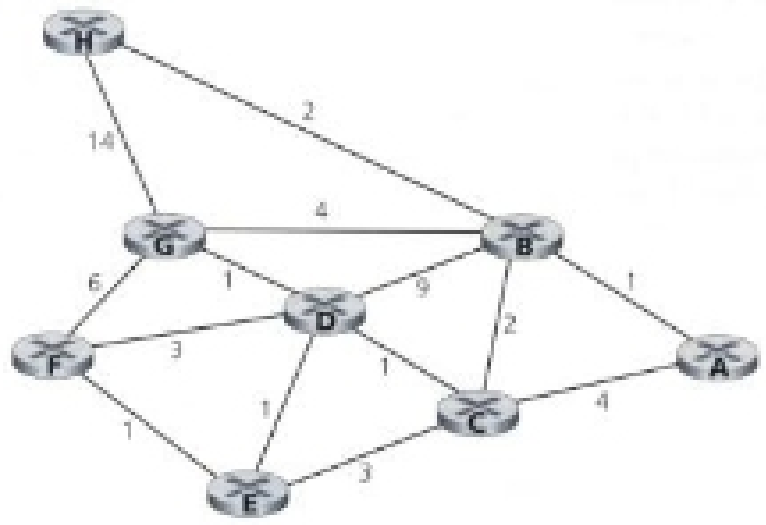

Figure 1: Network with edge weights

Step	N'	D(A),p(A)	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(G),p(G)	D(H),p(H)
0	F	∞	∞	∞	3,F	1,F	6,F	∞
1	FE	∞	∞	4,E	2,E		6,F	∞
2	FED	∞	11,D	3,D			3,D	∞
3	FEDC	7,C	5,C				3,D	∞
4	FEDCG	7,C	5,C					17,G
5	FEDCGB	6,B						7,B
6	FEDCGBA							7,B

Table 1: Solution table

Problem 1

In this problem we'll explore the impact of NATs on P2P applications. Suppose a peer with username Alice discovers through querying that a peer with username Bob has a file it wants to download. Also suppose that Bob and Alice are both behind a NAT. Is it possible to devise a technique that will allow Alice to establish a TCP connection with Bob without application-specific NAT configuration? Why or why not?

It is not possible to devise such a technique. In order to establish a direct TCP connection between Alice and Bob, either Alice or Bob must initiate a connection to the other. But the NATs covering Alice and Bob drop SYN packets arriving from the WAN side. Thus neither Alice nor Bob can initiate a TCP connection to the other if they are both behind NATs.

Problem 2

If you have a network that looks like Figure 1 and you use the link weights shown, use Dijkstra's shortest-path algorithm to compute the shortest path from F to all network nodes. Show how the algorithm works by computing a table like Table 4.3 (4th:page 373, 5th: page 379) in your book.

See Table 1.

Problem 3

Using the network in Figure 2, x has only 2 attached neighbors, w and y. w has a minimum-cost path to destination u (not shown) of 5, and y has a minimum-cost path to u of 6. The complete paths from w and

Problem 3 continued on next page...

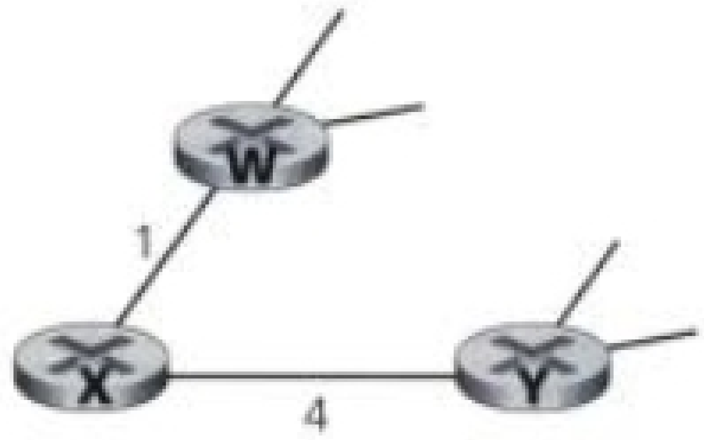

Figure 2: Fragment of network.

y to u (and between w and y) are not shown. All link costs in the network have strictly positive integer values.

a. Find x's distance vector for destinations w, y, and u.

b. Find a case of link-cost change (if any) for either $c(x, w)$ or $c(x, y)$ such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance vector algorithm.

c. Find a case of link-cost change (if any) for either $c(x, w)$ or $c(x, y)$ such that x will *not* inform its neighbors of a new minimum-cost path to u as a result of executing the distance vector algorithm.

a. $\mathrm{D}_x(\mathrm{y}) = 4$, $\mathrm{D}_x(\mathrm{w}) = 1$, $\mathrm{D}_x(\mathrm{u}) = 6$

b. 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) $= \epsilon \leq 1$, 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.

c. Any change in link cost c(x,y) will not cause x to inform its neighbors of a new minimum-cost path to u.