CS 268 Final Exam
Date: May 8, 2003

Name:
SID:

Problem	Points
1	
2	
3	
4	
5	
Total	

Remember to be concise.

1) Congestion Control (20 pts)

(a) Explain how TCP congestion control may interact with load balancing along two alternate routes (assume the load balancing happens on a per packet basis). What can go wrong? (10 pts)

(b) Explain how TCP congestion control may interact with link layer retransmissions. What can go wrong? (10 pts)

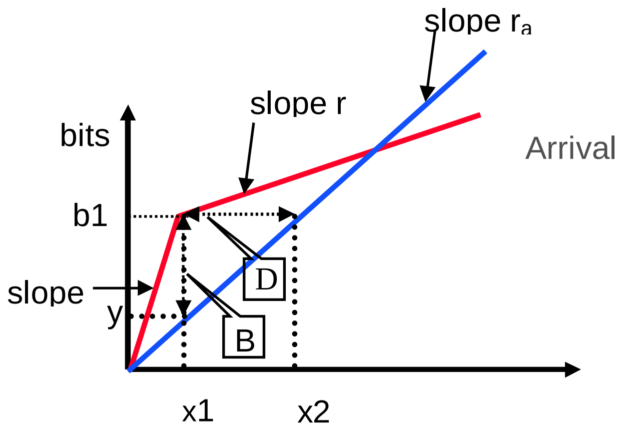
(a) If the packets of the same flow follow different paths with different delays this can cause spurious retransmissions. There are two ways this problem can arise. First, packets can arrive at the receiver out of order which will trigger dup-acks. Second, the RTT estimation algorithm can get confused, which can lead to triggering timeouts too early.

(b) Assume packet with sequence s is lost at time, and then a duplicate of packet s is generated by the link layer retransmission protocol. Assume the duplicate of s arrives at the receiver after packet with sequence number s+1. Then, the receiver will generate an ack asking for s. Thus, the receiver ends up getting two duplicates of s (instead of one), one sent by the TCP sender and one by the link layer retransmission protocol. The net effect is an increase of the congestion in the network.

2) Scheduling (20 pts)

(a) Consider a flow whose arrival rate is characterized by the token-bucket profile with parameters (b, R, r), where b is bucket depth, R is the peek rate, and r is the long term average rate. Assume that the flow is allocated a rate r_a ($r < r_a < R$) at a router. What is the maximum queueing delay experienced by the flow and what is the maximum buffer capacity required by the flow at that router? (10 pts)

(b) Consider the same flow that traverses two routers, and that the flow is allocated r1 at the first router and r2 at the second router.What is the total queueing delay experienced by the flow at the two routers? (10 pts)

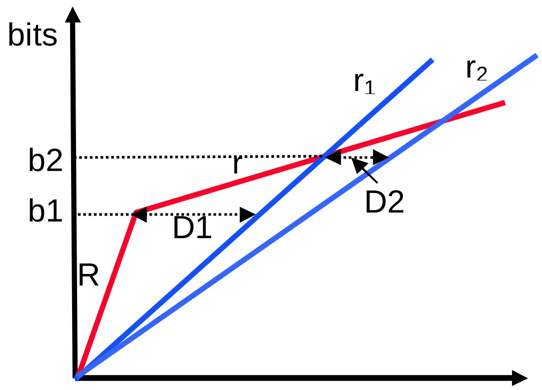


(a) Let b1 = b*R/(R-r) as shown in the figure. (**Note:** There were no points deducted if you assumed b1 = b.)
Using simple algebra yields D = b1*(R-r_a)/(R* r_a), and B = D* r_a = b1*(R- r_a)/R

(c) The question was a bit confusing because it refers to a flow rather than the packets of a flow. Thus, both the following answers were considered correct.

Answer 1. The maximum packet delay is determined by the minimum between r1 and r2. In particular D = b1*(R-min(r1,r2))/(R*min(r1,r2)).

Answer 2. If r2 < r1 then packets experience queueing delay only at the first router and the results are the same as in (a) after replacing r_a with r2.



If r2 > r1 then the total queueing delay is D1 + D2 and the (see figure above). In particular, we have D1 + D2 = b1*(R-r1)/(R* r1) + b2*(R-r2)/(R*r2), where b2 = b1*r1*(R-r)/(R*(r1-r)).

3) Router Architecture (20 pts)

(a) Name one advantage and one disadvantage for each of the following router architectures (i) input queueing, (ii) output queueing, and (iii) input-output queueing. (10 pts)

(b) Describe the head of line blocking problem, and give a solution to address this problem. (10 pts)

(a.i) Advantage: simple to implement; the speedup for output ports 1. Disadvantage: low utilization if output contention.

(a.ii) Advantage: easy to analyze; high throughput. Disadvantage: speedup for output ports as high as N (where N is the number of inputs).

(a.iii) Advantage: can emulate work-conserving disciplines in an output-queueing router with a speedup of 2. Disadvantage: complex to implement.

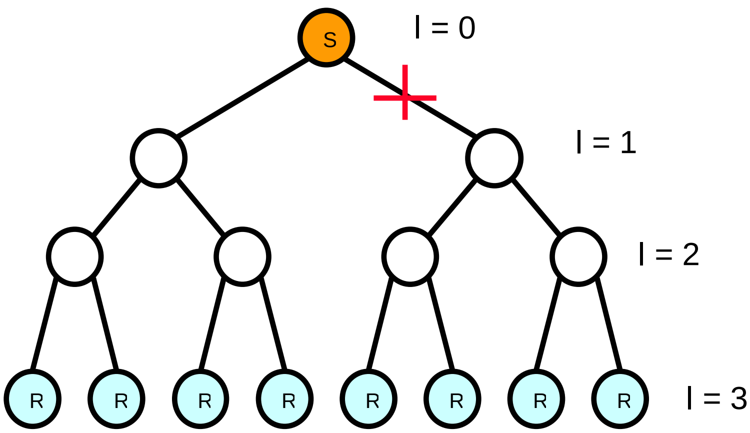
(b) See “Router Design” lecture; slides 12 and 13.

4) Scalable Reliable Multicast (SRM) (20 pts)

(a) **Briefly** describe SRM. (10 pts)

(b) Consider a **balanced binary** tree where the root represents the sender, and the leaves represent the receivers. Assume the propagation delay on each link is 1. Assume a packet is lost on a link between levels l and l+1. Levels are counted starting from root; the root is at the level l=0, and the leaves are at the level l=d.

Compute (1) the estimated time to receive the repair and (2) the estimated number of repair requests. Assume $C_1 = 0$, i.e., each receiver generates a repair request with a uniform distribution in the interval [0, C_2]. (**Note:** For simplicity assume $C_2 >= 2*d$.) (10 pts)



Example of a tree with depth d=3, where a loss happens on a link between levels 0 and 1.

(a) see paper.

(b) Let d1 be the depth of sub-tree that hangs of a link that experiences a loss. That is, if the loss occurs on a link between levels l and l+1, then d1 = d – (l+1). There are 2^{d1} receivers in this sub-tree. Thus, the expected time to generate the first repair request is $C2/(2^{d1}+1)$ and the time to receive the repair after the first repair request is generated is min(2*d, 4*(d1+1)) (depending whether the repair is received from the root or from a receiver in the other side of the tree). Thus, the expected total time to receive the repair is

$$C2/(2^{d1}+1) + \min(2*d, 4*(d1+1))$$

To compute the expected number of repair requests we use recursion. Let x_k be the expected number of requests generated in a sub-tree with height k. Note that the set of requests generated in a tree with height k are spread over an interval of at most 2*k time units (this is because it takes the first request 2*k time units to reach every other receiver in the sub-tree).

Next, we estimate the expected number of requests in a sub-tree of height k+1. Assume the first request is generated in the left sub-tree at time t. It will take 2*(k+1) to this request to reach the receivers in the right sub-tree. Assuming that requests in the right sub-tree are **uniformly** distributed in an interval I of size 2*k, the number of requests in the right sub-tree is $(k+2)*x_k/C_2$, where

- $2*x_k/C_2$ corresponds to the case when the interval I starts between t and t + 2. In this case no request in the interval I will be suppressed and there are x_k such requests
- $k*x_k/C_2$ corresponds to the case when the interval starts at a time t1, where t+2 <= t1 <= t+2*(k+1). In this case there will be $(t+2*(k+1)-t1)*x_k/C_2$ requests generated; integrating overt t1 we obtain $k*x_k/C_2$.

Thus, the expected number of requests generated in the sub-tree of depth k+1 is

$$x_{k+1} = x_k + x_k*(k+2)/C_2 = x_k*(1 + x_k*(k+2)/C_2)$$

where $x_0 = 1$.

As a result, we have

$$x_{d1} = 1*(1+2/C_2)*(1+3/C_2)*\ldots*(1+ (d1+1)/C_2)$$

Next, we give an alternate method to compute an upper-bound of x_{d1}.

Note that it takes the repair request message 2 time units to reach the leaf in the sub-tree at distance 2, 2*2 time units to reach the 2 leaves at distance 2*2. In general, it takes 2*k time units to reach the 2^{k-1} leaves at distance 2*k.

The expected numbers of repairs generated by the 2k-1 receivers at distance 2*k is simply $2^{k-1}*(2*k)/C2$

Thus, the expected number of repair requests generated before the first generated repair requests is received by every receiver in the tree that has experienced loss is

$$1 + 2*1/C2 + \ldots 2*d1*2^{d1-1}/C2 = 1 + 2*(1 + \ldots d1*2^{d1-1})/C2 = 1 + 2*((d1-1)*2^{d1}+1)/C2$$

5) Overlay/DHT (20 pts)

(a) Show that the average path length in Chord and Tapestry is O(log n).

(b) Compare recursive and iterative routing in terms of advantages and disadvantages.

(a) With Chord at every step the distance to the target is halved. With Tapestry at every hop at least one digit is fixed.

(b) Recursive routing: faster but less robust (if a node along the path fails, the lookup fails as well). Iterative routing: robust (if a node fails, the originator can try to use another node), but slower.