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

Bookmarks

Topic 4: Routing > 4.5 Lab 2: Network Layer > Lab 2 - Task 3

Bookmark

LAB 2 - TASK 3 (EXTERNAL RESOURCE) (0.0 / 1.0 points)

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 3

In Task 2, we initialized the routing table for all nodes using local information about each node and neighbors. In this task, we will learn how each node can use the distance vector algorithm to updat table iteratively based on its current routing table and information (advertisements) from its neares Eventually, the routing tables will converge to the optimal routing tables for the network that will ena to get from a source to a destination node following a path with least cost.

INSTRUCTIONS

In this exercise, your task is to revise the code to construct the routing table of all nodes by iterativ broadcasting and updating the tables. For ease of simulation, we assume the broadcasting proces and the tables of other nodes will be available in each iteration.

The initial MATLAB code first loads the three variables that we initialized in Task 2:

- **neighbors{i}**: contains the list of neighbors of node **i**
- **costs{i}**: contains the cost of the links to the neighbors
- **RT{i}**: contains the initial routing table we obtained in Task 2.

Then, the code executes the main loop for a specified maximum number of iterations **max_time**. B have a fairly simple network, the routing tables will converge quickly to their optimal values, so we s to a small value.

Within each iteration, the algorithm then loops over all nodes in the network.

In practice, each node **n** receives advertisements containing the current the routing tables from its (actually only the costs are needed) and uses that information to updates its own routing table. For our code, we will have each node **n** grab the required information from its neighbors.

Each node **n** loops over all of its neighbors (contained in the variable **neighbors{n}**).

For each neighbor, which is referred to by the variable **linked** and which it can reach with the value **cost**, the node will the loop over all nodes **k** in the network.

For each node **k**, node **n** computes the cost to reach node **k** through that node **linked**, which is th cost to get to the node **linked** and the cost to get from **linked** to **k**. If it can reach **k** through **linked** cost than the cost to node **k** stored in its current routing table, **RT{n}(k,2)**, then it will update the **k**tl routing table to indicate that packets destined for **k** should be forwarded to the node **linked**, and u cost to get to node **k**.

After node n has updated its routing table based on information gathered from its neighbors, it adv routing table using the function **sendAdvertisement**.

Your task is to revise the code to update the routing table. Basically, you need to complete the las each iteration where node **n** compares the cost of its current route to node **k**, **RT{n}(k,2)**, with the r its neighbor **linked**. If the route through **linked** is better, node **n** will update the values of both **RT{ RT{n}(k,2)**.

Please, revise the code between the lines

```
% % % % Revise the following code   % % % %
```

and

```
% % % % Do not change the code below % % % %
```

Do not change other parts of the code.

Your Solution

Reset MATLAB Documentation (https://www.mathworks.

```
                    %disp(B);
                    %disp(A+C);
                    %disp(linked);
                    %save values since other neighbors may have better
                    RT{n}(k,2) = A + cost;
                    RT{n}(k,1) = linked;
                    %savedCosts = [ savedCosts A+C ];
                    %savedNeighbors = [ savedNeighbors linked ];
```

Assessment Tests: Passed

✓ Is the problem unchanged?

✓ Is the routing table correct?

Output

```
ans =

     0     0
     2     2
     4     4
     4     1
     4     2
     3    10


ans =

     0     0
     2     2
     4     3
     4     1
     4     2
     4     4


ans =
    0     0
    2     2
    4     3
    4     1
    4     2
    4     4
```

© 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.

Download on the App Store

ANDROID APP ON Google play