**EE122: Introduction to Computer Networks** Homework 4
Department of Electrical Engineering and Computer Sciences SOLUTIONS
UNIVERSITY OF CALIFORNIA BERKELEY

**Total points = 10**

**Question 1** (2 points)

1. No VC number can assigned to the new VC; thus the new VC cannot be established in the network.
2. Each link has two available VC numbers. There are four links. So the number of combinations is $2^4 = 16$. One example combination is $(10, 00, 00, 10)$.

**Question 2** (1 point)

| *Destination Address* | *Link Interface* |
|---|---|
| 224/8 | 0 |
| 225.0/16 | 1 |
| 225/8 | 2 |
| otherwise | 3 |

**Question 3** (2 points) This problem was actually intended for solving the problem of finding the maximum bottleneck bandwidth path from a source to all nodes in the network.
**Definition:** *Bottleneck bandwidth* of a path is the minimum bandwidth of all the links along that path.
If we wish to maximize the bottleneck bandwidth, we can use Dijkstra's algorithm by taking the max bottleneck bandwidth path to be the metric (represented as $B(s)$) and maximizing it at each step by updating the bottleneck bandwidth of the nodes attached to the current node by taking the minimum of the link bandwidth to the attached node and the bottleneck bandwidth to our current node. ($p(s)$ represents the previous node along the max bottleneck path).

| *Step* | N′ | $B(s), p(s)$ | $B(t), p(t)$ | $B(u), p(u)$ | $B(v), p(v)$ | $B(w), p(w)$ | $B(y), p(y)$ | $B(z), p(z)$ |
|---|---|---|---|---|---|---|---|---|
| 0 | x | 0 | 0 | 0 | 3,x | 1,x | 6,x | 0 |
| 1 | xy | 0 | 4,y | 0 | 3,x | 1,x | | 6,y |
| 2 | xyz | 0 | 4,y | 0 | 3,x | 1,x | | |
| 3 | xyzt | 1,t | | 2,t | 4,t | 1,x | | |
| 4 | xyztv | 1,t | | 2,t | | 1,x | | |
| 5 | xyztvu | 2,u | | | | 2,u | | |
| 6 | xyztvuw | 2,u | | | | | | |

Note that just maximizing using Dijkstra's does **NOT** give an absolute maximum sum of bandwidths path between any two nodes. Instead, we are maximizing the sum at every step given the links and nodes that we know of at that step, not considering the fact that the future steps may reveal a more profitable path. Hence, it may be viewed as a greedy maximization. However, due to a misunderstanding (and "mis-wording") of the problem, we will consider that solution too here. (Its EECS122 you know!!) In this case, the $B'(s)$ is the greedy maximum sum. Also, note that though the order in which the nodes are added to the set $N'$ remains the same, the "path" generated is different.

| N′ | $B'(s),p(s)$ | $B'(t),p(t)$ | $B'(u),p(u)$ | $B'(v),p(v)$ | $B'(w),p(w)$ | $B'(y),p(y)$ | $B'(z),p(z)$ |
|---|---|---|---|---|---|---|---|
| x | 0 | 0 | 0 | 3,x | 1,x | 6,x | 0 |
| xy | 0 | 10,y | 0 | 7,y | 1,x | | 20,y |
| xyz | 0 | 22,z | 0 | 3,x | 1,x | | |
| xyzt | 23,t | | 24,t | 31,t | 1,x | | |
| xyztv | 23,t | | 32,v | | 32,v | | |
| xyztvu | 36,u | | | | 35,u | | |
| xyztvuw | 36,u | | | | | | |

**Question 4** (2 points)

1. $D_x(y) = 4, D_x(w) = 1, D_x(u) = 6$
2. First consider what happens if $c(x, y)$ changes. If $c(x, y)$ becomes larger or smaller (as long as $c(x, y) > 0$), the least cost path from x to u will still have cost 6 and pass through w. Thus a change in $c(x, y)$ will not cause x to inform its neighbors of any changes. Now consider if c(x,w) changes. If $c(x, w) = \epsilon \leq 5$, then the least-cost path to u continues to pass through w and its cost changes to $5 + \epsilon$; x will inform its neighbors of this new cost. If $c(x, y) = \delta > 5$, then the least cost path now passes through y and has cost 10; again x will inform its neighbors of this new cost.
3. Any change in link cost c(x,y) will not cause x to inform its neighbors of a new minimum-cost path to u .

**Question 5** (3 points) From $N$ input packets (for the same output), $L$ packets are chosen for the output in L stages. First stage involves a competition to eliminate $N - 1$ packets. Each losing packet originates from a unique $2X2$ switching element giving $N - 1$ switching elements in the first stage. In the same way, we get $N - 2$ elements in the second stage, $N - 3$ elements in the third stage and so on, giving
$\sum_{i=1}^{L}(N - i) = LN - \frac{L(L+1)}{2}$
for each output. Therefore, total number of switching elements required will be $LN^2 - \frac{NL(L+1)}{2}$