, 10}
   (2) Iteration 2: f = 35Mbps/6 = 5.85 Mbps; S(f) consists of flow 5, C =35Mbps –5Mbps = 30Mbps, n = 5, and S = {6, 7, 8, 9, 10}
   (3) Iteration 3: f = 30Mbps/5 = 6Mbps; S(f) is empty, thus max-min fair share is **6**Mbps

For C=60Mbps note that the sum of all arrival rates is 1+2+…+10 = 55Mbps, which means that all flows will get how much they requested. In this case the fair rate is 10, the maximum arrival rate of all flows. (You can reach the same result using the iterative algorithm above).

5. Question 5 (10 pt)

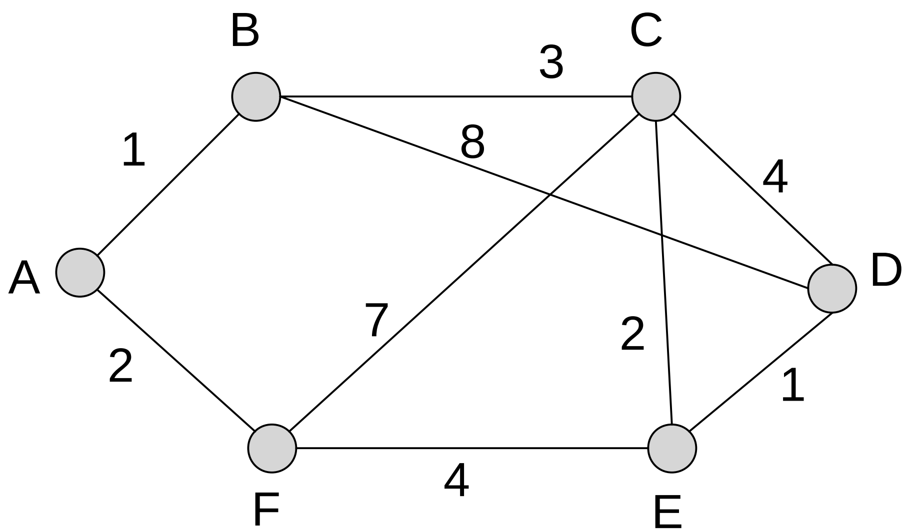
Derive the relation between the minimum size of an Ethernet packet l and maximum distance d between two end-hosts on an Ethernet cable. Denote the propagation speed of the signal in the communication media (i.e., cable) by c and the capacity of the cable by B.

Solution:

See slide 18 from the lecture "Ethernet and 802.11". The final result is: $d = (l*c)/(2*B)$

6. Question 6 (10 pt)

Consider the network below. Show the iterations followed by node A to compute the shortest paths from itself to all the other nodes using the Dijkstra algorithm. Assume A has already collected all link state information in the network.



Solution:

Step	S	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	1,A	∞	∞	∞	2,A
1	AB		4,B	9,B	∞	2,A
2	ABF		4,B	9,B	6,F	
3	ABFC			8,C	6,F	
4	ABFCE			7,E		
5	ABFCED					

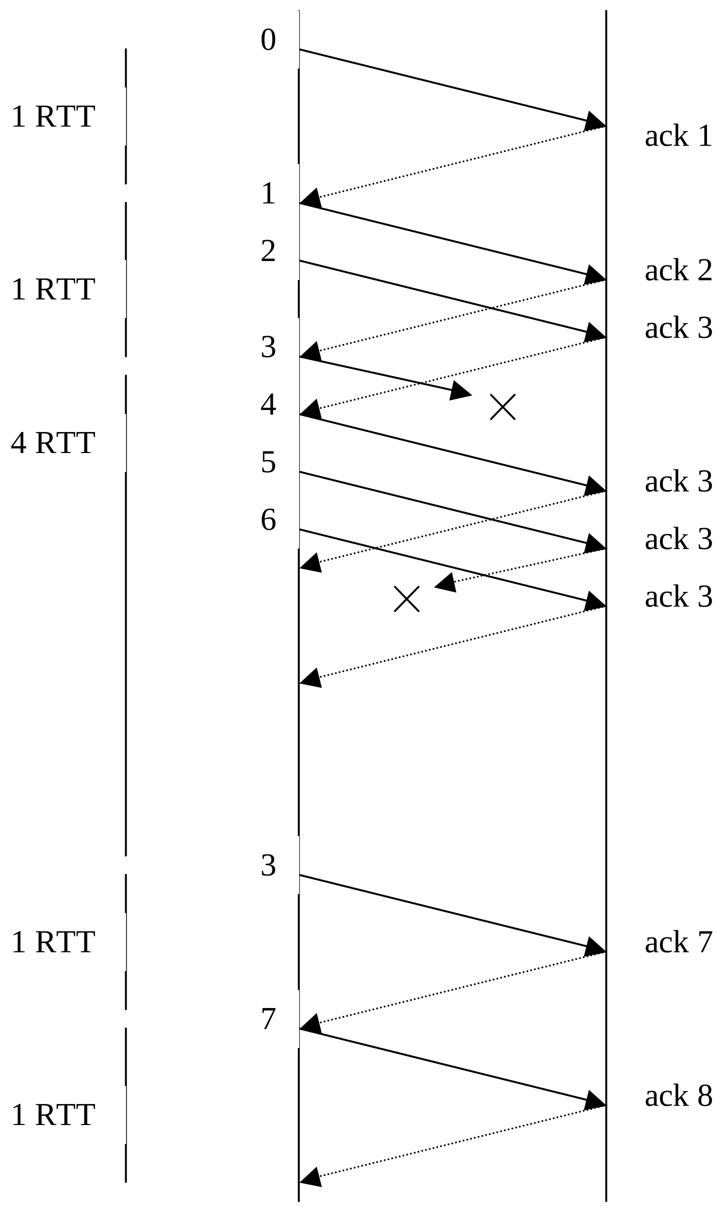
7. Question 7 (10 pt)

Consider a source A that uses TCP to transfer a file of 8KB to destination B. Each packet has a payload of 1KB. Assume that only two packets are lost during the transfer: the 4th data packet and the acknowledgment of the 6th data packet. Draw the time diagram of the entire file transfer. Assume that retransmission timeout (RTO) is 4*RTT. How many RTTs does it take to transfer the entire file?

Note: The transfer time represents the interval between the time the acknowledgement of the last packet was received by A and the time the first packet is transmitted by A. Ignore the connection setup, processing and transmission times. Fast retransmission and fast recovery are triggered after receiving 3 duplicate acknowledgements.

Solution:

The answer is 8 RTTs:

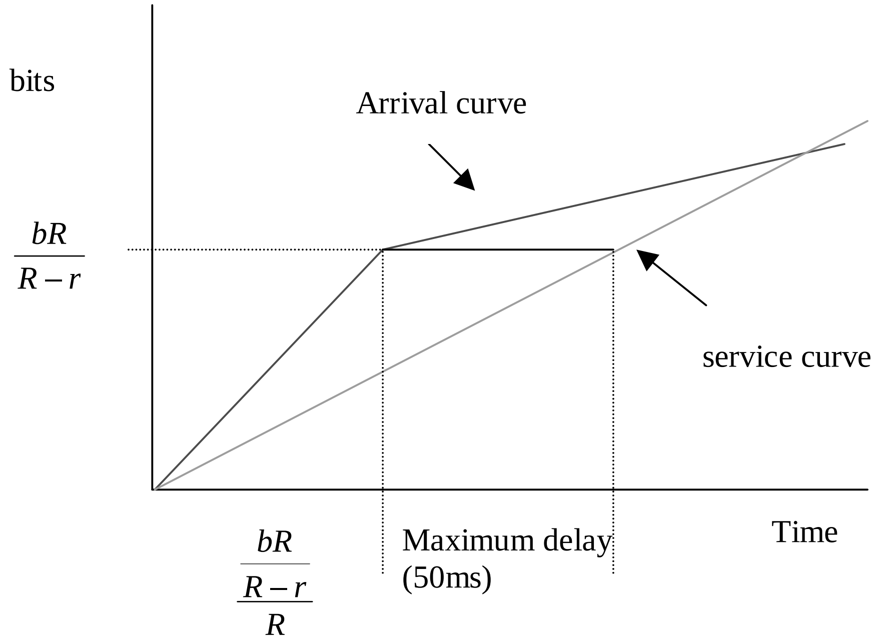


Some people assumed that duplicate acknowledgements open the congestion window. TCP, as described in class, does not do this, but it is a reasonable thing for it to do, so we did not deduct points for this.

+3 points for getting the first 2RTT from 3 packets in slow start
+2 points for having the sender recognize that packet 3 and 5 are unacked
+2 points for only retransmitting packet 3
+3 for having correct RTO (in solution given above) or Fast Retransmit (in alternate solution)

8. Question 8 (10 pt)

Consider the arrival traffic characterized by a token bucket with parameters r (average rate) = 1 Mbps, R (maximum rate) = 2 Mbps, and b (token depth) = 100Kb. What is the minimum rate r_a that needs to be allocated by a router in order to guarantee a delay no larger than 50 ms?



Solution:

The minimum rate of the service curve is the slope of the service curve. From the diagram, that slope is

$$\frac{\frac{bR}{R-r}}{\frac{\frac{bR}{R-r}}{R}+50ms}$$

$$=\frac{\frac{100Kb*2Mb/s}{2Mb/s-1Mb/s}}{\frac{\frac{100Kb*2Mb/s}{2Mb/s-1Mb/s}}{2Mb/s}+50ms}$$

$$=\frac{200Kb}{\frac{200Kb}{2Mb/s}+50ms}$$

$$= \frac{200Kb}{150ms}$$
=1.3Mb/s

+2 for having the correct y-intercept $\frac{bR}{R-r}$

+2 for identifying the 50ms separation between the arrival curve and the service curve at the correct point

+2 for having the correct delay until the bucker is emptied $\frac{\frac{bR}{R-r}}{R}$

+4 for assembling it into the correct final equation

9. Question 9 (10 pt)

Consider a Chord ring consisting of 10 nodes with identifiers 1, 11, 15, 23, 31, 40, 45, 51, 60, and 63, respectively. Assume the Chord ring spans the identifier space $0..2^6$-1.

(a) (3 pt) Give the routing (finger) tables of nodes with identifiers 1, 40, and 45.
(b) (4 pt) Assume node with identifier 1 initiates a query for the item with *id* = 55. Give the routing steps to resolve this query in the case of iterative routing.
(c) (3 pt) Give two advantages of the iterative routing over recursive routing. Briefly explain.

Solution:

a)

Node 1's finger table

index	Expected id	Actual id
0	1+1=2	11
1	1+2=3	11
2	1+4=5	11
3	1+8=9	11
4	1+16=17	23
5	1+32=33	40

Node 40's finger table

index	Expected id	Actual id
0	40+1=41	45
1	40+2=42	45
2	40+4=44	45
3	40+8=48	51
4	40+16=56	60
5	40+32=72 mod 64 = 8	11

Node 45's finger table

index	Expected id	Actual id
0	45+1=46	51
1	45+2=47	51
2	45+4=49	51
3	45+8=53	60
4	45+16=61	63
5	45+32=77 mod 64 = 13	15

+1 for each correct table

b)

1->40, 40->60. 60 has responsibility for id 55.

+4 for the right answer (all or nothing)

c)
Iterative querying is more fault tolerant because intermediate nodes do not have to be up for the entire query; they only have to be up long enough to receive and answer a query. Iterative querying is also more resistant to denial-of-service attacks because they do not have to hold state during queries

+2 for each right answer (max 3)

10. Question 10 (10 pt)

Consider a path consisting of three links A-B-C-D where the capacities of links (A, B) and (C, D) are C_1 and the capacity of link (B, C) is C_2. Assume node A sends two packets at times t_1 and t_2, respectively, to node D. Both packets have the same length l.

(a) (5 pt) What is the inter-arrival time d between the two packets at node D?
(b) (5 pt) Assume $C_2 < C_1$. Give a simple method to measure the capacity C_2 based on the inter-arrival time d and the packets' length l.

Notes: Assume there is no cross-traffic; the two packets are the only ones in the network. The departure time of a packet is the time when the last bit of the packet is transmitted. The arrival time of a packet is the time when the last bit of the packet is received. With a store-and-forward router a packet is forwarded only after the last bit of the packet was received. The inter-arrival time d is equal to the arrival time of the second packet minus the arrival time of the first packet at node D.

Solution:

We'll calculate the arrival time of the first packet and then the arrival time of the second packet.

arrival time of packet 1 = initial transmission time +fixed delay on AB + transmission delay on AB + fixed delay on BC + transmission delay on BC + fixed delay on CD + transmission delay on CD

$$= t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + d + \frac{l}{C_1}$$

The arrival time of packet two depends on whether it queues behind packet 1. Assuming it never queues behind packet one, it arrives at D at

$t_2 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + d + \frac{l}{C_1}$.Therefore, the interarrival time is

$$(t_2 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + d + \frac{l}{C_1}) - (t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + d + \frac{l}{C_1})$$

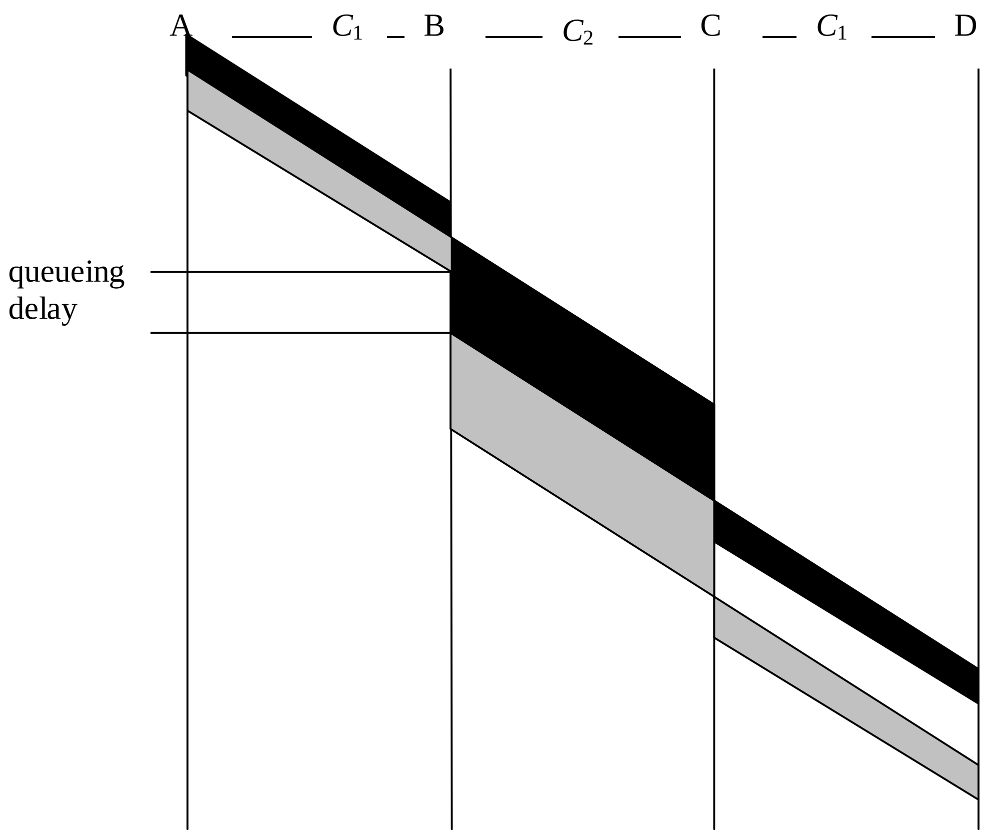
$$= t_2 - t_1$$

If there is queueing, one possibility is that the second packet queues behind the first at B:

$$t_2 + d + \frac{l}{C_1} < t_1 + d + \frac{l}{C_1} + \frac{l}{C_2}$$

Simplifying, we have $t_2 - t_1 < \frac{l}{C_2}$.

This is shown in the following figure:



As a result, packet 2 arrives at C at $t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + \frac{l}{C_2}$ and subsequently at D at $t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + \frac{l}{C_2} + d + \frac{l}{C_1}$. The interarrival time is then

$$t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + \frac{l}{C_2} + d + \frac{l}{C_1} - \left(t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + d + \frac{l}{C_1} \right)$$

$$= \frac{l}{C_2}$$

Another possibility is that packet 2 queues at C. However, for this to happen, $t_2 + d + \frac{l}{C_1} + d + \frac{l}{C_2} < t_1 + d + \frac{l}{C_1} + d + \frac{l}{C_2} + \frac{l}{C_1}$, which simplifies to $t_2 - t_1 < \frac{l}{C_1}$. This can't happen because it contradicts the definition of packet departure times which states that $t_2 - t_1 \geq \frac{l}{C_1}$.

+2 for identifying queueing at the BC link as the distinguishing condition

b) From part a), we know that if $t_2 - t_1 < \frac{l}{C_2}$, then the packet interarrival time is $\frac{l}{C_2}$. To measure C_2's bandwidth, we ensure $t_2 - t_1 < \frac{l}{C_2}$ and take $C_2 = \frac{l}{d}$.