

# 1 Table of Contents

1	Table of Contents.....	1
2	Introduction.....	1
3	User Guide: .....	2
3.1	How to run:.....	2
3.2	How to Use.....	2
3.2.1	Create a node.....	3
3.2.2	Auto Parameter Setter.....	4
3.2.3	Send data.....	4
3.2.4	Route Error packet testing .....	5
3.3	Icons:.....	5
3.4	Conclusion.....	5

## 2 Introduction

This educational software simulates an Ad-hoc network with AODV algorithm, it uses a high level view and only works in Routing layer. it has a user-friendly graphical user interface and has been written in java language. Here is some of its features:

- Platform independent because of implementation in java language.
- interactive user interface (user can change nodes arrangement during the simulation)
- Drag&Drop feature
- Searching nodes by their name
- Automatic parameter setter : user can specify each node parameters: at least should see how many nodes and if their connection is bidirectional. Then the application fills automatically the nodes power, IP and name.
- Independence of UI and simulation engine; so developers can use another user interface for that engine.
- Logging the state of each node
- Animated UI

Developing characteristics:

- Java based

- Logging with Log4j
- Thread based

This application developed for Computer Network course in summer 2006 by [Masoud Moshref Javadi](mailto:moshref@ce.sharif.edu) (moshref@ce.sharif.edu) & [Ali Dabirmoghaddam](mailto:dabirmoghaddam@ce.sharif.edu) (dabirmoghaddam@ce.sharif.edu).

### 3 User Guide:

#### 3.1 How to run:

You can run the application by exe file, jar file, or sources.

- Exe file
  - Only run the exe file!!
- Jar file
  - Make sure that JRE has been installed and its bin directory is in PATH. (check this by typing “java” in command prompt (or shell).
  - Open a command prompt (or shell)
  - Change directory to which contains the jar file.
  - Type “java -jar JARFILENAME (replace JARFILENAME with the jar filename)
  - Now you should see the map initialization window.
- Source codes
  - The main class is UI.Myform. The source files packaged as an IntelliJ IDEA project. You can open it the program or import in Eclipse.

#### 3.2 How to Use

When you run the program, first the map size should be initialized. See figure 1

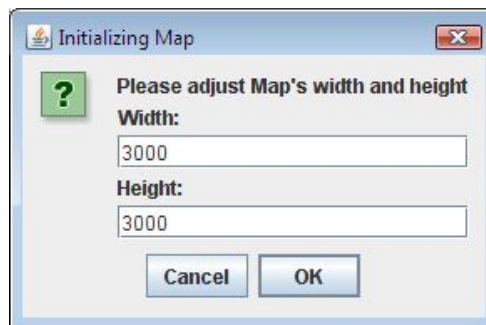
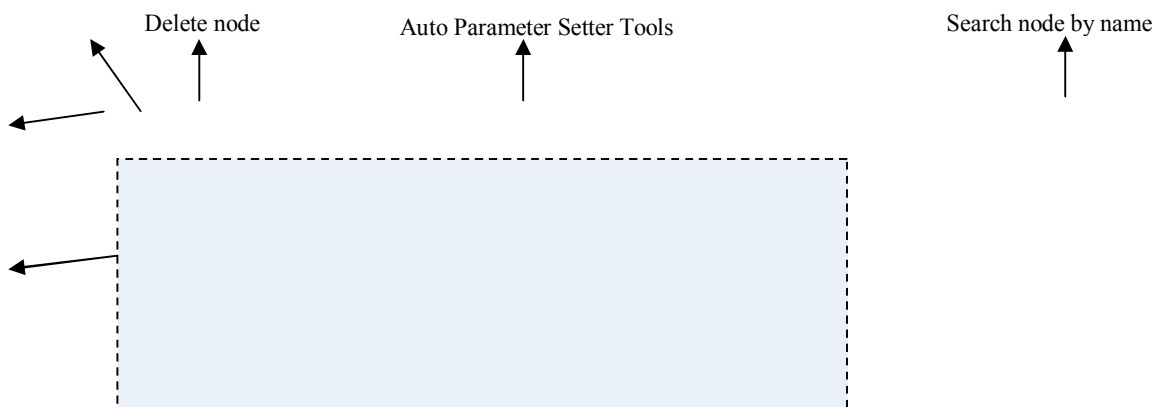


Figure 1: The simulation map size initialization dialog

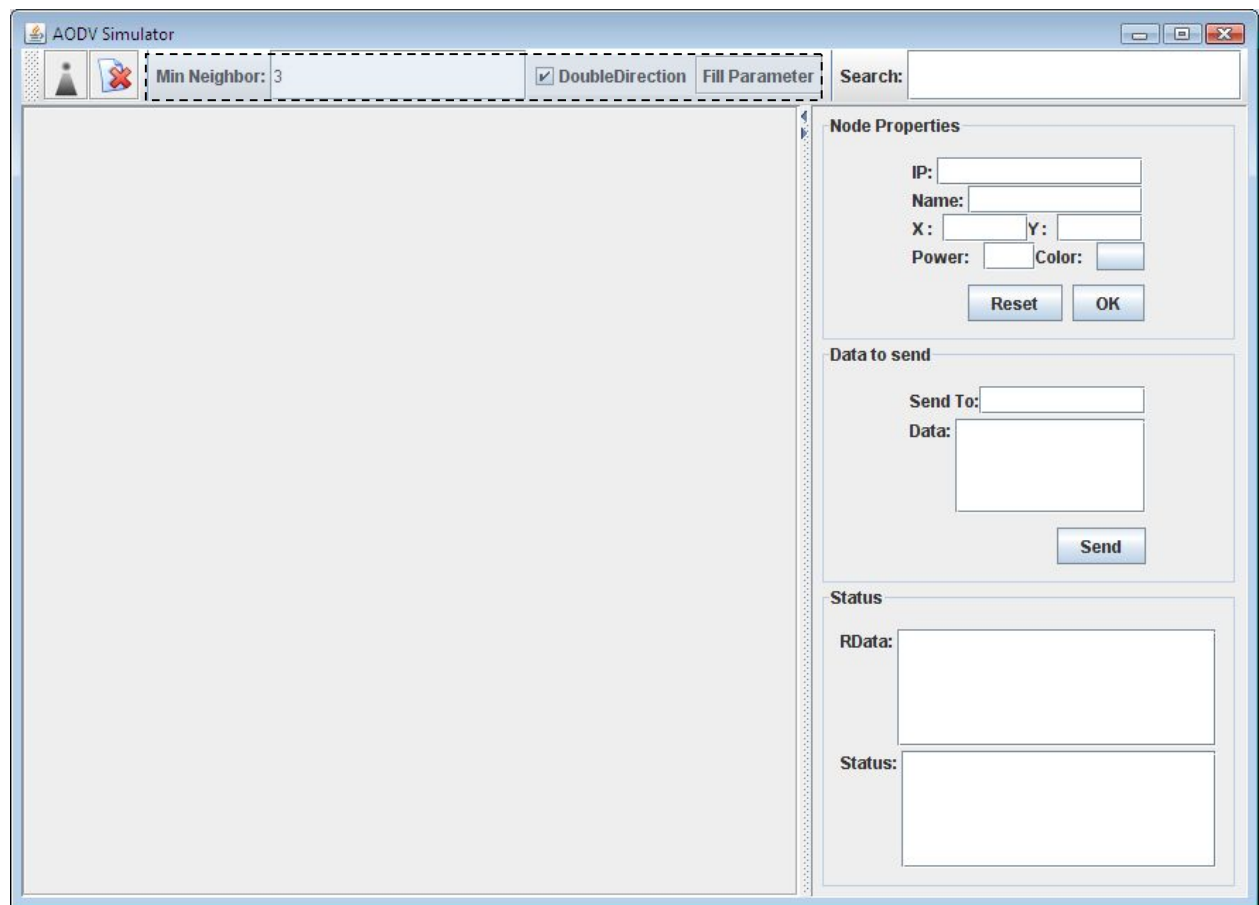
After that you can see the main window:



New node drag

Toolbar Handle

Map Area



Receiver

Data to send

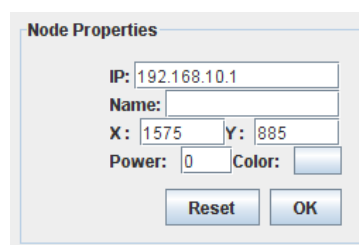
Received data

Node status

Figure 2: Main window

### 3.2.1 Create a node

To create a node, click on the “new node drag” icon  and drag it to the map. By default it has IP address 192.168.10.1 and no name:



You can set the name, IP and power of each node individually or use “Auto parameter setter tools” to set them see . Note the name of a node cannot be changed. The position of the node can be changed by drag & drop the nod or “Node Properties” panel.

### 3.2.2 Auto Parameter Setter

Auto parameter setter tools are positioned in the toolbar. It sets node's name, IP, and power automatically. Min neighbors and Double Direction parameters control the auto power setter algorithm. The first specifies number of node's neighbors. The neighbors connection can be either one directional or bidirectional (double direction) based on the second parameter.

A toolbar with three elements: a label 'Min Neighbor:' followed by a text input field containing the number '3', a checkbox labeled 'DoubleDirection' which is checked, and a button labeled 'Fill Parameter'.

Figure 3: Auto parameter setter

Usually first you position the nodes on the map, and then fill the parameters using this tool. To set the parameters click on “Fill Parameter” button. After that if you select a node you can see the parameters and set range on the map as a circle.

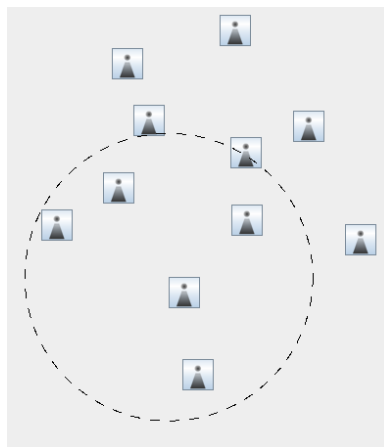


Figure 4: Node's range circle

### 3.2.3 Send data

To send data select the sender, type receiver name in the “Send to” field, and the message in “Data” field. Then push the “Send” button. You can find and select a node using the search box. The nodes' status and received data have been printed in “Status” panel for each node:

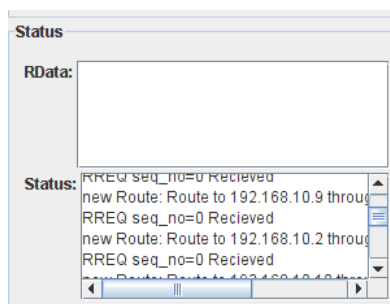


Figure 5: Node log panel

Also all nodes' logs have been printed in console window.

### 3.2.4 Route Error packet testing

To test the RERR packet, first send a data packet. Now before expiring the routes delete or move one of the nodes in data packet route.

### 3.3 Icons:



:Sending RREQ packet



: Sending RREP packet



: Sending Data packet



: Sending RERR packet

### 3.4 Conclusion

We know that the simulator is very imperfect. But we believe it can be used for educational purposes and demo AODV protocol mechanism.

Documentation by Masoud Moshref.