Assignment #5

CSc 4220/6220 - Computer Networks

November 16, 2017

→ ***You should submit your own work with programming assignment.***

1. Consider the topology shown in Figure 1. Denote the three subnets with hosts (starting clockwise at 12:00) as Networks A, B, and C. Denote the subnets without hosts as Networks D, E, and F.

   (a) Assign network addresses to each of these six subnets, with the following constraints: All addresses must be allocated from 214.97.254/23; Subnet A should have enough addresses to support 250 interfaces; Subnet B should have enough addresses to support 120 interfaces; and Subnet C should have enough addresses to support 120 interfaces. Of course, subnets D, E and F should each be able to support two interfaces. For each subnet, the assignment should take the form a.b.c.d/x or a.b.c.d/x e.f.g.h/y.

   (b) Using your answer to part (a), provide the forwarding tables (using longest prefix matching) for each of the three routers.

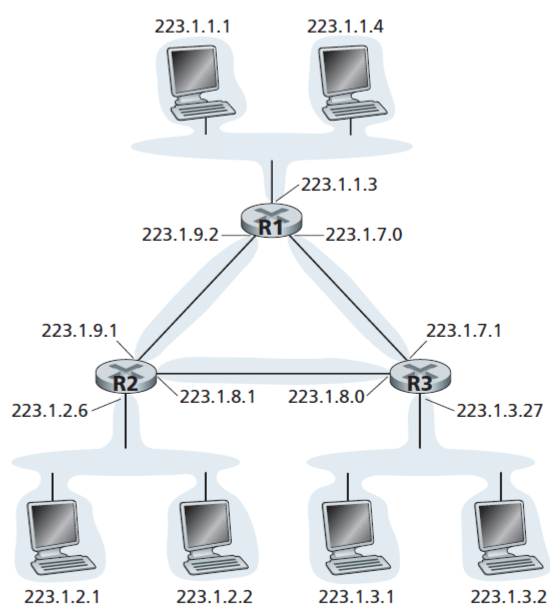

Figure 1: Network topology

2. Consider the following network in Figure 2. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1 in the textbook.

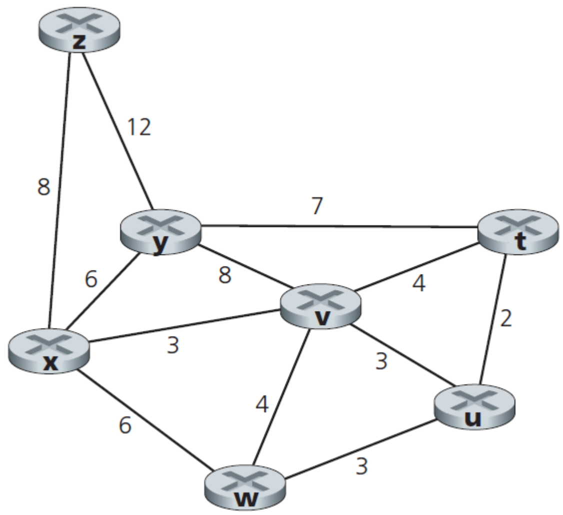

Figure 2: Network topology

3. Consider the network shown in Figure 3, and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.

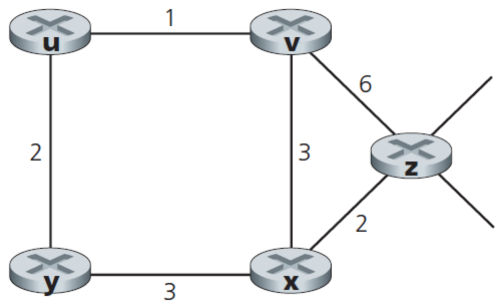

Figure 3: Network topology

4. Consider the network fragment shown in Figure 4. x has only two 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 y to u (and between w and y) are not shown. All link costs in the network have strictly positive integer values.

   (a) Give x's distance vector for destinations w, y, and u.

   (b) Give a link-cost change 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) Give a link-cost change 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.

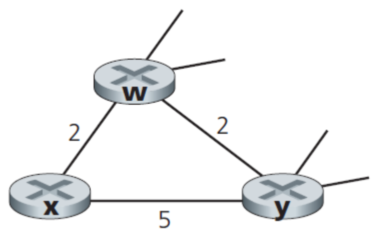

Figure 4: Network topology

5. Consider the three-node topology shown in Figure 5. Rather than having the link costs shown in Figure 4.30, the link costs are $c(x, y) = 3, c(y, z) = 6, c(z, x) = 4$. Compute the distance tables after the initialization step and after each iteration of a synchronous version of the distance-vector algorithm (as we did in our earlier discussion of Figure 5).

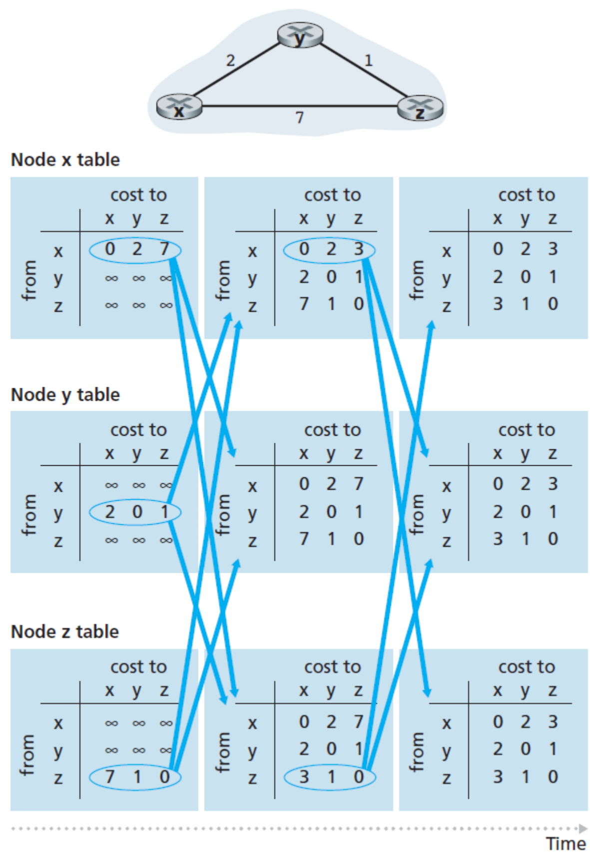

Figure 5: Network topology

6. ***Programming Assignment:***

Based on the content covered in the class, you are required to implement a simplified network store-forward simulating program. You can conduct the homework by yourself or work with another student as a group. You are required to implement shortest path algorithm (Dijkstra) for the graph, which is shown in Figure 6.

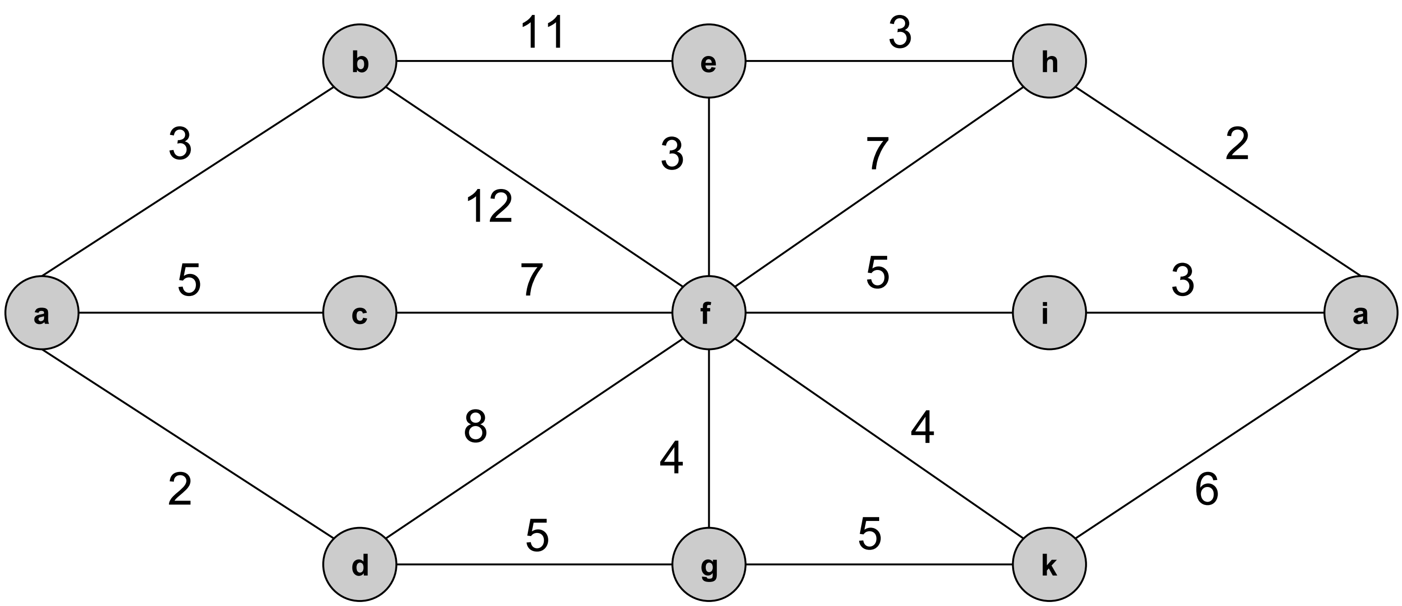

Figure 6: Network graph model

In the figure, a number along a link shows the cost of each link. To find a path from one node to another node, you should use Dijkstra's shortest path algorithm.

- Inputs of the program:
  - Our specific graph model
  - Starting point (Sender)
  - Ending point (Receiver)
- Outputs of the program:
  - Cost of the path
  - Visited nodes on the path

You have to:

(a) Implement algorithm (with your preferred programming language) running on the control plane.

(b) Submit all your code blocks into the iCollege

(c) Demonstrate your program to the TA during the TA's office hour