Report : Routy - A small routing protocol

Vasileios Giannokostas , vasgia@kth.se

September 27, 2014

1 Introduction

*T*he aim of this seminar is to implement a link-state routing protocol in Erlang. This protocol is performed by every switching node in our network. The basic concept is that every node creates a map of the connectivity of the network showing the connections among the nodes. The link-state routing protocol widely used for OSPF protocol for Internet routers. After the end of the seminar we will be able to:

- Understand the structure and functionality of the link-state routing protocol.
- Describe the concept of Djikstra algorithm.
- Understand how a consistent connections of routers is mainted.
- Do some practice in Erlang.

*W*e will depict a connection among several countries that communicate each other. We will define every router with a unique name of a city and we will try to connect them in the same local network. The routers connect with direct links but also they communicate with indirect links (paths that connect routers and they are not directly connected).

Our routing process must construct a table and determine which is the proper gateway for a node to communicate with the others. The system rejects messages for which there is no a logical path to reach the receiver.

2 Main problems and solutions

During the implementation of the routing protocol several issues occur such as :

- How can I illustrate a map of nodes and links?

- How can I decide which is the shortest path to reach a node?
- How the system is updated?

For every question there is a solution that satisfies the requirements of our system.

First of all we constructed functions to represent every direct link of a node. We implemented the map module that constructs a map which illustrates the nodes of our network and the direct links of every node. So , the map is a list that consists of cities(Nodes) and the directly connected cities with the Nodes. The functions of this module are useful because they give us the opportunity to keep the map updated , we can change the map or retrieve data from it.

The second step , the construction of Dijkstra algorithm ,was the most difficult for me. I faced several problems to understand and construct the algorithm but I perceived that this algorithm is the heart of our protocol. The Dijkstra algorithm defines the shortest path to reach a Node by defining the proper gateway for every Node. The algorithm knows the Map (the direct links between the cities) , the gateways and the cost of a route (how many hops is required to reach a Node). Considering this inputs the algorithm define the paths to the routing table and "creates" new paths for every possible destination. Of course , the algorithm is clever so updates the routing table only if the new path is shorter that the existing one. The most valuable output of the Dijkstra is the proper gateway suitable to a route message to a node.

Another crucial step of our implementation was the construction of history. The history is useful because we don't want to receive old messages and also avoid infinite loops of a received message. The history keep tracks for the sending messages of every Node by using a counter , so it knows when a message is old or new.

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 gateway to send the message.

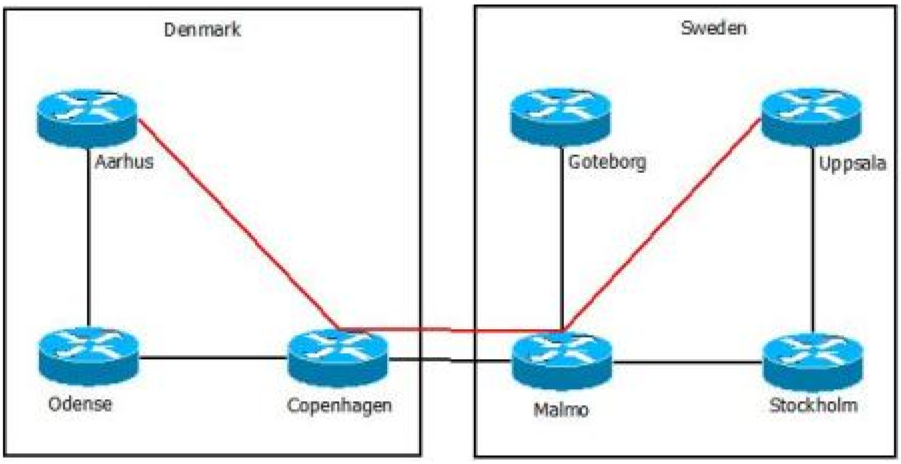
The basic functions of a router are:

- Creates new routers.
- Add direct connections between two Nodes.
- Constructs routing table , updates and maintains it.

- Sends messages to another Node.
- Evaluates if a message is old or new.

3 Evaluation

*I*t's now the time to implement the actual routing. I runned the routing protocol in the same machine using two different command prompts. I named the first Sweden and the second Denmark , I created several routers for every country that every of them represents a city. The figure below represents the initial connection(black lines).



A topology of the Nodes.

I represent two scenarios running my routing protocol:

Scenario 1: I will send a message from Uppsala to Aarhus. There is no direct link between these two nodes so the routers route the message to the next. The message from Uppsala routes through Stockholm, Malmo, Copenhagen, Odense and finally received from Aarhus.

Scenario 2: In this example I add two new links(red lines). The first between Uppsala-Malmo and the second between Copenhagen-Aarhus. Now , the path is different from the previous example (red path , the red connection between Malmo - Copenhagen is the same as previous). The message routes though Malmo and Copenhagen and arrives to Aarhus. The protocol choose the shortest path , assuring us that the Dijkstra algorithm works properly and the routing table is updated.

4 Conclusions

This seminar taught me the basic logic of a routing protocol. Moreover , the difficulty to complete this assignment boosted me to try hard and learn more than I expected. 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.