

Routy: a small routing protocol

Zainab Alsaadi

September 22, 2021

1 Introduction

This report briefly demonstrates the work of implementing of link-state protocol used for a small network of routers. It consists of 5 modules; map, dijkstra, interface, history and routy. All modules are used by the final one; routy, which implements the routing process.

The aim of this assignment is to understand the structure of a link-state routing-protocol, be able to describe how a consistent view is maintained and reflect to network-related failures.

2 Main problems and solutions

The assignment was to complete functions that handle the routing table, set of interfaces and maps. The encountered problems for implementing these functions were mostly caused by the lack of test cases, which makes it harder to know whether the implementation was fully or partly correct before moving on to implementing next functions and modules.

Furthermore, The assignment was focusing a bit more on the implementation of the methods than explaining the process in more details, which redirects the focus on learning how to use Erlang rather than approaching the listed learning outcomes.

Otherwise, a good thing was that the assignment was decomposed into smaller sub-problems. That makes it easy to implement and follow step-by-step until being able to grasp the whole picture of how this process works.

3 Evaluation

Dijkstra's algorithm is used to find the shortest route between the nodes by calculating the routing table. The input to this algorithm is a map and a list of gateways to which we have directed access.

The map is used to represent a list of entries where each entry consists of a city. It describes how all nodes and gateways are connected. The implementation of a map consists of nodes with directed outgoing links.

This was the first module to be implemented and it was properly tested before moving on.

The interface is used to handle a set of interfaces for directly connected routers. It is described by a symbolic name, a process reference and a process id.

The history module is implemented to avoid cyclic paths so that messages are not resent to the same destination forever. This is solved by setting a counter on each message and decrement it on each hop.

The main module routy was given. It gives a view of the network and constructs optimal routing tables in order to be able to route messages through a network of connected nodes.

4 Conclusions

The presented problem was very time consuming, but interesting and informative. I have learnt about routing messages as well as developed my skills in using Erlang.