

Routy - a routing network

Chrysoula Dikonimaki

September 22, 2021

1 Introduction

This report describes Routy, a small routing network implemented in Erlang. My main goal was to implement a link-state routing protocol.

2 Main challenges

The main challenge was to understand the lists module. It was very useful because Erlang doesn't have for loop statement or while statement so I used the module's functions to avoid recursion. Understanding Dijkstra in depth was also an important part of the assignment.

3 Experiment: The country

At the first experiment, I will create create 3 routers in Sweden: Stockholm, Lund and Gothenburg. I will connect Stockholm with Lund and Lund with Gothenburg (Stockholm- \rightarrow Lund- \rightarrow Gothenburg) but I won't connect Lund with Stockholm and Gothenburg with Lund so the graph can be considered as a directional graph. In Figure 1 we can see the graph. After updating all the tables, I will try to send a message from Stockholm to Gothenburg through Lund. It is clear that it won't work because Lund is not connected with Stockholm which means that Lund's set of interfaces doesn't contain Stockholm, so Lund will never share with Stockholm that it is connected with Gothenburg. We can see the statuses in Figure 2.

From this we can conclude that the only way to send a message from Stockholm to Gothenburg is to make the graph bidirectional (Figure 3) or have a circle (connect Gothenburg to Stockholm) (Figure 4). At the first



Figure 1: Directional graph

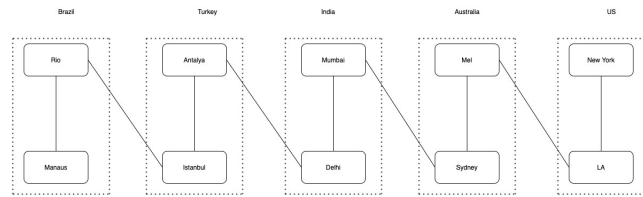


Figure 5: The world

```
(brazil@130.229.190.152)16> manaus ! {send, newyork, 'hello from manaus'}.
manaus: routing message ('hello from manaus'){send,newyork,'hello from manaus'}
rio: routing message ('hello from manaus')(brazil@130.229.190.152)17> |
```

Figure 6: Manaus sends message to New York, Manaus routes the message to Rio

```
the gateways are : {rio,antalya}
update
(istanbul) node: routing message: ('hello from manaus') from: (manaus)
Pid is : {antalya,'turkey@130.229.172.253'}
(antalya) node: routing message: ('hello from manaus') from: (manaus)
Pid is : {delhi,'india@130.229.147.236'}
(turkey@130.229.172.253)13> |
```

Figure 7: Turkey routers are routing the message

```
{newyork,manaus},{manaus,antalya},{la,manaus},{melbourne,mumbai},{rio,antalya},{delhi,antalya},{istanbul,antalya},{sydney,mumbai},{mumbai,manaus},{antalya,antalya}}{delhi: routing message ('hello from manaus')}
(mumbai): routing message ('hello from manaus')
```

Figure 8: India routers are routing the message

```
(mel) node: routing message ('hello from manaus') from: (manaus)
Pid is : {delhi,'india@130.229.147.236'}
(melbourne): routing message ('hello from manaus') from: (manaus)
Pid is : {la,'usa@130.229.191.122'}
```

Figure 9: Australia routers are routing the message

```
(americ@130.229.179.191)12> la: routing message ('hello from manaus')(americ@130.229.179.191)12> newyork: received message 'hello from manaus'
```

Figure 10: New York receives the message

5 Conclusions

According to the experiment described in Section 3, we need to have a bidirectional graph in order to be sure that our Routy protocol will work as expected. According to the experiment described in 4 connecting multiple computers with Erlang and sending messages is not that hard.