CMPS 4750/6750 Computer Networks – Fall 17

Homework 4 - Solution

Due 05/02/17 at the beginning of the class

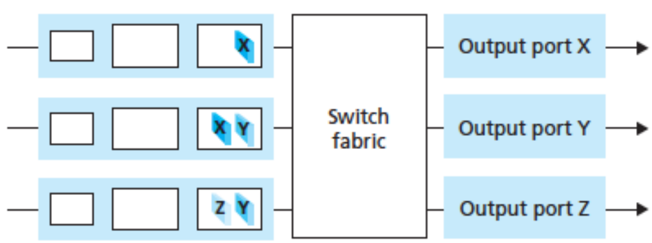
1. (6 points) **(switching)** Consider the switch shown below. Suppose that all datagrams have the same fixed length, that the switch operates in a slotted, synchronous manner, and that in one time slot a datagram can be transferred from an input port to an output port. The switch fabric is a crossbar so that at most one datagram can be transferred to a given output port in a time slot, but different output ports can receive datagrams from different input ports in a single time slot. What is the minimal number of time slots needed to transfer the packets shown from input ports to their output ports in the following two cases, respectively?

   (a) packets are served in a first-come-first-served (FCFS) manner

   (b) packets can be served in any order you want (i.e., it need not have HOL blocking)

   Solution: (a) Using FCFS, the minimal number of time slots needed is 3. One possible schedule is as follows. Slot 1: send X in top input queue, send Y in middle input queue. Slot 2: send X in middle input queue, send Y in bottom input queue. Slot 3: send Z in bottom input queue.

   (b) If one can use any scheduling policiy, the minimal number of time slots needed is 2. One possible schedule is as follows. Slot 1: send X in top input queue, send Y in middle input queue, and send Z in the bottom input queue. Slot 2: send X in middle input queue, send Y in bottom input queue.



2. (8 points) **(longest prefix matching)**

   Consider a datgram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11000000 00000000 00000000 00000000 through 11000000 00111111 11111111 11111111	0
11000000 01000000 00000000 00000000 through 11000001 01111111 11111111 11111111	1
11000001 10000000 00000000 00000000 through 11000001 11111111 11111111 11111111	2
otherwise	3

Provide a forwarding table that uses longest prefix matching, and forwards packets to the correct link interfaces. You may use more than one extries for a prefix, but try to make the forwarding table as small as possible.

Solution: See the following table for one possible solution.

Prefix Match	Link Interface
11000000 00	0
1100000	1
11000001 1	2
otherwise	3

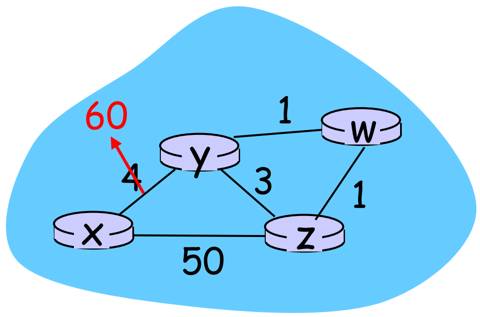
3. (6 points)**(subnets)**

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support at least 120 interfaces, Subnet 2 is required to support at least 30 interfaces, and Subnet 3 is required to support at least 60 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisify these constraints.

Solution: One possible solution is as follows:
Subnet 1: 223.1.17.0/25
Subnet 2: 223.1.17.128/27
Subnet 3: 223.1.17.192/26

4. (10 points)**(distance-vector routing)**

Consider the following network with 4 routers. The initial costs of all links are given as follows: $c(x,y) = 4, c(x,z) = 50, c(y,w) = 1, c(z,w) = 1, c(y,z) = 3$. Suppose that poisoned reverse is used in the distance-vector routing algorithm. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if positioned reverse is used? Justify your answer.



Solution: Yes, there will be a count-to-infinity problem. The following table shows the routing converging process. Assume that at time t0, link cost change happens. At time t1, y updates its distance vector and informs neighbors w and z. In the following table,"$\rightarrow$" stands for "informs". We see that w, y, z form a loop in their computation of the costs to router x.

	t0	t1	t2	t3	t4
z	→ w, $D_z(x)=\infty$ → y, $D_z(x)=6$		No change	→ w, $D_z(x)=\infty$ → y, $D_z(x)=11$	
w	→ y, $D_w(x)=\infty$ → z, $D_w(x)=5$		→ y, $D_w(x)=\infty$ → z, $D_w(x)=10$		No change
y	→ w, $D_y(x)=4$ → z, $D_y(x)=4$	→ w, $D_y(x)=9$ → z, $D_y(x)= \infty$		No change	→ w, $D_y(x)=14$ → z, $D_y(x)= \infty$

If we continue the iterations shown in the above table, then we will see that, at t27, z detects that its least cost to x is 50, via its direct link with x. At t29, w learns its least cost to x is 51 via z. At t30, y updates its least cost to x to be 52 (via w). Finally, at time t31, no updating, and the routing is stabilized.

	t27	t28	t29	t30	t31
z	→ w, $D_z(x)=50$ → y, $D_z(x)=50$				via w, ∞ via y, 55 via z, 50
w		→ y, $D_w(x)=\infty$ → z, $D_w(x)=50$	→ y, $D_w(x)=51$ → z, $D_w(x)= \infty$		via w, ∞ via y, ∞ via z, 51
y		→ w, $D_y(x)=53$ → z, $D_y(x)= \infty$		→ w, $D_y(x)= \infty$ → z, $D_y(x)= 52$	via w, 52 via y, 60 via z, 53