



HKUSTx: ELEC1200.3x A System View of Communications: From Signals to Packets (Part 3)



Bookmarks

- ▶ Pre-course Materials
- ▶ Topic 1: Course Overview
- ▶ Topic 2: The Link Layer
- ▶ Topic 3: The Network Layer

▼ Topic 4: Routing**4.1 Routing****4.2 Routing: Distance Vector Algorithm**

Week 2 Quiz due Feb 01, 2016 at 15:30 UTC


**4.3 Routing: Routing Link State Algorithm**

Topic 4: Routing > 4.5 Lab 2: Network Layer > Lab 2 - Task 4




Bookmark


LAB 2 - TASK 4 (EXTERNAL RESOURCE) (1.0 points possible)

Week 2 Quiz due Feb 01,
2016 at 15:30 UTC 

4.4 Summary of Routing Algorithms

Week 2 Quiz due Feb 01,
2016 at 15:30 UTC 

4.5 Lab 2: Network Layer

Lab due Feb 01, 2016 at
15:30 UTC 

- MATLAB download and tutorials

Lab 2 - Task 4

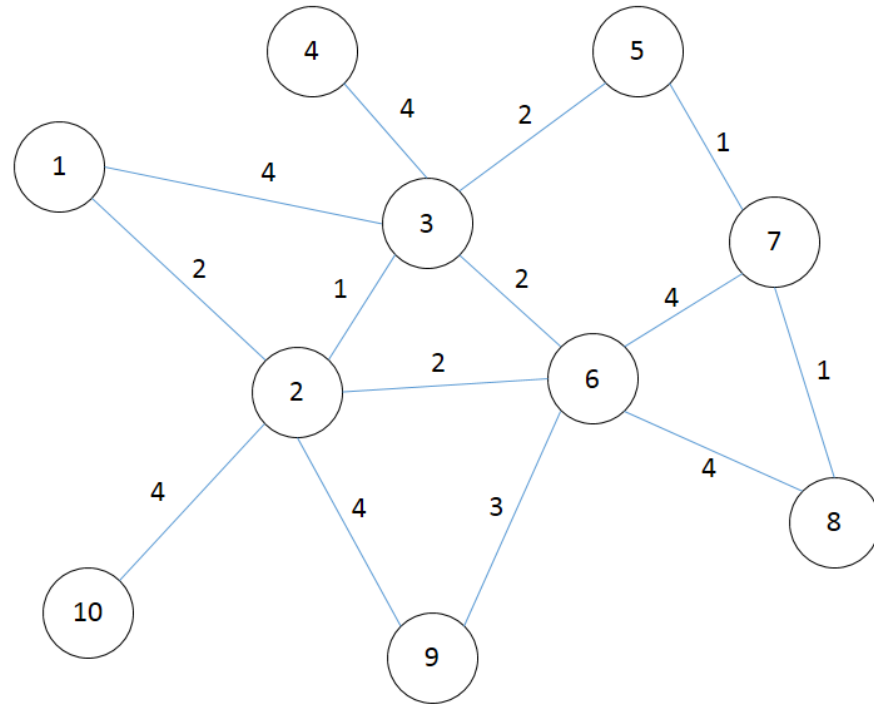
In this task, we will learn how Dijkstra's algorithm used in the link-state routing can be used to determine the shortest path between any two nodes.

INSTRUCTIONS

Routing using the link-state algorithm is different than routing using the distance vector algorithm. In the link-state algorithm, each node first estimates the graph of the complete network at every node. The information about the complete network is obtained by message flooding, where each node broadcasts information about its connections to its neighbors, and these neighbors then forward that information on and so on. This information is propagated globally over the entire network, allowing each node to construct an estimate of the complete network. Given the graph of the complete network, each node then computes the shortest path to every other node in the network using Dijkstra's algorithm. It then uses this information to construct a routing/forwarding table. On the other hand, nodes in the distance vector algorithm never estimate the complete network. Although nodes know the cost of the route to get to each of the nodes through their neighbors, they do not know anything about the path taken by the packets after they arrive at their neighbors. The distance vector algorithm is a decentralized algorithm where each node broadcasts its own routing table iteratively. However, this information is not re-broadcast explicitly, but rather only through changes in its neighbors' routing tables.

The MATLAB code in the window below uses the Dijkstra algorithm to compute the shortest path between two nodes. The implementation of the Dijkstra's algorithm is too complicated to be covered in this course exercise, your task is to learn how to represent the network connection by a matrix and input the matrix to the function implementing Dijkstra's algorithm (**getPathDijkstra**).


In order to run Dijkstra's algorithm and get credit for this task, you must set the value of the variable `graph` which is initialized to represent the networks used in Task 1 through 3, so that it represents the network shown in the figure below. We have already discussed the matrix representation of a graph in Task 2 and video. Once you have revised the current matrix, try to change the source and the destination nodes and check whether the retrieved path is the one with the smallest cost.



If you are looking for a challenge, try implementing your own version of Dijkstra's algorithm and check the results of `getPathDijkstra`.

Your Solution

 Reset

 MATLAB Documentation (<https://www.mathworks.com>)

```
12
13 % find the shortest path from source to destination
14 source = 8;
15 dest = 10;
16
17 [path, cost] = getPathDijkstra(graph, source, dest);
18 path
19 cost
20
```



Assessment Tests: Passed

✓ Is the graph correct?

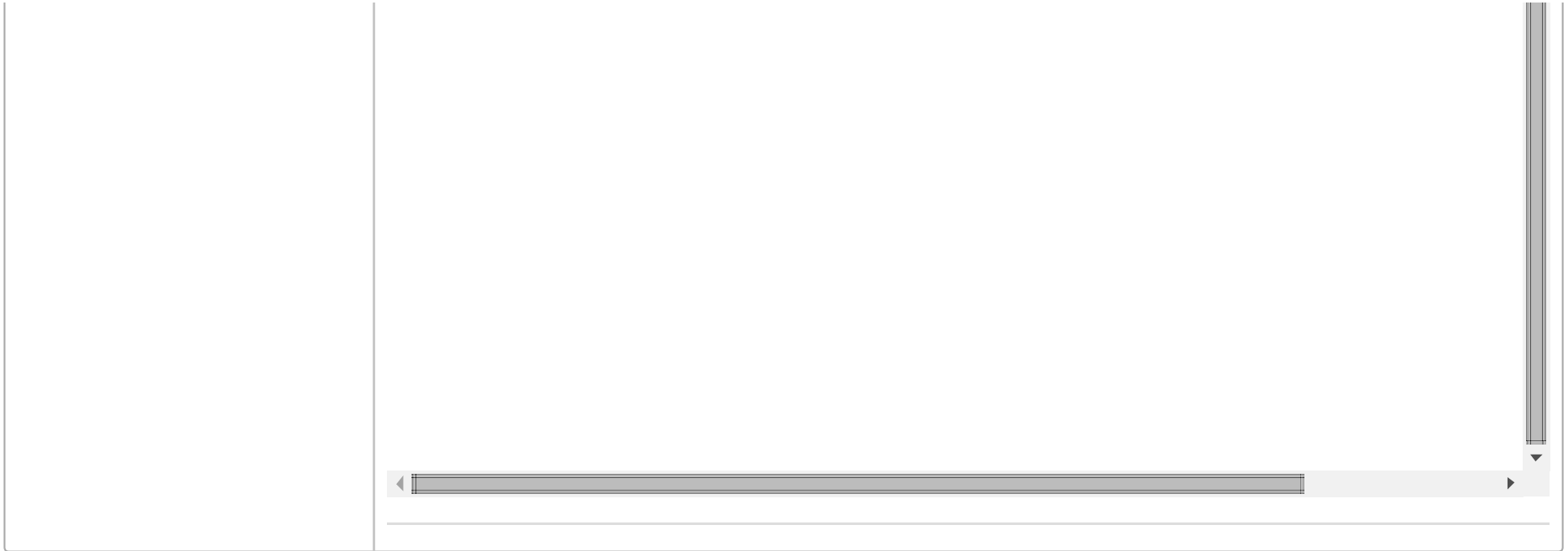
Output

path =

8
7
5
3
2
10

cost =

9



© All Rights Reserved



© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX



