University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 122 I. Stoica
Fall 2009

FINAL EXAMINATION
December 17, 2009

INSTRUCTIONS—READ THEM NOW! This examination is OPEN BOOK. There is no need for complex calculations, and so you may not require a calculator, Palm Pilot, laptop computer, or other calculation aid. Please put them away. You MAY use one 8.5" by 11" double-sided crib sheet, as densely packed with notes, formulas, and diagrams as you wish. All work should be done on the attached pages.

In general, if something is unclear, write down your assumptions as part of your answer. If your assumptions are reasonable, we will endeavor to grade the question based on them. If necessary, of course, you may raise your hand, and a TA or the instructor will come to you. Please try not to disturb the students taking the examination around you.

________________________________ SID: ________________________
(Signature)

________________________________ Discussion Section (Day/Time): ________
(Name—Please Print!)

QUESTION	POINTS ASSIGNED	POINTS OBTAINED
1	12	
2	18	
3	10	
4	10	
5	20	
6	20	
7	10	
TOTAL	100	

1. Miscellaneous (12 points)

a) *End-to-end Argument* (3 points). Does IP multicast violate the e2e argument? Argue, why, or why not.

Each of the following answers received full credit:

1. Based on the radical interpretation of the e2e argument, IP multicast is a violation because it does not keep the network layer minimal.
2. Based on the moderate interpretation, IP multicast does not violate the e2e argument because implementing multicast at the network layer provides a high performance gain as opposed to implementing multicast at the application layer. At the same time, implementing IP multicast impacts other services minimally.

b) *Little Theorem* (3 points). Assume a queue at which packets arrive at an average rate of 10pkts/sec, and assume that each packet spends 2sec in the queue on average. What is the average number of packets in the queue?

10 pkts/sec * 2 sec = 20 pkts

c) *Network Address Translation* (3 points). What is the maximum number of simultaneous connections that machines behind a NAT can open to the public Internet? Justify your answer.

2^{16} because that is the number of different NAT ports that can be used.

d) *TCP* (3 points). Assume the advertised window of a TCP receiver does not exceed 3 segments. Give a "scenario" in which TCP Reno achieves better performance than TCP Tahoe, if any. Justify your answer. (A "scenario" can be specified by the number of segments sent by the sender, and by indicating which segments are lost, if any.)

There is no scenario because TCP Reno can never obtain the three duplicate ACKs necessary to do fast retransmissions and fast recovery when the receiver's advertized window does not exceed 3 segments.

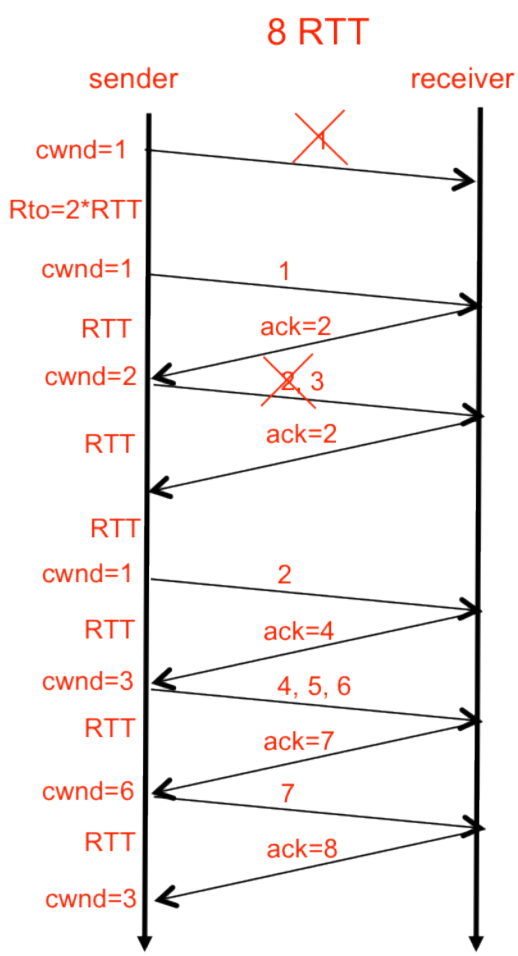
2. TCP (18 points)
Consider a TCP flow sending 7 segments. Due to congestion, the TCP flow experiences exactly 2 losses. Only segments can be lost, acknowledgements are not lost.

The time it takes the sender to send all segments represents the difference between (i) the time the sender receives the ack for the last segment, and (ii) the time the sender sends the first segment. The transmission time of a segment is negligible, and the round-trip-time is RTT. The retransmission timeout (RTO) is twice the RTT, i.e., RTO = 2*RTT.

Assume we use TCP Tahoe.

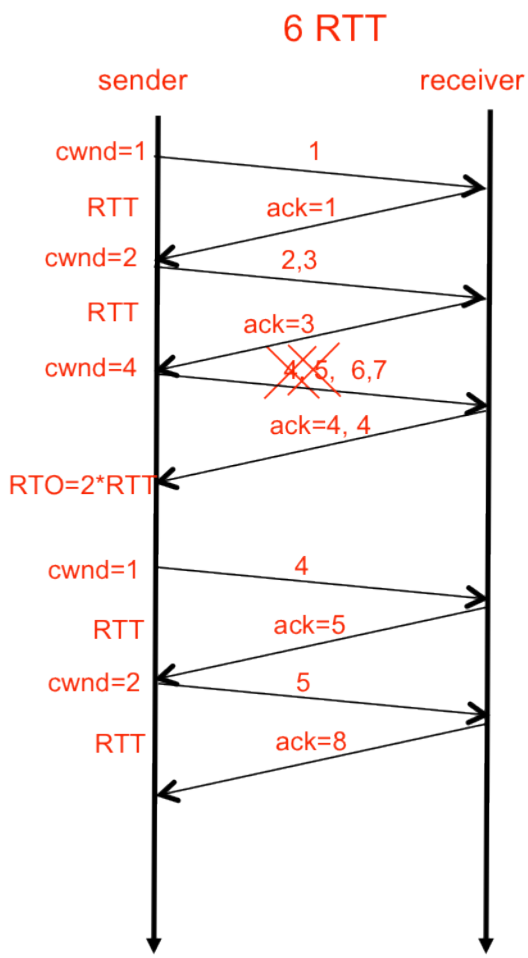
(a) (6 points) What is the maximum time it could take the sender to send all segments as a function of RTT? Use the following time diagram to illustrate a scenario that achieves this maximum. Show the cwnd used by the sender when sending each segment.

Here is one scenario; other scenarios arriving at the same result received full credit.



(b) (6 points) What is the minimum time it takes the sender to send all segments? Use the following time diagram to illustrate a scenario that achieves this minimum. Show the cwnd used by the sender when sending each segment.

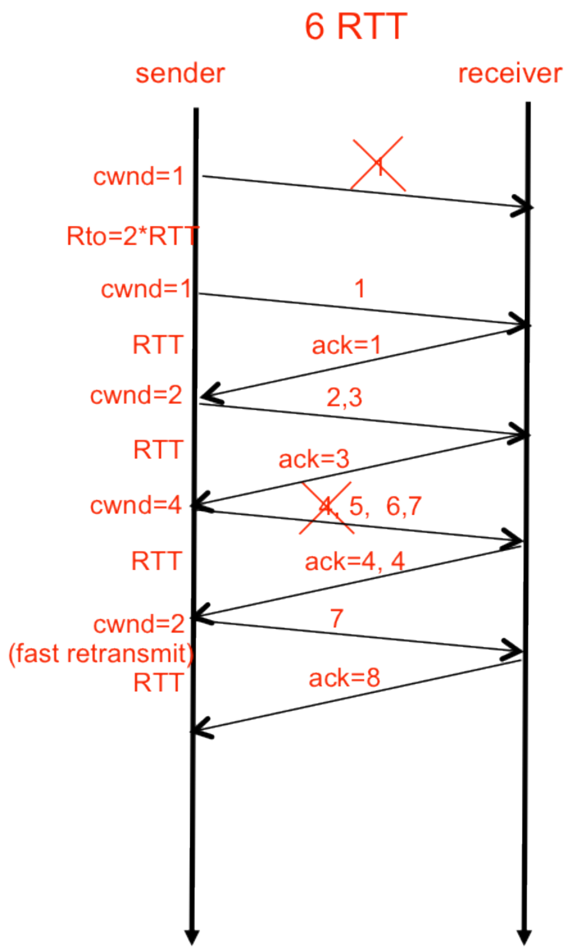
Here is one scenario; other scenarios arriving at the same result received full credit.



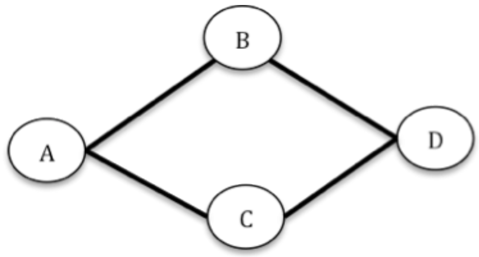
(c) (6 points) Repeat question (b) for TCP Reno.

Note: In the case of TCP Reno, after a fast retransmit, the sender waits for the ack. If the sender doesn't receive ack, it resends the segment when RTO triggers, and then enters the slow start phase.

Here is one scenario; other scenarios arriving at the same result received full credit.
(Note: Quite a few people gave "scenarios" in which the time to transmit all segments was less than 6 RTT. Almost in all cases the common mistake was to assume that the sender performs fast-recovery and fast-retransmit after receiving only 2 dupacks.)



3. Load-sensitive routing (10 points)
Consider the following network, where A sends traffic to D. Assume the routing protocol computes the shortest path, where the cost of a link represents its traffic load (i.e., the amount of traffic on the link). The traffic from A to D represents the only traffic in the network.



(a) (5 points) Which route is the traffic from A to D going to use? Is the route stable? Explain why, or why not.

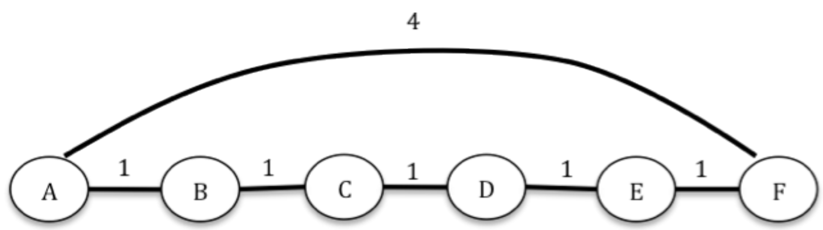
The route will oscillate between A -> B -> D and A -> C -> D. It will not be stable because the route not being used will always have a smaller cost than the one being used. The route being used will always have its cost increase due to the traffic using it.

(b) (5 points) Assume the traffic from A to D is TCP and assume that the round-trip time along A-B-D is much higher than the round-rip-time along A-C-D. What is the impact of the routing algorithm on the TCP performance? Justify your answer.

TCP is unable to get a good estimate of the RTO, which leads to spurious retransmissions.

4. Routing (10 points)

a) (5 points) For the following topology, fill in the link-state table provided below using Dijkstra algorithm. The ties are broken based on the node ID, where the node with the lower ID wins.



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	Inf	Inf	Inf	4,A
1	A B		2, B			
2	A B C			3, C		
3	A B C D				4, D	
4	A B C D E					
5	A B C D E F					

b) (5 points) Repeat question (a) after add the following two links to the above topology: A—D with a cost of 2, and C—F with a cost of 2.

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	Inf	2, A	Inf	4, A
1	A B		2, B			
2	A B C					
3	A B C D				3, D	
4	A B C D E					
5	A B C D E F					

5. Token bucket (20 points)

Consider a source sending traffic at 1.6Mbps for the first second of each period, and at 0.4Mbps after that. For simplicity, neglect the packet size.

a) (5 points) Calculate the token bucket parameters *R*, *r* and *b* for the flow.

R = 1.6Mbps, r= 0.4Mbps, b = (R – r)*1s = 1.2Kb = 150KB,

b) (5 points) Assume a router X that can allocate at most 100KB of buffer to this flow. What is the minimum rate r_a that X needs to allocate to the flow so that the flow experiences no losses?

(R*1sec – 100KB*8)/1sec = 0.8Mbps

c) (5 points) Repeat question (b) assuming that router X can allocate 200KB to the flow.

(R*1sec – 200KB*8)/1sec = 0Mbps. However, r_a cannot be smaller than r, and thus r_a = r = 0.4Mbps

d) (5 points) Assume the maximum delay that the flow can experience at router X is 200ms. What is the minimum rate r_a that X needs to allocate to the flow to meet this delay constraint?

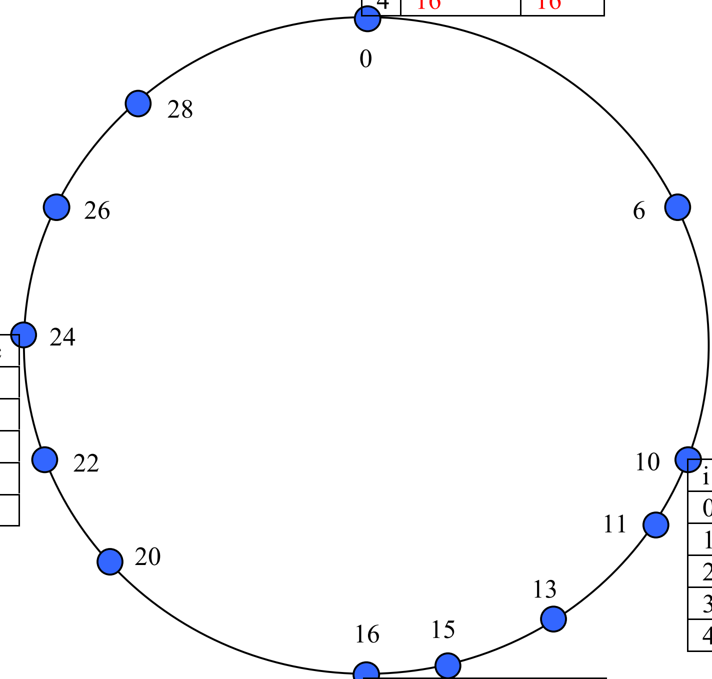
(R*1sec)/(1sec + 0.2sec) = 1.33Mbps

Question 6. *Multicast on Chord* (*20 Points*)

The following diagram shows a Chord P2P network with name space [0 … (2^5 -1)].

(a) (5 points) Fill in finger tables of nodes 0, 10, 16 and 24 in the diagram.

i	id + 2^i	Succ
0	1	6
1	2	6
2	4	6
3	8	10
4	16	16



i	id + 2^i	Succ
0	25	26
1	26	26
2	28	28
3	0	0
4	8	10

i	id + 2^i	Succ
0	11	11
1	12	13
2	14	15
3	18	20
4	26	26

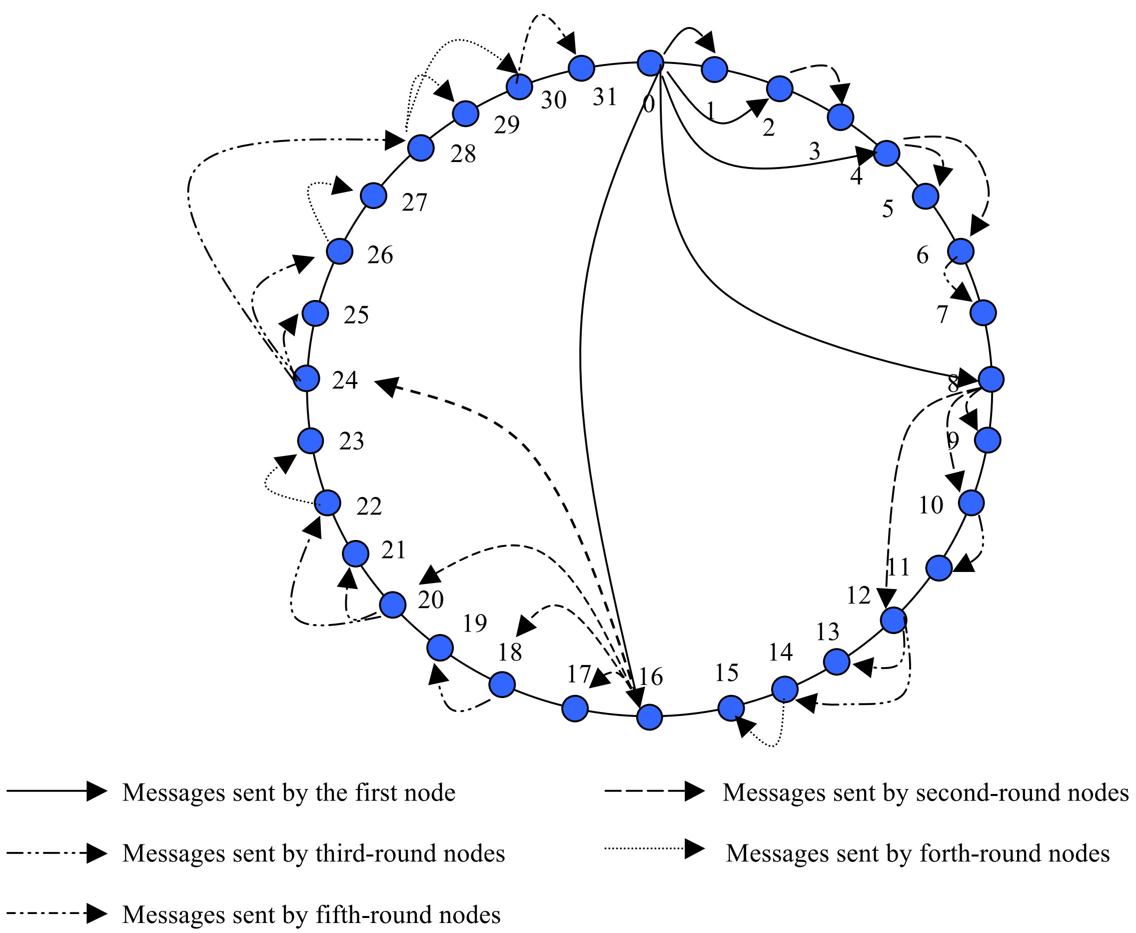
i	id + 2^i	Succ
0	17	20
1	18	20
2	20	20
3	24	24
4	0	0

(b) (4 points) Design a multicast protocol for the Chord Ring. Consider a multicast group with node 0 as the source node. Other nodes in the ring want to receive data from node 0. A naive protocol only uses the first successor in the finger table to forward messages from one to another until messages reach all nodes, i.e., data are forward 0 -> 6 -> 10 -> … -> 28. List two major disadvantages of this protocol.

(i) slow, significant latency for nodes with high id

(ii) single point of failure. If one node fails, all remaining nodes won't get the packet

(c) (5 points) Design a more efficient multicast protocol on a complete Chord Ring. The following shows a multicast traffic pattern of a protocol on the first half of the ring. Messages are forwarded recursively via finger tables within certain range limitation. Draw the traffic pattern on the second half of the ring using different arrow patterns for different round of messages.



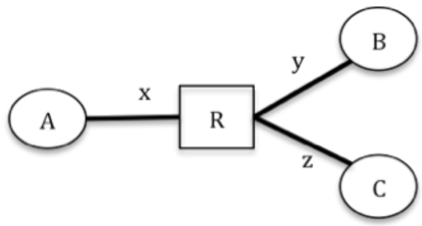
(d) (6 points) Provide pseudocode every node uses to implement a protocol resulting in the traffic pattern in (c).

Notation: A call ***n.forward***() is to execute function ***forward***() at node ***n***. This function can remotely invoke ***forward***() at other nodes in the network. For example, if you call ***m.forward***() at node ***n***, this means ***n*** sends a message to node ***m*** and asks node ***m*** to execute ***forward***(). You don't have to follow this notation and you can come up with your own notation.

```
// The initial setup for node 0
current_range.lower_bound = 0;
current_range.upper_bound = 31;
n = 0;     // 0 is the root node;
n.forward(msg, current_range);       // ask node 0 to do multicast forwarding

// Every node implements forward() to multicast messages.
n.forward (msg, current_range)
{
For ( i=0; current_range.lower_bound+2^i <= current_range.upper_bound; i++ ) {
new_range.lower_bound = current_range.lower_bound+2^i;
new_range.upper_bound = current_range.lower_bound+2^(i+1) – 1;
m = current_range.lower_bound+2^i;
m.forward (msg, new_range);
}
```

7. Measuring link capacities (10 points)
Consider the network below where three hosts A, B, and C, respectively, are connected to router R. The capacity of the links are x Kbps, y Kbps, and z Kbps, respectively.

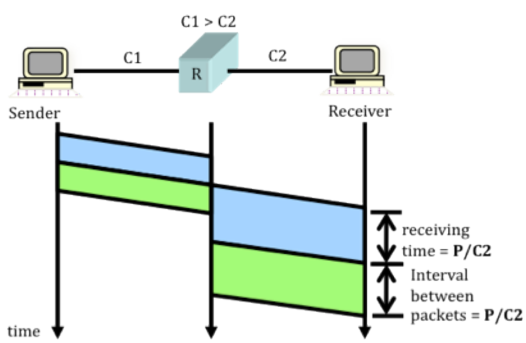


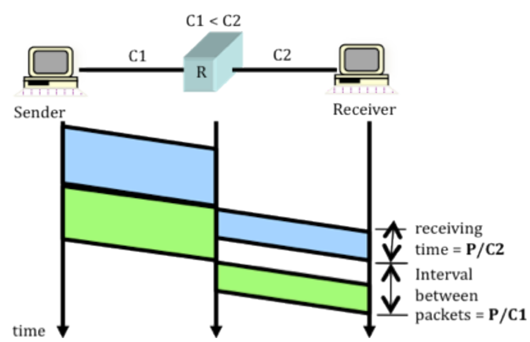
Assume the users of hosts A, B, and C, respectively, want to measure the capacities of the three links. To achieve this, they send the following messages:

- From A to B: Send two back-to-back 1Kbit packets; the interval between receiving the first and the second packet at B is 100ms; the time it takes B to receive each packet (i.e., the interval between receiving the first bit of the packet and the last bit of the packet) is 100ms.
- From B to C: Send two back-to-back 1Kbit packets; the interval between receiving the first and the second packet at C is 200ms; the time it takes B to receive each packet is 200ms.
- From C to A: Send two back-to-back 1Kbit packets; the interval between receiving the first and the second packet at A is 200ms; the time it takes B to receive each packet is 50ms.

Assume router R is store-and-forward, and the receiving time of a packet represents the time the last bit of the packet is received. Assume no other traffic in the network and a 0 packet processing delay.

a) (5 points) Based on the above measurements, compute capacities x, y, and z, respectively.





The above figure illustrates the impact of the relationship between link capacities on (i) the inter-arrival between receiving the two packets, and (ii) the time it takes to receiving a packet. The packet length is P bits.

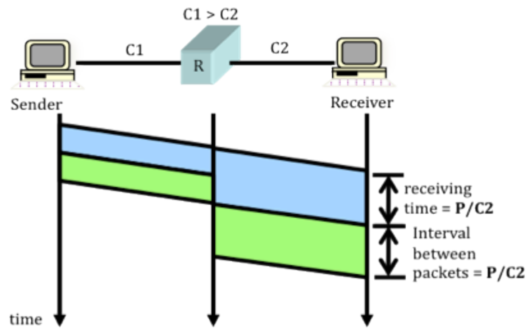
Along path A—R—B, the bottleneck link is 1Kbit/0.1s = 10Kbps, and the capacity of the last link (i.e., R—B) is 1Kbit/0.1s = 10Kbps. Thus R—B is the bottleneck link and has the capacity of 10Kbps, while A—R has a capacity which is at least 10Kbps.

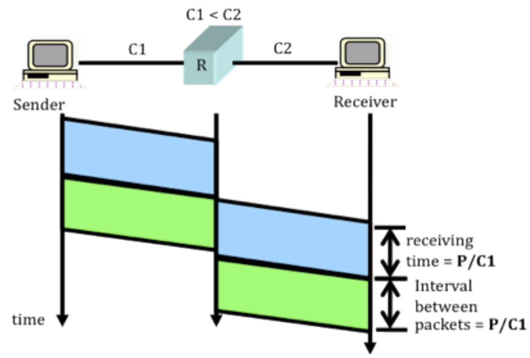
Along path B—R—C, the bottleneck link is 1Kbit/0.2s = 5Kbps, and the capacity of the last link (i.e., R—C) is 1Kbit/0.2s = 5Kbps. Thus R—C is the bottleneck link and has the capacity of 5Kbps.

Finally, along path C—R—A, the bottleneck link is 1Kbit/0.2s = 5Kbps, and the capacity of the last link (i.e., R—A) is 1Kbit/0.05s = 20Kbps.

Thus, A—R: 20Kbps, A—B: 10Kbps, A—C: 5Kbps

b) (5 points) Answer (a) assuming R is a cut-through router. Neglect the header size. (Note in the class: For this point you ignore the times to receive a packet.)





The above figure illustrates the impact of the relationship between link capacities in the case of a cut-through router on (i) the inter-arrival between receiving the two packets, and (ii) the time it takes to receiving a packet. The packet length is P bits.

All the links along a path must transmit at the same rate, and thus transmission rate along a path is dictated by a bottleneck link. So,

min(X, Y) = 1Kbits / 100msec = 10Kbps
min(Y, Z) = 1Kbits / 200msec = 5Kbps
min(Z, X) = 1Kbits / 200msec = 5Kbps

Thus, Z = 5Kbps.
Note we cannot know exact values of X and Y though we know they must meet min(X, Y) = 10Kbps