Homework 2 Report

Rafat Khan

September 22, 2021

1 Introduction

The purpose of this homework is to implement a link-state routing protocol in Erlang and learn about directional links between routers. The protocol allows to build and check routing tables to determine which gateway is best to forward a message using dijkstra algorithm. Implementing the routing protocol, configuring routers and message passing between different nodes are the main aspects of this seminar. Every node in the network is named as a city for conducting network tests easily.

2 Main problems and solutions

This assignment was way more complex than the previous one. More knowledge in Erlang was necessary. Understanding the lists and tuples was the first challenge. Apart from the programming difficulties,understanding the relationships among the components of the network protocol, in particular how to use the sorted list in order to update the routing tables was another tricky part.

Another difficulty was to understand the relationships among the various components of the network protocol, in particular how to use the sorted list in order to update the routing tables. The dijkstra algorithm and the iteration function, which was the hardest part in my opinion. Understanding the whole process took a lot of time and effort. This function find the list of reachable nodes and sort the list to find the shortest way. The final stage was the construction of the router module. The given code merges all the previous steps and creates routers that are able to communicate. In order to route a message to a node, the router constructs the routing table and finds the proper and shortest gateway to send the message.

3 Evaluation

Test 1: In this test case, I used seven routers named as different cities of Sweden. I connect the routers and send message from one to another.

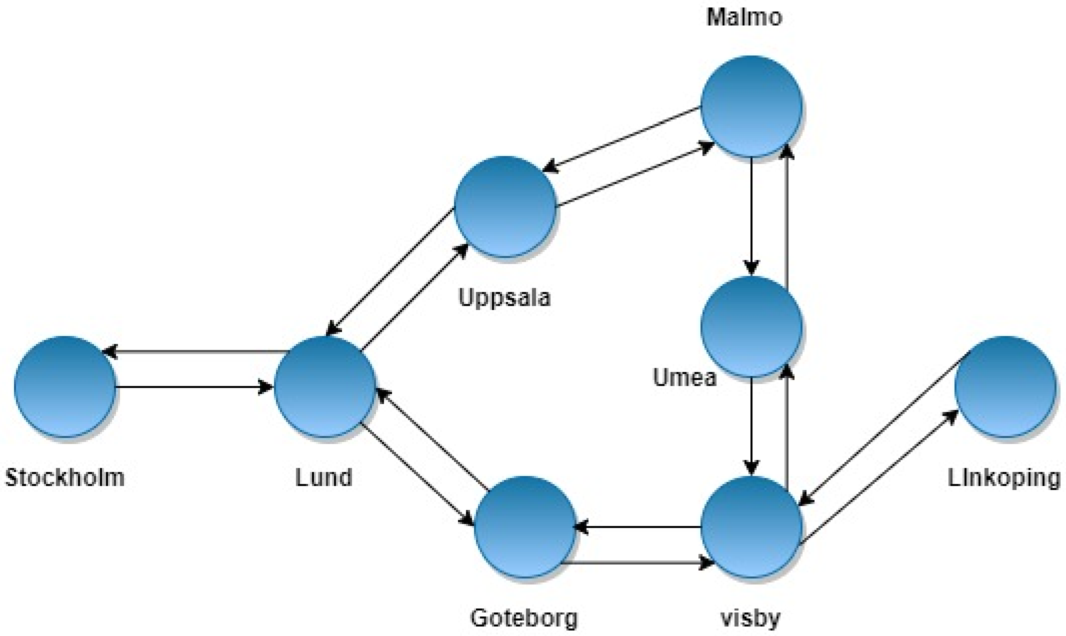
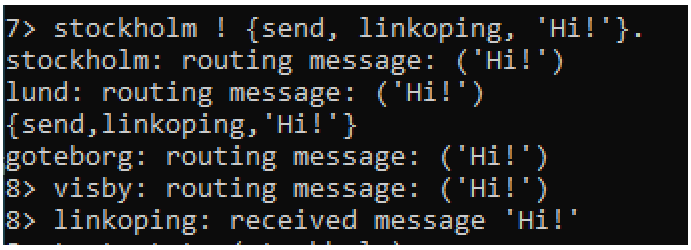


Figure 1: The structure of test 1.

At first, I have configured these seven routers and connect them to each other. The longest distance was between Stockholm and Linkoping. To send a message between these two routers, minimum three hopes is required (lund,goteborg,visby). There is another path, which requires 4 hops (lund, uppsala, malmo, umea). As we used dijkstra algorithm to find the path, it should select the shortest path. For testing, I have sent a message to linkoping from stockholm. stockholm ! send, linkoping, 'Hi!'.

As a result I got the following response:
stockholm: routing message: ('Hi!')
lund: routing message: ('Hi!')
goteborg: routing message: ('Hi!')
visby: routing message: ('Hi!')
linkoping: received message 'Hi!'

These response shows that the message is delivered using the shortest path.



```
7> stockholm ! {send, linkoping, 'Hi!'}.
stockholm: routing message: ('Hi!')
lund: routing message: ('Hi!')
{send,linkoping,'Hi!'}
goteborg: routing message: ('Hi!')
8> visby: routing message: ('Hi!')
8> linkoping: received message 'Hi!'
```

Figure 2: Routing a message to linkoping from stockholm in Test 1.

Test 2: In this test case, I used same routers, but I have broke the communication link between lund and gotheborg.

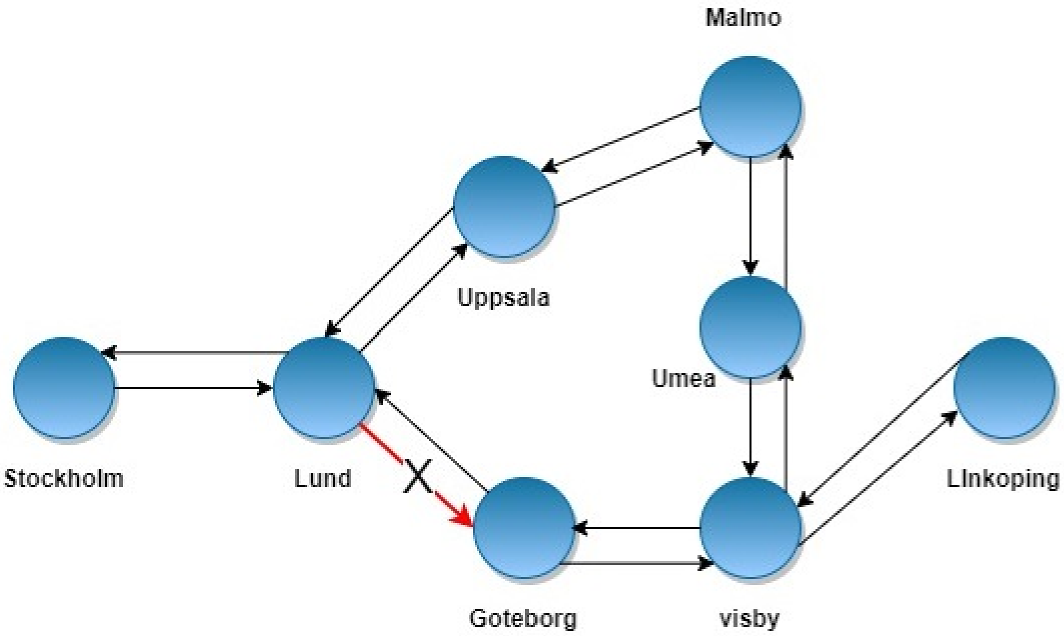
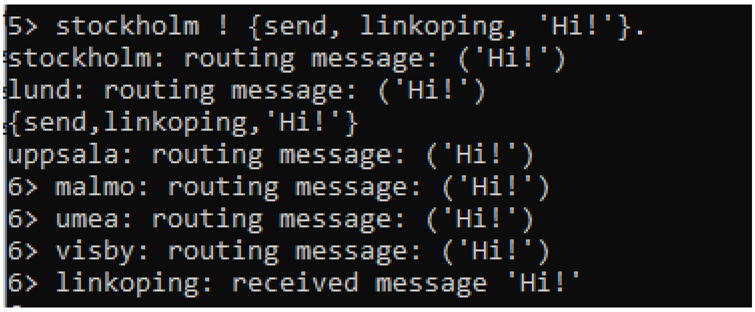


Figure 3: The structure of test 2.

So, if we want to send a message to linkoping from stockholm, routy will have to select the longer path (lund, uppsala, malmo, umea). After sending a test message, this is the response that I got:
stockholm: routing message: ('Hi!')

```
lund: routing message: ('Hi!')
uppsala: routing message: ('Hi!')
malmo: routing message: ('Hi!')
umea: routing message: ('Hi!')
visby: routing message: ('Hi!')
linkoping: received message 'Hi!'
```



```
5> stockholm ! {send, linkoping, 'Hi!'}.
stockholm: routing message: ('Hi!')
lund: routing message: ('Hi!')
{send,linkoping,'Hi!'}
uppsala: routing message: ('Hi!')
6> malmo: routing message: ('Hi!')
6> umea: routing message: ('Hi!')
6> visby: routing message: ('Hi!')
6> linkoping: received message 'Hi!'
```

Figure 4: Routing a message to linkoping from stockholm in Test 2.

4 Conclusions

In this assignment, I learned the basic structure of a routing protocol. In addition, I learned to know about lists and how to handle with tuples in Erlang. Now , I'm able to describe the main concept of the link-state routing protocol, the Djikstra algorithm and a general approach for the routing procedure.