Advanced Communications 4

Assessed Exercise 2010

Adrián Gómez Llorente

0904327G

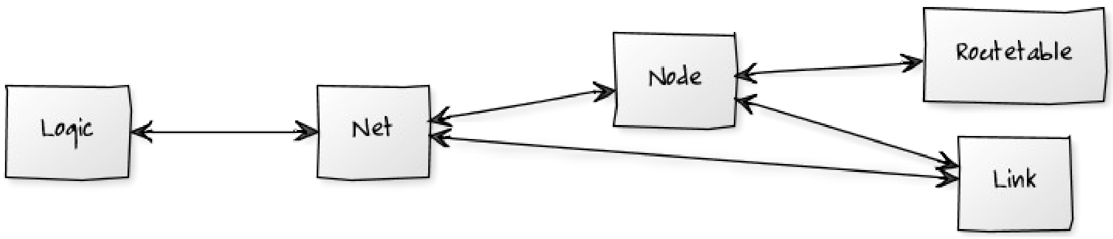
Introduction

In this document is going to be explained a brief about the net simulator and you will be able to find next chapters:

- Overview of the design
- User manual
- Example with normal convergence
- Example with normal convergence and link changes
- Example with slow convergence
- Status report of the application

Overview of the design

In the next diagram we can see an overview of the application design and how each object works with others.



- **<u>Logic:</u>** Is the main object because is the one that interacts with the user and controls the whole application.
- **<u>Net:</u>** Is the object that represent the network and offers all the operations needed to work with it.
- **<u>Link:</u>** Represents a link between two nodes and the distance is also included
- **<u>Node:</u>** Represents a node of the network
- **<u>Routetable:</u>** Represents the routetable of a given node containing all the information of all the possible connections, it also gives operations for working with it.

User manual

Requirements

The application has been developed using java because this programming language offer the possibility to be executed in almost every operating system so only Java Runtime Environment (JDK) is needed to compile and execute this application, to get it working in a fresh Windows XP SP3 installation we need to follow next steps:

1. Downloading Java from Sun website (http://java.sun.com/javase/downloads/widget/jdk6.jsp)
2. Once downloaded we only need to execute the application and install it as a normal Windows application, just click on next until the end.
3. Now we need to change the Environment variable PATH, to do this we need to follow next steps
    1. Right click on MI PC > Properties > Advanced Options > Environment variables
    2. We find the variable Path and we click on edit
    3. In the end of the value we have to add semicolon and the path in where we have installed the JDK, by default is: ;C:\Program Files\Java\jdk1.6.0_18\bin

Installing the application

To install the application we have to follow next steps:

1. Double click in NetSimulator
2. Sometimes windows ask if you want to continue with the installation for security reasons we click in Yes.
3. A wizard will appear and we select the destination folder, default is program files.
4. That's all!

Compiling the application

The application only include the sources so we need to compile them to get an executable file, to do this we click in the archive called **build**

Executing the application

The application is now ready to run so if we want to execute it you only need to click on **execute** and a shell with the main menu will appear

Running the application

To run the application you only have to go to the folder where you have installed it and double click in “execute”. A shell will appear with the main menu of the application. Next we are going to explain different options of the main menu:

- **Parse net:** This option will offer the possibility to parse a network from a given file, is the first action we have to do to start using the application. We only have to type the path of the file and the application will parse it. Two example networks are included so if we want to use them we only have to type **net1** or **net2**

- **<u>Simulate net:</u>** This option run a simulation in the network parsed. To use it we will have to type desired iterations and the application will show us routetables of each iteration
- **<u>Change value:</u>** This offer the possibility to change the cost of current link. To change it we only have to type "**link node1;node2;newcost**" and the application will change the cost and will be prepared to other simulation to check how the new cost will affect to routetables
- **<u>Show net:</u>** Show the structure of the current net
- **<u>Show routetables:</u>** Show current routetables of all nodes
- **<u>Exit:</u>** Close the application

<u>Creating a new network</u>

To create a new network we have to follow next steps:

- First we have to include all the nodes with this structure: node <name>
- Second we have to include all the links with this structure: link <node1>;<node2>;<cost>

Next an example of a simple network is given:

```
node n1
node n2
node n3
link n1;n2;10
link n2;n3;20
link n3;n1;5
```

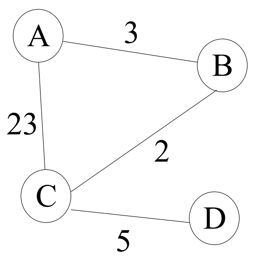
In the previous example we have created 3 nodes (n1,n2,n3) and 3 links:

- Link between n1 and n2 with cost 10
- Link between n2 and n3 with cost 20
- Link between n3 and n1 with cost 5

Example with normal convergence

In this chapter we are going to test the application with a complete example using normal convergence. I have decided to use one example found in Wikipedia.

Next we can see a diagram of the network:



The code needed to represent this net in our application is shown next

```
node A
node B
node C
node D
link a;b;3
link a;c;23
link b;c;2
link c;d;5
```

So we load this network in the application. We know that routetables of this network are stables in 4 iterations so we are going to setup 4 iterations for the simulation.

Iteration 1

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	23	A	C	5	C
C	23	C	C	2	C	B	2	B			
						D	5	D			

As can be seen in previous data obtained from the application this corresponds to the information that have all the nodes about the neighbours they are connected to. Is the information which is going to be used by the algorithm to calculate next values.

Iteration 2

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	23	A	A	28	C
B	25	C	A	25	C	A	5	B	B	7	C
C	5	B	C	26	A	B	26	A	C	5	C
C	23	C	C	2	C	B	2	B			
D	28	C	D	7	C	D	5	D			

In the previous tada we can see that the nodes has changed all data obtained in the first iteration, I am going to explain some of them to know how this algorithm is working:

Node B has send all data to their neighbours so for example Node A has received this data from Node B (C,2,C). This means that Node B reaches Node C with cost 2 so Node A, who knows that is capable of getting to Node B in 3, knows now that is capable of getting to Node C with cost 5 so it updates its routetable with the information (C,5,B).

And this occurs for every node of the net until getting an stable

Final Iteration

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	23	A	A	10	C
B	25	C	A	7	C	A	5	B	B	7	C
C	5	B	C	8	A	A	15	D	C	5	C
C	23	C	C	2	C	B	26	A			
D	10	B	D	13	A	B	2	B			
D	28	C	D	7	C	B	12	D			
						D	33	A			
						D	9	B			
						D	5	D			

I have marked the minimum cost for each node to get to other nodes.

Example with normal convergence and link changes

Now I am going to change one link cost, for example link between A and C will have a cost of 1 instead of 23 and we will repeat the simulation to see how it gets to a result.

Iteration 0

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	1	A	A	10	C
B	25	C	A	7	C	A	5	B	B	7	C
C	5	B	C	8	A	A	15	D	C	5	C
C	1	C	C	2	C	B	26	A			
D	10	B	D	13	A	B	2	B			
D	28	C	D	7	C	B	12	D			
						D	33	A			
						D	9	B			
						D	5	D			

In the previous routetables the change in the cost is marked.

Iteration 1

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	1	A	A	6	C
B	3	C	A	3	C	A	5	B	B	7	C
C	5	B	C	4	A	A	11	D	C	5	C
C	1	C	C	2	C	B	4	A			
D	10	B	D	13	A	B	2	B			
D	6	C	D	7	C	B	12	D			
						D	11	A			
						D	9	B			
						D	5	D			

In the previous routetable we have seen the results of the first iteration in which Node A and Node C have sent new information to their neighbours this is why in this iteration Node B knows that it can reach to Node C with cost 3 through Node A. This will continue until stability

Final iteration

NODE A			NODE B			NODE C			NODE D		
Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out	Dest	Cost	Out
B	3	B	A	3	A	A	1	A	A	6	C
B	3	C	A	3	C	A	5	B	B	7	C
C	5	B	C	4	A	A	11	D	C	5	C
C	1	C	C	2	C	B	4	A			
D	10	B	D	9	A	B	2	B			
D	6	C	D	7	C	B	12	D			
						D	7	A			
						D	9	B			
						D	5	D			

We can see in the previous routetables the minimum cost for each node to reach to the rest nodes, and we can see that this is different from the information before the change in the link between A and C.

So the network is stable again.

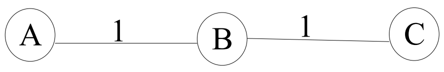
Example with slow convergence

In this example we are going to make a change in an stable network and check how this network is not going to be stable again because of the problem of slow convergence of the Distance-Vector algorithm.

Remember that this problem can be solved using Poison reverse and Split Horizont.

Creating the network

The network that we are going to simulate is next one:



This network correspon with next code:

```
node A
node B
node C
link a;b;1
link b;c;1
```

Stable routetables

NODE A			NODE B			NODE C		
Dest	Distance	Outgoing	Dest	Distance	Outgoing	Dest	Distance	Outgoing
B	1	B	A	1	A	A	2	B
C	2	B	A	3	C	B	1	B
			C	3	A			
			C	1	C			

In the previous routetables we can see the cost for each node to get to the others and the network is stable. Minimum values for each node are marked.

Changing a value

We are going to change the cost of the link between B and C to infinite and look what is going to happen.

Next we have shown the routetables with 50 iterations

Iteration 49

NODE A			NODE B			NODE C		
Dest	Distance	Outgoing	Dest	Distance	Outgoing	Dest	Distance	Outgoing
B	1	B	A	1	A	A	Infinite	B
C	52	B	A	Infinite	C	B	Infinite	B
			C	51	A			
			C	Infinite	C			

Iteration 50

NODE A			NODE B			NODE C		
Dest	Distance	Outgoing	Dest	Distance	Outgoing	Dest	Distance	Outgoing
B	1	B	A	1	A	A	Infinite	B
C	52	B	A	Infinite	C	B	Infinite	B
			C	53	A			
			C	Infinite	C			

As can be seen in the previous routetables, after 50 iterations the network is still unstable this is happening because Node A and Node B are still sending incorrect information.

This information has been marked with the previous arrows and is going to grow until infinite. To avoid this we can apply Poison reverse or Split Horizont.

Status report of the application

If more time will be given the application can include new features that are explained below ordered in categories

- Relative to the application
  - A GUI can be implemented, right now the application is object oriented and a new interface would be easily developed for it
  - Possibility to create networks directly using the application using a GUI interface
- Relative to the algorithm
  - Include possibility of using Poison reverse and Split Horizont