

# Report 2: Routy, a small routing protocol

Pierre Fleitz

September 23, 2015

## 1 Introduction

During this seminar the main goal is to implement, in Erlang of course, a routing protocol based on link-state routing protocol used in the OSPF.

In our implementation, to have router with a complete map of the network we proceed that way : It's important that a router can broadcast link-state messages to all their gateways so it can tell to its neighbor what are all its gateways. Then these neighbor routers process the link-state message (save the information) and broadcast it to all their own gateways. Then each link-state message of a router goes through the whole network, and all the routers know what are the gateways of the router who has initially send the link-state message. We can understand that if each router does this, all routers will have a complete map of the network.

In this seminar we implemented the Dijkstra algorithm, which is an algorithm used to find shortest paths in the network. We needed it so each router can compute a routing table to know where to route a message that has to go to a destination and all of that through the shortest path possible.

## 2 Main problems and solutions

I think the main problem, but also the most important thing, about this seminar is to understand how procedures from lists module work, but also when to use them. At the beginning of this seminar, I had almost no idea how to use `lists:keydelete` for example, but for this seminar we didn't have an helpful code skeleton, then the only solution was to go to [www.erlang.org](http://www.erlang.org) and to read all the man page of lists. Once you understand how and when to use them, the seminar starts to be a little bit more accessible.

About the implementation of the Dijkstra algorithm. Even if it has been for me the most interesting point of the seminar, I used to have a graph theory course last year, I encountered a lot of difficulties about how to solve things. I was not able to understand how to code the `iterate/3` function until I really understood what was `list:foldl/3` and how to use it.

After that everything was quite good.

As it is said in the subject for the interfaces module it has been pretty easy to implement it because we studied the lists module before and at this point of the seminar it was easier to understand how to use them properly.

For the router module, the code is given and the subject is pretty clear.

### 3 Evaluation

When I wanted to test my router after implementing everything, I tried to send a message from 1 router to another one to manually connect them and here I had a problem. This is the terminal output I had :

```
(sweden@192.168.1.13)1> c(routy).
routy.erl:62: Warning: variable 'From' is unused
{ok,routy}
(sweden@192.168.1.13)2> c(dijkstra).
dijkstra.erl:57: Warning: variable 'Dest' is unused
{ok,dijkstra}
(sweden@192.168.1.13)3> c(hist).
{ok,hist}
(sweden@192.168.1.13)4> c(map).
{ok,map}
(sweden@192.168.1.13)5> c(intf).
{ok,intf}
(sweden@192.168.1.13)6> routy:start(r1,stockholm).
true
(sweden@192.168.1.13)7> routy:start(r2,lund).
true
(sweden@192.168.1.13)8> lund ! {add,stockholm,{r1,'sweden@192
.160.1.13'}}.
** exception error: bad argument
    in operator !/2
        called as lund ! {add,stockholm,{r1,'sweden@192
.160.1.13'}}}
(sweden@192.168.1.13)9>
```

I literally spent hours to try to understand where this came from. I can understand the error message. Basically it tells me that `add,stockholm,r1,'sweden@192.160.1.13'` is a wrong argument, so it might be `"add"` or `"stockholm"` or `"r1,'sweden@192.160.1.13'"`. It is the same type of error that if you try to do `1/0` in your erlang terminal. But still, at the end I haven't been able to find why I have this message error, and this is probably the first thing I'll ask during the seminar because it has been really frustrating.

### 4 Conclusions

This seminar is really interesting. It's a long seminar and it asks to be really involved and focus on what you do but this is a really good way to practice your erlang, but also to understand how a routing protocol based

on link-state works (wich is pretty important since OSPF is the most used routing protocol for Internet routers). It's a good way to understand how the dijkstra algorithm works and how to implement it in Erlang. I have to admit that I am really frustated not to understand where the error I face comes from because I am not able to do some tests and I really spent a lot of time on this seminar because I wanted to be able to "play" with it at the end. If I have understood well the implementation I think that the conclusion we wanted to reach with this seminar is that : fault-tolerancy (when the router is working) is supposed to be handled. Indeed if somes routers crash, messages are supposed to be successfully routed (that's one of the thing I wanted to test). And finally I think that the fact that each router builds a routing table for the entire network can be a problem, indeed if we have a little network it's okay, but if we have a huge one, that could be problematic. I think it would be interesting to calculate how huge the network could be with this implementation without facing delay-related problems or else.