



**COSC 328 – LAB 6**  
***Introduction to Networks***  
2020 Winter Term 1

---

**Deadline:** November 20<sup>th</sup>, 2020 at 11:59 PM Pacific Time (Sharp). Delayed assignments will receive penalty as described in the course outline.

**Introduction**

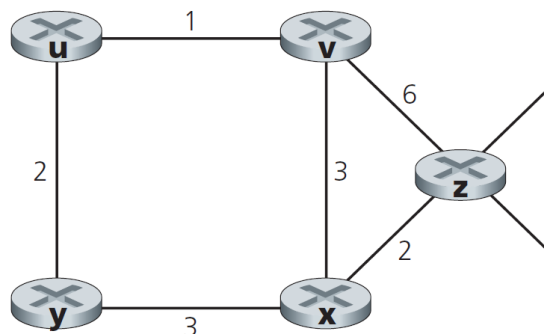
In this lab, we will do some practice questions on the network layer, control plane, by covering the control-plane component of the network layer along with traditional routing algorithms and SDN controllers, ICMP, and SNMP. All work must be shown for marks. This lab should be electronically submitted on Canvas.

**Warm-up Problems:** The authors of the text have provided interactive problems to help you with understanding the concepts presented in chapter 5. Please take the time to review these problems before attempting the questions as they will help. These problems are for your own learning and practice (no marks) but will allow you to review solutions.

- [Dijkstra's Link State Algorithm](#)
- [Dijkstra's Link State Algorithm - Advanced](#)
- [Bellman Ford Distance Vector algorithm](#)
- [Openflow Flow Tables](#)

**Answer the following questions: (40 marks)**

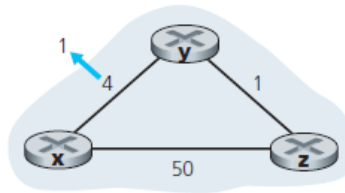
Question 1) Consider the network shown below and assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z. **(10 marks)**



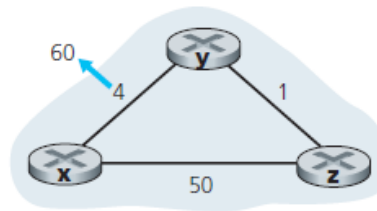


Question 2) Consider the right-side Figure. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows:  $c(x,y) = 4$ ,  $c(x,z) = 50$ ,  $c(y,w) = 1$ ,  $c(z,w) = 1$ ,  $c(y,z) = 3$ . Suppose that poisoned reverse is used in the distance-vector routing algorithm. **(10 marks)**

- When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?
- Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.
- How do you modify  $c(y,z)$  such that there is no count-to-infinity problem at all if  $c(y,x)$  changes from 4 to 60?



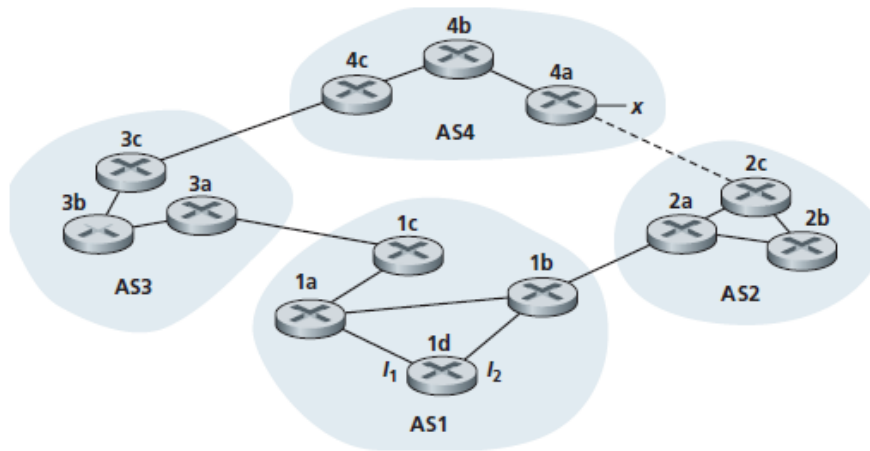
a.



b.

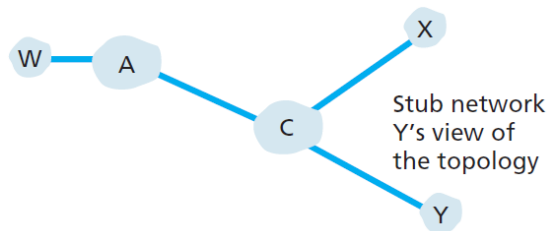
Question 3) Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4. **(10 marks)**

- Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?
- Router 3a learns about x from which routing protocol?
- Router 1c learns about x from which routing protocol?
- Router 1d learns about x from which routing protocol?



Question 4) In Figure 5.13 of the textbook, consider the path information that reaches stub networks W, X, and Y. Based on the information available at Wand X, what are their respective views of the network topology? Justify your answer. The topology view at Y is shown below. **(10 marks)**

Show the X's view and W's view of the topology (It will be similar to a tree).



Here is the figure 5.13 for your reference:

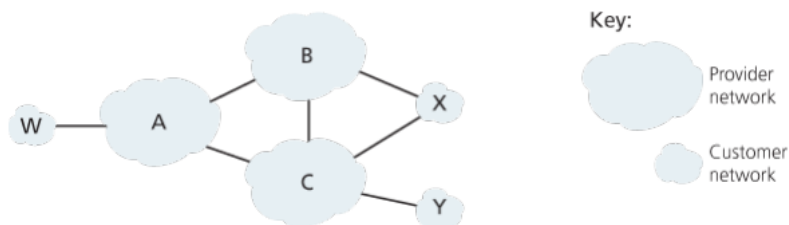


Figure 5.13 A simple BGP policy scenario



## Let's have some fun (20 marks)

Two python programs are uploaded with this assignment. The first one is a class to create directed graphs with weights on the edges. This is named as graphSuff.py. Another file is the one that you can use to answer the following questions. It is the DijkstraCOSC328.py file (which imports the graphSuff). You probably need to install a few packages as commented in the file. In this file, two implementations of the Dijkstra's algorithm with arrays and heaps are provided. As we discussed in class, the complexity of the Dijkstra's algorithm is  $O(n^2)$ , but better implementations can reduce this cost. Follow the program to investigate how to create a graph and then traverse it to find the shortest paths from one node to all other nodes. You will be able to print the shortest path with different distances from a node to other nodes.

This is an example of a random graph with 5 nodes and its edges:

Vertices:

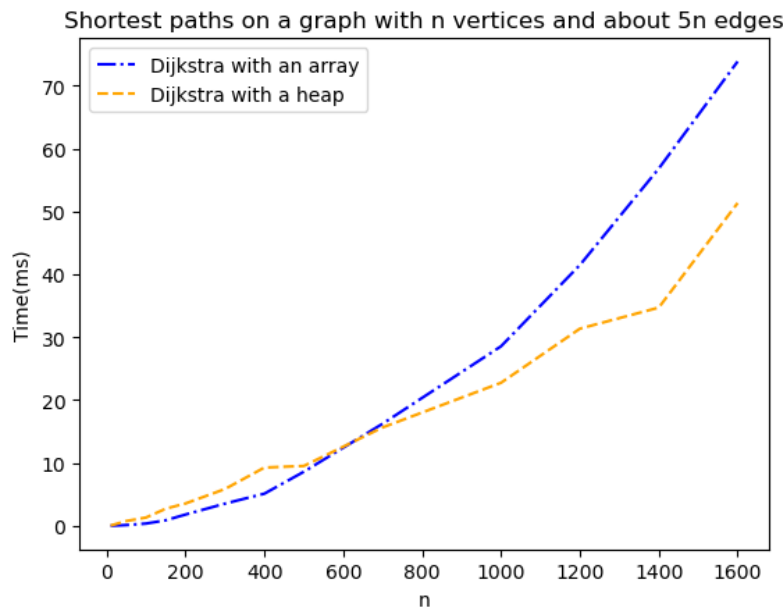
0,1,2,3,4,

Edges:

(1,0; wt:4) (3,2; wt:1) (3,4; wt:3)

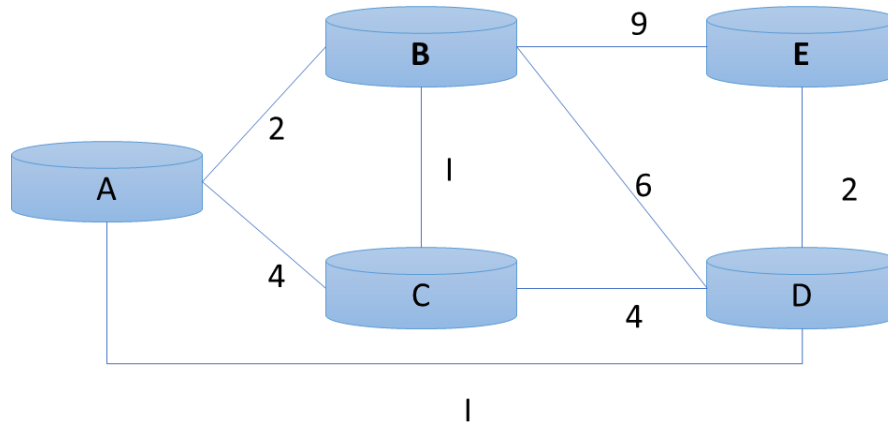
Each edge is shown as the tuple. The first element is the source node, the second element is the destination node, and the third element shown by wt is the weight of the link from source to destination.

When you run the program, you should be able to see a plot similar to the following which shows the running time of the two implementations to find the shortest paths with randomly generated graphs with  $n$  nodes and  $5n$  edges, where  $n$  is in  $[10,50,100,150,200,300,400,500,700,1000,1200,1400,1600]$ .





- a) Use this program to generate the graph shown below and find the shortest path from node A to all other nodes. This is the same graph that you have practiced in class. Show the output of the generated graph in your terminal and paste the figure as your answer. **(10 marks)**



- b) Find the shortest path of the graph in Question 1, from node z to all other nodes. Show the output of the generated graph in your terminal and paste the figure as your answer. **(5 marks)**
- c) Interpret the generated plot that compares the two implementations of the same algorithm in two-three sentences. **(5 marks)**

Do you want to practice more? Find the shortest paths for Question 2, part a and b. This part has no marks and is for your practice.