Distributed Systems
Report 2: Routy

Yerrapragada Akhil

September 19, 2019

1 Introduction

This seminar is intended to explain the process of creating routing network using Erlang and describe the functionality of it. The key technical aspects implemented in the routing network are: A map, Dijkstra algorithm, Interfaces, History and the Router.

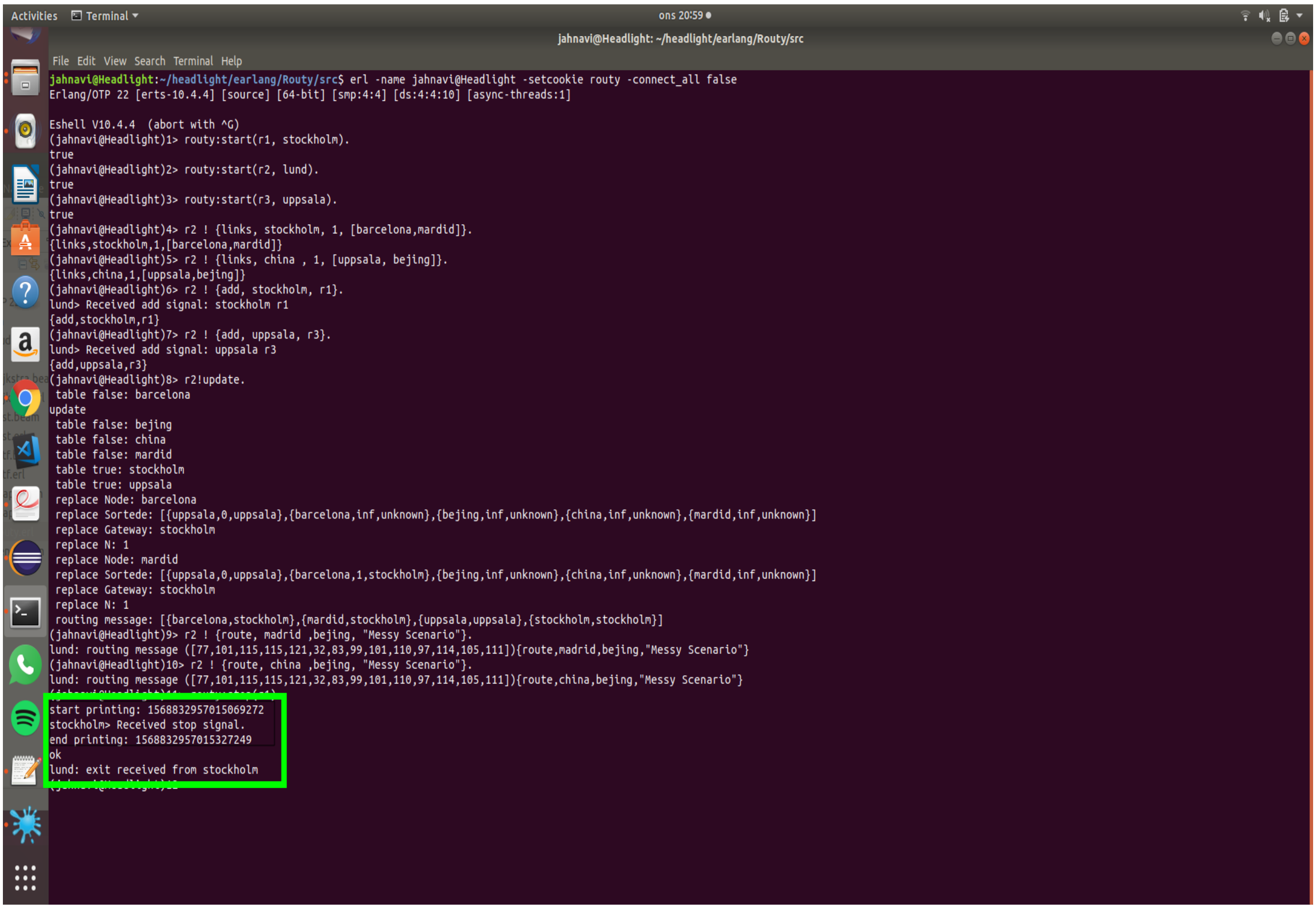
Routers communicate with each other, distributing information that enables them to select routes between different nodes on a network. A routing protocol specifies how the routers function and uses Dijkstra algorithm to perform the routing. In order to represent the connection between the nodes we use a Map and to route message to a node, the router uses a routing protocol to find the gateway and then sends the message. We also have Interfaces which are used to broadcast the messages to interface processes. Old messages are replaced with latest details using History Table.

2 Main problems and solutions

Framing the flow between multiple modules and description to code conversion made it more time consuming. Deriving routing table from assumed map made it difficult in picturizing a routing table. Understanding the gateway mechanism, especially, with unspecified paths took more toil. Identifying list and dict methods such as fold, members, etc. which were used in the code were implemented after a lot of research on list modules using erlang. Redirecting, handling node unavailability scenario, connectivity between components added complexity to the functionality. Overall, there was a lot of research involved in completion of this assignment.

3 Evaluation

1) The below screenshot depicts the time between disabling a network and delivery of 'DOWN' message to other nodes. The "Start Printing" displays the time when the node was shut and "End Printing" displays the time when the exit message was displayed. The time difference explains the time taken to receive the exit message.

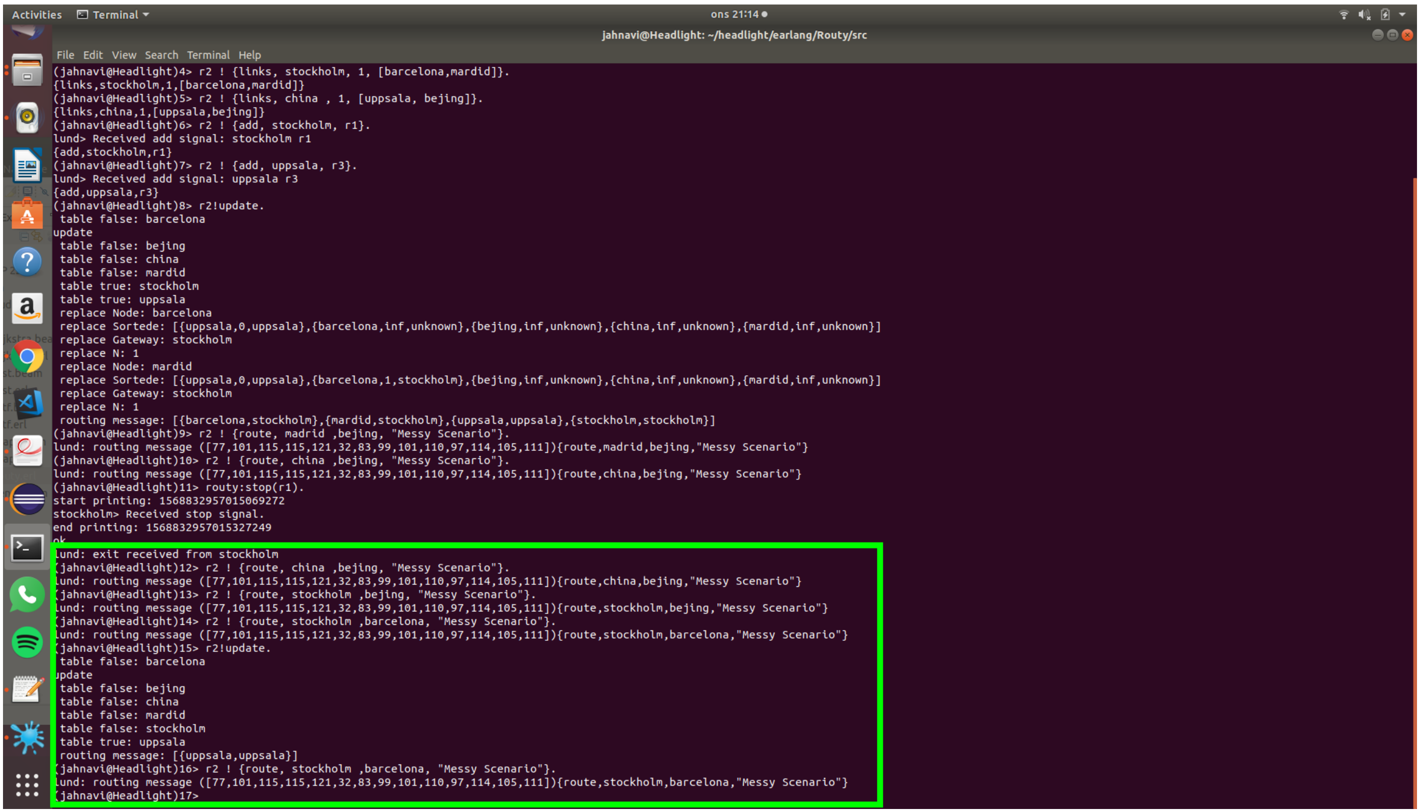


Activities
Terminal
ons 20:59
jahnavi@Headlight: ~/headlight/earlang/Routy/src
File Edit View Search Terminal Help

```
jahnavi@Headlight:~/headlight/earlang/Routy/src$ erl -name jahnavi@Headlight -setcookie routy -connect_all false
Erlang/OTP 22 [erts-10.4.4] [source] [64-bit] [smp:4:4] [ds:4:4:10] [async-threads:1]

Eshell V10.4.4  (abort with ^G)
(jahnavi@Headlight)1> routy:start(r1, stockholm).
true
(jahnavi@Headlight)2> routy:start(r2, lund).
true
(jahnavi@Headlight)3> routy:start(r3, uppsala).
true
(jahnavi@Headlight)4> r2 ! {links, stockholm, 1, [barcelona,mardid]}.
{links,stockholm,1,[barcelona,mardid]}
(jahnavi@Headlight)5> r2 ! {links, china , 1, [uppsala, bejing]}.
{links,china,1,[uppsala,bejing]}
(jahnavi@Headlight)6> r2 ! {add, stockholm, r1}.
lund> Received add signal: stockholm r1
{add,stockholm,r1}
(jahnavi@Headlight)7> r2 ! {add, uppsala, r3}.
lund> Received add signal: uppsala r3
{add,uppsala,r3}
(jahnavi@Headlight)8> r2!update.
 table false: barcelona
update
 table false: bejing
 table false: china
 table false: mardid
 table true: stockholm
 table true: uppsala
 replace Node: barcelona
 replace Sortede: [{uppsala,0,uppsala},{barcelona,inf,unknown},{bejing,inf,unknown},{china,inf,unknown},{mardid,inf,unknown}]
 replace Gateway: stockholm
 replace N: 1
 replace Node: mardid
 replace Sortede: [{uppsala,0,uppsala},{barcelona,1,stockholm},{bejing,inf,unknown},{china,inf,unknown},{mardid,inf,unknown}]
 replace Gateway: stockholm
 replace N: 1
 routing message: [{barcelona,stockholm},{mardid,stockholm},{uppsala,uppsala},{stockholm,stockholm}]
(jahnavi@Headlight)9> r2 ! {route, madrid ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,madrid,bejing,"Messy Scenario"}
(jahnavi@Headlight)10> r2 ! {route, china ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,china,bejing,"Messy Scenario"}
(jahnavi@Headlight)11> routy:stop(r1).
start printing: 1568832957015069272
stockholm> Received stop signal.
end printing: 1568832957015327249
ok
lund: exit received from stockholm
(jahnavi@Headlight)12>
```

2) The below screenshot portrays the scenario when one of the nodes in the network is made unavailable.


Activities
Terminal
ons 21:14
jahnavi@Headlight: ~/headlight/earlang/Routy/src
File Edit View Search Terminal Help

```
(jahnavi@Headlight)4> r2 ! {links, stockholm, 1, [barcelona,mardid]}.
{links,stockholm,1,[barcelona,mardid]}
(jahnavi@Headlight)5> r2 ! {links, china , 1, [uppsala, bejing]}.
{links,china,1,[uppsala,bejing]}
(jahnavi@Headlight)6> r2 ! {add, stockholm, r1}.
lund> Received add signal: stockholm r1
{add,stockholm,r1}
(jahnavi@Headlight)7> r2 ! {add, uppsala, r3}.
lund> Received add signal: uppsala r3
{add,uppsala,r3}
(jahnavi@Headlight)8> r2!update.
 table false: barcelona
update
 table false: bejing
 table false: china
 table false: mardid
 table true: stockholm
 table true: uppsala
 replace Node: barcelona
 replace Sortede: [{uppsala,0,uppsala},{barcelona,inf,unknown},{bejing,inf,unknown},{china,inf,unknown},{mardid,inf,unknown}]
 replace Gateway: stockholm
 replace N: 1
 replace Node: mardid
 replace Sortede: [{uppsala,0,uppsala},{barcelona,1,stockholm},{bejing,inf,unknown},{china,inf,unknown},{mardid,inf,unknown}]
 replace Gateway: stockholm
 replace N: 1
 routing message: [{barcelona,stockholm},{mardid,stockholm},{uppsala,uppsala},{stockholm,stockholm}]
(jahnavi@Headlight)9> r2 ! {route, madrid ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,madrid,bejing,"Messy Scenario"}
(jahnavi@Headlight)10> r2 ! {route, china ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,china,bejing,"Messy Scenario"}
(jahnavi@Headlight)11> routy:stop(r1).
start printing: 1568832957015069272
stockholm> Received stop signal.
end printing: 1568832957015327249
ok
lund: exit received from stockholm
(jahnavi@Headlight)12> r2 ! {route, china ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,china,bejing,"Messy Scenario"}
(jahnavi@Headlight)13> r2 ! {route, stockholm ,bejing, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,stockholm,bejing,"Messy Scenario"}
(jahnavi@Headlight)14> r2 ! {route, stockholm ,barcelona, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,stockholm,barcelona,"Messy Scenario"}
(jahnavi@Headlight)15> r2!update.
 table false: barcelona
update
 table false: bejing
 table false: china
 table false: mardid
 table false: stockholm
 table true: uppsala
 routing message: [{uppsala,uppsala}]
(jahnavi@Headlight)16> r2 ! {route, stockholm ,barcelona, "Messy Scenario"}.
lund: routing message ([77,101,115,115,121,32,83,99,101,110,97,114,105,111]){route,stockholm,barcelona,"Messy Scenario"}
(jahnavi@Headlight)17>
```

Conclusion

Overall, there has been a significant learning in understanding the routing mechanism. Especially creating a network, establishing connectivity between the network components and understanding node failure scenario. Dijkstra algorithm provided a detailed understanding on routing functionality.