Report 2: Routy

Peter Hamberg

September 14, 2011

1 Introduction

Construction of a router system based on the dijkstra algorithm.

This is an example of a distributed system that has to communicate via message passing between both processes and between computers.

2 Main problems and solutions

The core of this assignment was to understand and implement the dijkstra algorithm in a routing network.

In order to more effectively grasp the code examples provided in the lab manual, i set up some basic unit tests. By simply including the header

```
-include_lib("eunit/include/eunit.hrl").
```

an exported function test/0 will be added to the module that when called in turn calls all functions with names matching the pattern /^\w_test_?$/. An example test could then be added like so

```
dijkstra_update_properly_replaces_route_if_shorter() ->
    [{london,1,stockholm}, {berlin, 2, paris}] =
        dijkstra:update( london, 1, stockholm,
            [{berlin, 2, paris}, {london, 3, paris}]).
```

Since this made away with most of the manual testing required to verify the correctness of the code, my hands were saved from a lot of repetative typing in the console.

Another nuiscance was that setting up a proper routing network is very chatty. I solved this by using escript.

```
#!/usr/bin/env escript
%%! -name us -sname us -setcookie routy -connect_all false

main(_) ->
   make:all(),
   routy:start(washington),
   [...]
   washington ! {add, boston, {boston, 'us@AOEU'}},
   [...]
   send().

send() ->
   case io:fread("command:", "~a ~a ~s") of
      {ok, [Target, b, _]} ->
         Target ! broadcast;
      {ok, [washington, To, Msg]} ->
         washington ! {send, To, Msg };
      [...]
   end,send().
```

To make the scripts run on my windows machine I use the mingw32 shell that came with my git install.

This type of script can of course be modified to run unit tests or whatever one prefers. After coming to rely on these solely for building and testing, I finally made away with eclipse and started using notepad++ instead.

3 Evaluation

The routing system set up is rather robust and seems to handle even violent node takedowns. Still a lot of manual guidance is needed to get the system up and running. It is not a fully autonomus system, despite all the automated scripting added.

4 Conclusions

I was rather happy to be able to unit test my code. The abundant typing in eclipse's rather awful console that lacked support for copy&paste and command history was starting to take a toll on my wrists.

Code-wise this lab was very interesting to set up. It is always exciting to create larger inter-dependent structures that actually work seamlessly together. I suppose the seminar will be the final arbiter of that last item.