EE 607 Spring 17 Homework 6 Solution Total Points = 13 pts

Hualiang Li

Problem A. You will explore UNIX functions netstat and arp.

Log into spectra and do a manual on netstat

man netstat

This will give you an explanation of this function.

(a) (1 pt) After looking at the manual, list four related network functions (these are listed at the bottom of the manual, e.g., “arp”). What are the entries of the routing tables in quantum and spectra (type netstat -r)?

1. inet 2. iostat 3. nfsstat 4. Route

Destination Gateway Flags Refs Use Netif Expire

(b) (1 pt) The function arp will show the logical bindings between IP addresses and Ethernet addresses. Log into quantum and find the ethernet addresses for the kaala, quantum, spectra, and mahimahi.

```
hualiang@wiliki:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
mail.eng.hawaii.edu      ether   00:0d:60:1a:83:00   C                     eth0
dns2.eng.hawaii.edu      ether   00:14:5e:69:19:a2   C                     eth0
www4.eng.hawaii.edu      ether   00:1e:0b:c8:e9:42   C                     eth0
ldap2.eng.hawaii.edu     ether   00:1a:64:08:32:a2   C                     eth0
dns.eng.hawaii.edu       ether   00:0d:60:ec:77:82   C                     eth0
gateway                  ether   00:09:0f:c5:24:38   C                     eth0
192.168.101.251          ether   44:37:e6:66:23:cf   C                     eth0
backup.eng.hawaii.edu    ether   00:06:5b:fc:92:36   C                     eth0
waiono.eng.hawaii.edu    ether   1c:c1:de:1f:13:60   C                     eth0
```

Problem B (TCP Reno throughput versus probability of blocking *p*). (1 pt) This is a calculation using some very simplifying assumptions. However, it illustrates the relationship between TCP throughput and the packet loss probability p.

Consider a TCP Reno connection through a network and the following parameters

- All TCP packets have the same size equal to 1.
- We will assume that the TCP connection transmits packets in batches, where each batch is equal to the current window size. W_k is the average size of the TCP window in numbers of packets when the k^{th} batch of packets are transmitted.
- A packet is likely to be lost (or dropped) with probability p, which is constant. This is a simplifying assumption since p would increase with a function of packet traffic load. We will assume that p is small so that there is at most one packet loss in a batch. We will assume that the probability of a packet loss in the k^{th} batch is approximated by pW_k. The probability of no packet loss in the batch is $1 - pW_k$.
- T is the round trip time of a packet from the time of transmission until feedback is received about it.

If there is a single packet loss in a batch then the batch has window size 1/2 of the previous window due to TCP's fast retransmit and fast recovery. If there is no loss in the batch the the window size of the batch is one more than the previous window size due to TCP's congestion avoidance. This is expressed in the following equation

$$W_{k+1} = pW_k \cdot (0.5 \cdot W_k) + (1 - pW_k) \cdot (W_k + 1)$$

The next assumption is pretty big. Now let's compute the "steady state" value of the window size per batch *W*. It should satisfy

$$W = pW \cdot (0.5 \cdot W) + (1 - pW) \cdot (W + 1)$$

Solve for *W* (hint: quadratic equation wink wink). Use your solution to write a formula for throughput, which is *W/T*. *(Comment: There's quite a few big assumptions here but the formulas come out about right.)*

W=0.5pW^2+W+1-pW^2-pW
0.5pW^2+pW-1 = 0

W= (-p+sqrt(p^2+2p))/p since W > 0
Throughput = (-p+sqrt(p^2+2p))/pT

Problem C [1 pt]. In this problem you will derive Moore's bound:

Consider an arbitrary directed graph (V,E).

- Each node has at most *d* outgoing links.
- Let *s* be some node in *G*.
- Let *V*[*h*] be all nodes at most *h* hops away from *s* on a directed path from *s*.

Note that *V*[0] = {s}. *V*[1] includes {s} and all nodes that are one hop away from s on a directed path from s. Since s has at most *d* outgoing links, there is at most *d* nodes from *s* that are one hop away. Thus, |V[1]| ≤ d+1. (Recall, |X| is the cardinality of set X, i.e., the number members in set X.)

Similarly, there are at most nodes d^2 are two hops away from *s*. In general, $|V[h]| \le \sum_{k \le h} d^k$.

Let $\delta(u,v)$ be the shortest hop distance from *u* to *v*. Let the *diameter* of a graph *G* be the maximum $\delta(u,v)$ over all pairs of nodes {*u*,*v*}.

Use the inequality $|V[h]| \le \sum_{k \le h} d^k$ to find a lower bound on the diameter of *G*.

D <= 2h

Problem D (1 pt). Suppose you have a network of *N* nodes and *L* directed links, each with unit capacity. The network must support a *uniform traffic*, and let *r* denote the traffic rate for a source-destination pair.

(a) Find a lower bound on the minimum number of links *L* given *N* and *r*.

Use the following hints.
Hint 1: Assume the traffic is routed in some way, using simple routes, i.e., no repeated nodes. Let *T*(*h*) denote the total traffic in the network that exactly goes *h* hops. Then the total traffic in the network equals $\sum_{h=1}^{N-1} T(h)$. This must be less than or equal the total amount of bandwidth in the network. The total amount of bandwidth is equal to the number of links *L* multiplied by the link capacities, which is 1. Thus, $\sum_{h=1}^{N-1} T(h) \le L$.

Hint 2: Now we can derive a lower bound for the sum.

Note that a source-destination pair of nodes can have one-hop traffic if they are directly connected by a link. Thus, the number of source-destination pair of nodes that have one-hop traffic is equal to 2*L*. The total traffic generated by these pairs is $2L \cdot r$.

All other pairs of nodes have traffic that take two or more hops. The number of source-destination pairs that have two or more hops of traffic is $N(N-1)-2L$. The total traffic generated by these pairs is at least $(N(N-1)-2L)\cdot 2r$, where 2*r* corresponds to the traffic taking at least 2 hops.

Thus, the total traffic that the network must support is

Total one-hop traffic + Total multi-hop traffic, which is at least

$2L \cdot r$ + $(N(N-1)-2L)\cdot 2r$

Okay, with all this, you can get a lower bound for *L*.

2Lr+2N^2r-2Nr-4Lr <= L
2N^2r-2Nr<=(1+2r)L
L>=2Nr(N-1)/(1+2r)

(b) Fill out the following table. Note that the lower bound *L* is what you calculate should be integer.

Number of nodes	**Number of unidirectional links in a bidirectional ring**	**Number of unidirectional links in a bidirectional star topology**	**Lower bound L**
3	6	6	**4**
4	8	8	**8**
5	10	10	**13**
6	12	12	**20**
7	14	14	**28**
8	16	16	**37**
9	18	18	**48**
10	20	20	**60**

Problem E. (1 pt) In the lecture notes LectRouter, there is shown a way to compute the upper bound on the throughput of a network topology. This is shown on pages 37 and 38. Assume that the traffic is uniform, and the rate of traffic from a source node s to a destination node d is r. Also assume that the link capacities are 1, and denote the number of nodes in the network topology by N.

(a) Give the upper bound on *r* for the bidirectional ring. (Note that you may have two cases, for N even and odd.) Also give the upper bound on nodal throughput, which recall is $r \cdot (N-1)$

$r <= 2/(n/2)^2$

$r <= 8/N^2$

throughput <= rN

throughput <=8/N

(b) Give the upper bound on r for square grid topology, assuming N is the square of some integer. Also give the upper bound on the nodal throughput.

Let n = sqrt(N)

r<=n/(nx*(n-X))

when x is n/2, r is max

$r <= n^2/4$

So r<=N/4

throughput =r*n*n/2*n/2=4/n

So throughput <=4/sqrt(N)

(c) Give the upper bound on r for a star topology. Also give the upper bound on the nodal throughput.

r<=(N-2)*2+1/(N-1)=(2N-3)/(N-1)

throughput <= (N-1)/(2N-3)

(d) Fill in the following table, which compares the upper bounds for the two topologies

Number of nodes	**Upper bound on nodal throughput for a ring topology**	**Upper bound on nodal throughput for square grid**	**Upper bound on nodal throughput for star topology**
4	2	2	**3/5**
9	8/9	4/3	**8/15**
16	1/2	1	**15/29**
25	8/25	4/5	**0.5**
36	2/9	2/3	**0.5**
49	8/49	4/7	**0.5**
64	1/8	1/2	**0.5**
81	8/81	4/9	**0.5**
100	2/25	2/5	**0.5**

Problem F. Consider a bipartite multigraph (i.e., pairs of nodes can have more than one edge) with n nodes on the left and right. A *proper coloring* of the links is an assignment of colors to the links so that any node has incident links with distinct colors. For simplicity, assume the colors are positive numbers 1, 2, ...

(a) (1 pt) Consider a bipartite multigraph B with a proper link coloring. Let's add a link (L,R) which has left node L and a right node R. Let B* denote this new multigraph, and let d denote the maximum degree of B*. Show that the new link (L,R) can be colored with a color at most 2d-1 without changing the colors of the other links.

Proof by contradiction:

Let suppose there are more than 2d-1 colors, which is at least 2d colors, because the graph has a degree of d, with 2 nodes construct a link, the color is at most 2d, this conflicts with the assumption that at least 2d can be colored. Therefore, new link (L,R) can be colored with a color at most 2d-1.

(b) Suppose the link coloring of B uses at most *d* colors (the degree of B*), and without loss of generality assume that the colors are {1, 2, ..., *d*}. Assume (L,R) is uncolored. Show that you can color (L,R) with one of the *d* colors, though you may have to recolor the other links. In addition, at most n-1 links are recolored.

Here's an argument to prove the above. (The example on the last page will help to visualize the concepts.)

Since nodes L and R have degree at most d, they have at most d incident links. At most d-1 of these links are colored since (L,R) is uncolored. This means that each node has a color that is not used by any of its incident links. We shall refer to the color for node L by C(L), and the color for node R by C(R).

As an example, suppose d = 4, and L and R both have degree 4. Since (L,R) is uncolored, then only three of the incident links have colors. Suppose node L has incident links with colors {1, 2, 4}, and node R has incident links with colors {2, 3, 4}. Then C(L) can be 3, and C(R) can be 1. We refer to C(L) to be a "free color" for node L because if we use it to color an uncolored incident link, node L will still have its incident links with distinct colors. We refer to C(R) to be a “free color” for node R for the same reason.

There are two cases. The first case is when C(L) = C(R). Then we can color the link (L,R) to be C(L) (= C(R)). Then the links incident to L and R have distinct colors. So we have a proper link coloring using at most d colors, and we are done.

The second case is when C(L) is not equal to C(R). This is the more difficult case and requires recoloring links. Let B’ denote the sub-multigraph of B composed of the links colored {C(L), C(R)} and all the other links are ignored. Note that the degrees of the nodes in B’ are at most two. Thus, B’ is composed of disjoint paths and cycles. Let P(L) be the path or cycle that contains node L.

(b.i) (1 pt) Show that P(L) must be a path (i.e., it cannot be a cycle) with L as an end node.
Proof by contradiction
Suppose there is a cycle on path P(L), then, it means P(L) contains at least 2 links with node on L, because these two links have different color, they have to be either C(L) and C(R), which means in the original graph B, it contains C(L) and C(R), this is contradicted with the assumption that C(L) has to be free color.
(**b.ii**) (1 pt) Show that P(L) cannot contain R.
If P(L) contains R, this means link L->R has been colored. Which is not true. Therefore, the P(L) cannot contain R.

(**b.iii**) (1 pt) Note that, starting from L, links of the path P(L) are colored C(R), C(L), C(R), C(L),...... Explain how you can recolor the links along P(L) so that the multigraph B maintains a proper link coloring and none of the links incident to L are colored C(R).

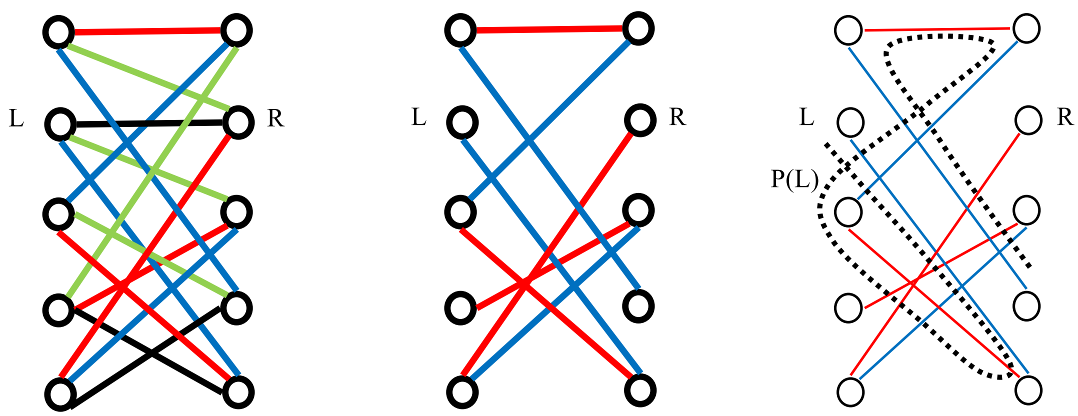
Note that after recoloring the links, both L and R have C(R) as a free color. Then (L,R) can be colored C(R).
Starting from node L, we replace the color of link C(R) with C(L), the switch back to C(R) in path P(L) until the end of this path. Then we can color link R->L with C(R).
(**b.iv**) (1 pt) Note that the length of P(L) is the number of recolorings. What is the maximum possible length of P(L)?
2(n-1), becasue R is not in path P(L), so at most (n-1) nodes are in path P(L), each node has most 2 links.
(b.v) (1 pt) To reduce the number of link recolorings, we can also consider recoloring the path of B’ that is connected to node R. Let P(R) denote this path. Thus, to minimize the recolorings, we should recolor the shorter of the paths P(L) and P(R). Show that the maximum possible length of the shorter path is n-1.
Because P(L) + P(R) <= 2n-1, so the shorter path has to be not greater than n-1
Hints (examples) for Problem B

Here is an example for part (b). The first picture on the left shows a bipartite graph with degree d = 3 that is partially colored with red, blue, and green. The uncolored links are shown as black. Note that node L has two links colored green and blue, and R has links colored green and red. Recall C(L) is some color not incident to node L. Red is the only such color of the existing three colors, so we choose C(L) = red. Similarly, we choose C(R) = blue.

The present link coloring does not allow us to color the link (L, R) with any of the three existing colors. But the following will allow us to recolor links to allow (L,R) to be colored with one of the existing colors.

To simplify this example, the middle picture is the bipartite graph B* that just shows only the links colored with {C(L), C(R)}, which is {red, blue}. The picture on the right shows the path P(L) as a dotted line; the path follows the colors {red, blue}.

Note that if we switch the colors on the path (blue to red and red to blue), the link coloring will remain valid. L will have its blue link turn red. Then link (L, R) can be colored blue.



Problem G. (1 pt) Consider a 3-stage Clos network. The network has

1. N inputs and N outputs, numbered 1, 2, ..., N
2. r first stage switches, labeled F_1, F_2 ,..., F_r . The value r is assumed to divide N.
3. k second stage switches, labeled S_1, S_2,...., S_k
4. r third stage switches, labeled T_1, T_2, ..., T_r.

Assume the Clos network is empty, and there are a set of connections to set up, which refer to as a *connection request*. A connection request is denoted by a pair (i, j), where i is the input and j is the output of the connection. Show that the problem of setting up the connections in a 3-stage Clos network is equivalent to coloring edges in a bipartite multigraph (recall that a multigraph can have multiple edges between nodes).

Hint: First determine how a connection can be set up in the network, i.e., what must be specified in the network for the connection to follow.

To connect from i to j, we should avoid the link that has been in use, we have to choose next stage swith that has idel link between them. This is same with the link coloring problem when we can not color a same color on a same node.