**CHAT APPLICATION**

**USING TCP**

**GROUP 4 SE1309:**

**Members:**

**Nguyễn Tiến Đạt**

**Phạm Tiến Mạnh**

**Đỗ Minh Hải**

**Phạm Quốc Đức**

**CONTENTS**

1. **INTRODUCTION.**
2. **SYSTEMS ARCHITECHTURE.**
3. **Specifically.**
4. **Flow chart.**
5. **Operational Concepts and Scenario.**

* **List Form.**
* **Chart Form.**

1. **RESPECT TO THE DEVELOPED PROGRAM.**

* **Routing in the Internet.**
* **IEEE 802.11 wireless LANs.**

1. **Network Security.**

* **What is network security.**
* **Message integrity.**
* **Securing TCP connections: SSL.**
* **Operational security: firewalls and IDS.**

1. **CONCLUSION.**
2. **INTRODUCTION**

Teleconferencing or Chatting, is a method of using technology to bring people and ideas “together” despite of the geographical barriers. The technology has been available for years but the acceptance it was quit recent. Our project is an example of a chat server. It is made up of 2 applications the client application, which runs on the user’s Pc and server application, which runs on any Pc on the network. To start chatting client should get connected to server where they can practice one kinds of chatting: Private (between any 2 users only)

1. **SYSTEM ARCHITECTURE**

The user interacts with the software using a GUI.

• The GUI operates in two forms, the List form & the chat form.

• The List form contains the names of all the systems connected to a network.

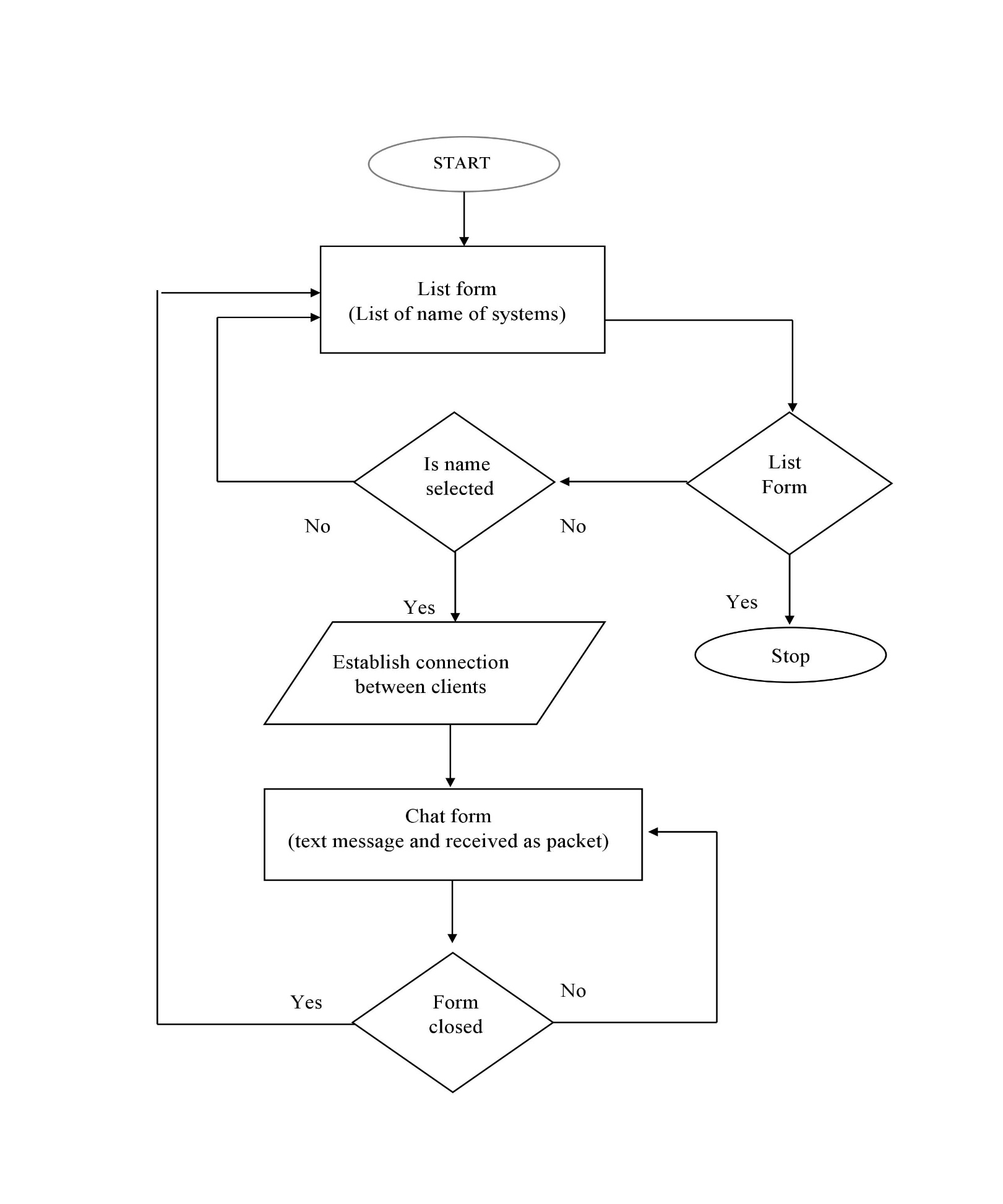
• The chat form makes the actual communication possible in the form of text.

**Specifically:**

• List form: In this form, all the names of the systems connected to a network are enlisted. These names can later be used for communication with the help of mouse event, or in simple language: a click or a double click.

• Chat form This form is called only when an element is selected from the List form. In this form, a connection is created between the host system and the selected system with the help of a socket.

**Flow chart:**



**\*\*Operational Concepts and Scenarios**

Operation of the application based on the inputs given by the user:

**List Form:**

• When initialized, The list panel is empty, clicking the start button will intiate

connection to server and list all user currently connected in List panel

• When a name is double clicked, the chat form is initialized with a connection between the host and the client machine.

**Chat form:**

• Contains a textbox which cannot be edited but only displays the messages from one user to another, including the self sent message, as in any chat application.

• Contains a textbox for messages to be written that is sent across the network. • Contains a Send button.

• When the sent button is clicked, in the background, the text in the textbox is encoded and sent as a packet over the network to the client machine. Here this message is decoded and is shown in the textbox.

• To make it more realistic, the self sent message is shown in the textbox as well. Both the messages is differentiated by the help of the identifier name at the beginning of each message in the text box.

EXIT: The user exits the software in two scenarios:

• Exits the chat form, the list form remains intact. Exits the list form, this is when the application is closed.

**III. RESPECT TO THE DEVELOPED PROGRAM.**

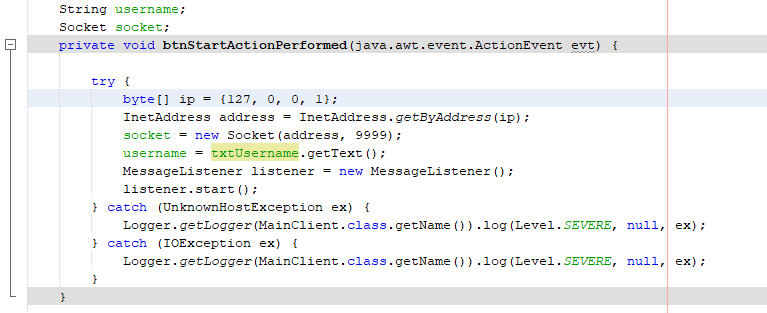
**1.1 Routing in the Internet: BGP**

- BGP uses TCP to communicate between routers (any devices exchanging routing information.) The information exchanged is used by the BGP peers, to better choose the way they choose where to send, (aka, next-hop) packets that they need to transmit.

**1.2 IEEE 802.11 wireless LANs**

- Each computer on your WiFi network will have an IP address. Using one computer as the host, you create your ServerSocket. Your other computer will be the client, and can connect to the IP address of your host.

- For example:



Here the clients will connect to the host IP which is:” 127.0.0.1 aka localhost from your computer, that will resign the clients to connect via wifi.

- The base station (or access point) will be the router

- The host (your computer) will:

+ Scan channels, listen for each SSID and Mac address of other computer

+ Select AP to associate with

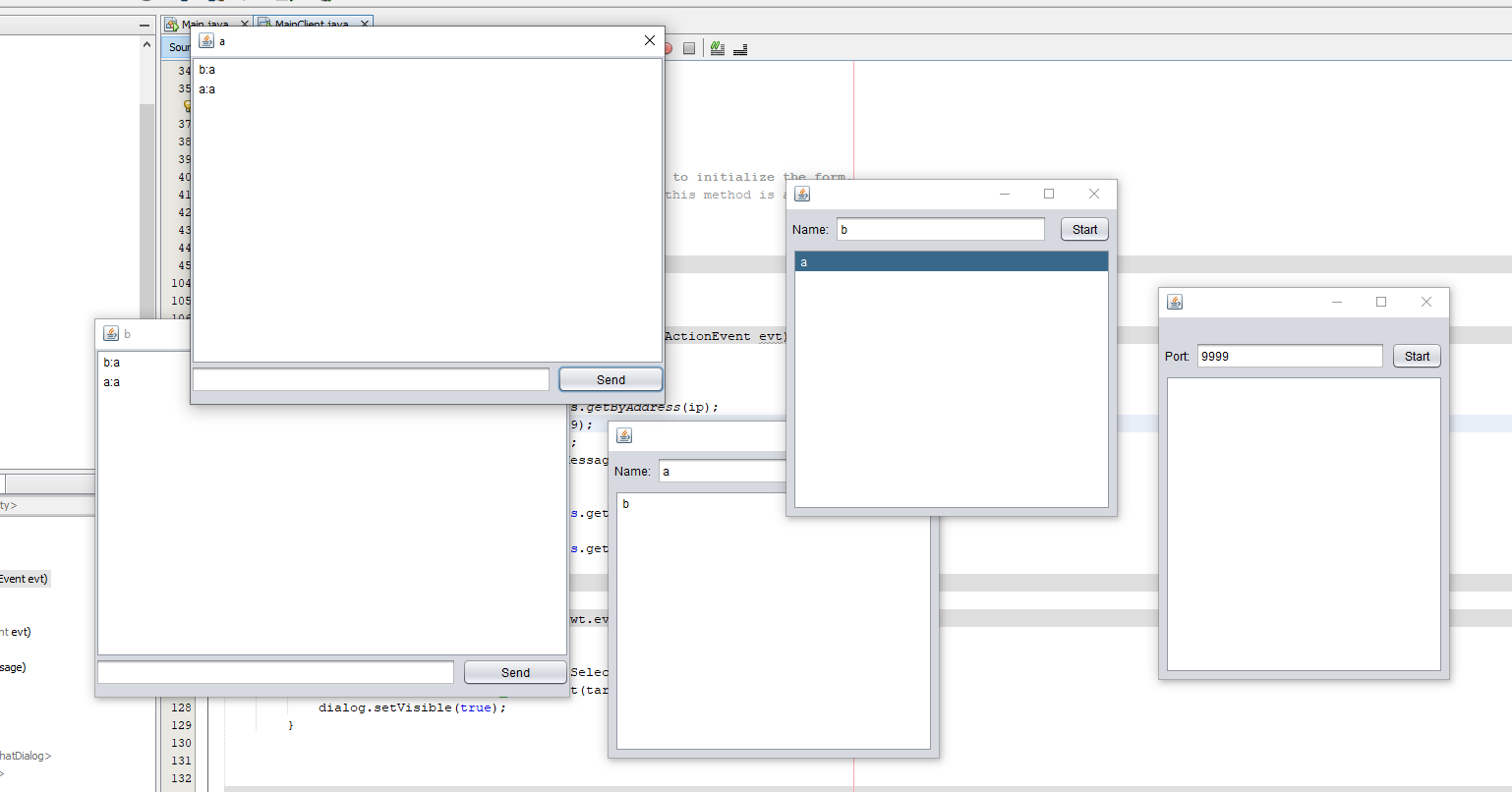
+ Run DHCP to get IP address in AP’s subnet

**2. Network Security:**

**2.1 What is network security?**

Confidentiality:

* The Client A send the encrypts message to the Client B, the message must go through the Server (middle point) then the Client B can decrypts message.



Authentication:

* The Client A has username a, and the Client B has username b, the all see each other as username and can confirm identity of each other to begin chat.

Message integrity:

* The message will only show on each client chat ui, not from other, not anyone else can see the message, if the close the chat ui, the message is gone without detection

Access and availability:

* The Server must start and open IP host, port to the client to connect, and the client must type a username otherwise cant begin the chat system

**2.2 Message integrity**

- Client A send a message to Client B has:

+ message originally came from A

+ message not changed since sent by A

**Digital Signatures**

- Client A signed as a, no one can signed as a again or another

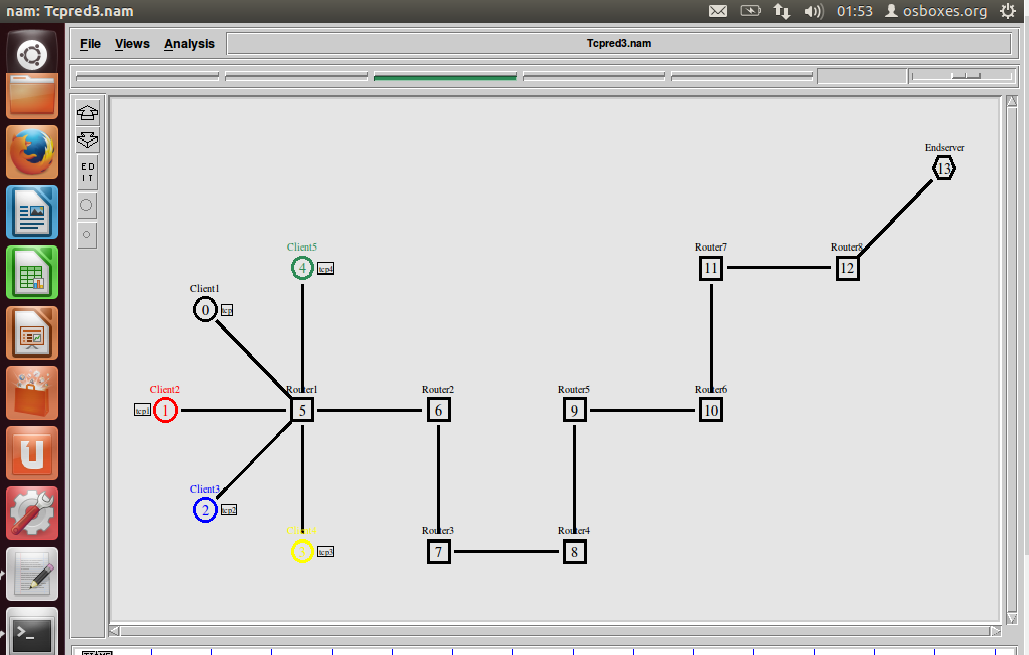
**2.3 Securing TCP connections: SSL**

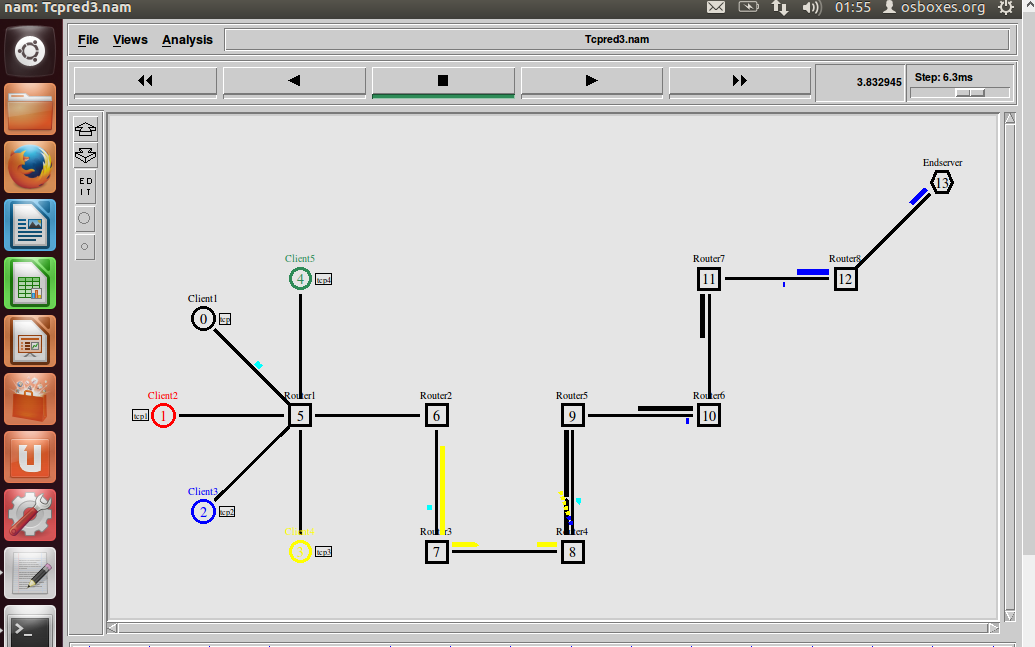
- provides transport layer security to any TCP-based application using SSL services.

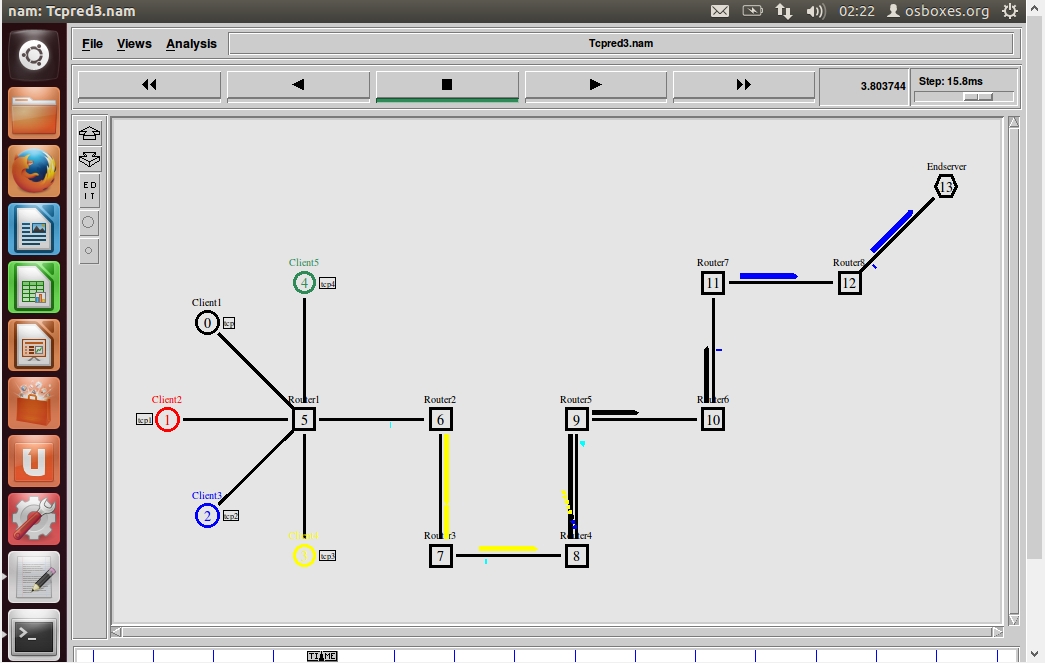
**2.4 Operational security: firewalls and IDS**

- The server only start with signed and opened port to the client to connect, if has the firewall, the other computer cant signin to the chat system without going to the right port of server

**Simulate the network using NS2 software**







**IV. CONCLUSION**

This chapter has given a broad picture of the design of the software in terms of the different modules used. It also gives us an idea about the degree to which each module performs related tasks. We also get an idea about the interdependence among the modules.