Report document for Chat Room Program and Network Simulation

NWC202 - Group 1

Nguyen Minh Quang  
Student code: HE130059  
Class: SE1310Email: quangnmhe130059@fpt.edu.vn

Pham Quang Huy  
Student code: HE130022  
Class: SE1310Email: huypqhe130022@fpt.edu.vn

Nguyen Quang Huy  
Student code: HE130069  
Class: SE1310Email: huynqhe130069@fpt.edu.vn

Dieu Gia Bao  
Student code: HE130033  
Class: SE1310Email: baodghe130033@fpt.edu.vn

*Abstract*—Due to remote communication problems of people in several regions and the popularity of Internet, group chat application is obviously necessary in daily life. Our project includes an example of group chat program and the network simulation for it. Group chat program contains 2 applications, which are client application – runs on user’s computer which has installed JRE and server application – runs on any computer that installed JRE on the network. To start chatting client must connect to the server. The simulation of network is also clearly described.

Keywords—chat room program, client, server, message, network simulation

# Introduction

This document describe 2 main parts: details of chat room program, and simulate the network of system, how message was sent from clients to server. The simulation is visualized by visualization tool of ns2 – NAM (Network Animator).

The first part is about chat room program (CRP), nowadays there are numerous products available for real time “chatting” over Internet. One of main objectives of this project is to implement a Java based chat application using TCP protocol that allow communication between users in LAN. This could be useful with those system that connects intranet, i.e. system of a company, or public Internet center where admin could interact with users and vice versa…

The second part is about network simulation of the program, visualize how packet was sent from clients to server. As noted above, this simulation is implement by using NAM, a visualization tool of ns2. Some evaluations are also generated by considering the trace file and graph of cwnd (congestion window).

# CHAT ROOM PROGRAM

## Product perspective

* CHAT ROOM PROGRAM is aim toward user who:
  + Want to message with your friend or your family
  + Make friend with people in chat room
  + Keep in touch with other people in chat room

## Operating Environment

* Runs on any operating system has internet.

## Hardware interfaces

* Client: Any computer has same internet connection
* Server: Any computer
* Protocol: TCP

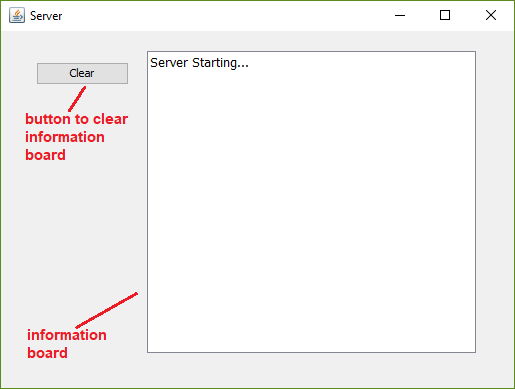
## Software interfaces

* All are Java only

## System main functions

### **Server application**

* 1. *Open server*
* Interface

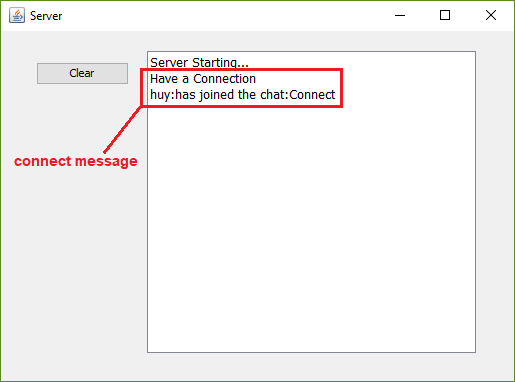


* Description:

After execute server program, above window will appear and server start running.

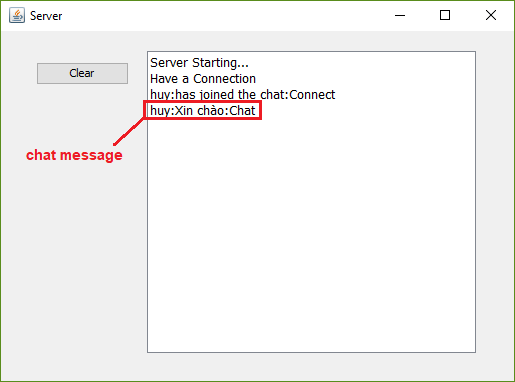
* 1. *Clear information board*
* Description:

After clicking the ‘Clear’ button, all text in information board is removed.

* 1. *When a client connect to server*
* Interface
* Description:

When a client connect to server, above message will appear on information board.

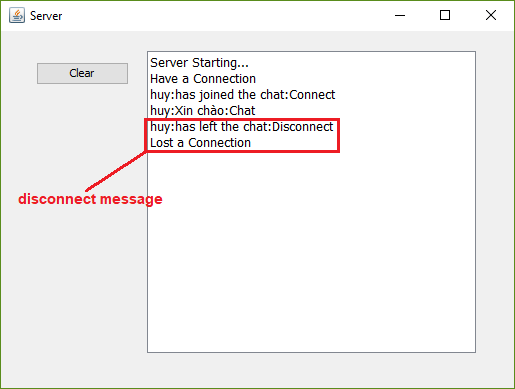
* 1. *When a client chat*
* Interface



* Description:

When a client send a chat message on group chat, above message will appear on information board.

* 1. *When a client disconnect from server*
* Interface

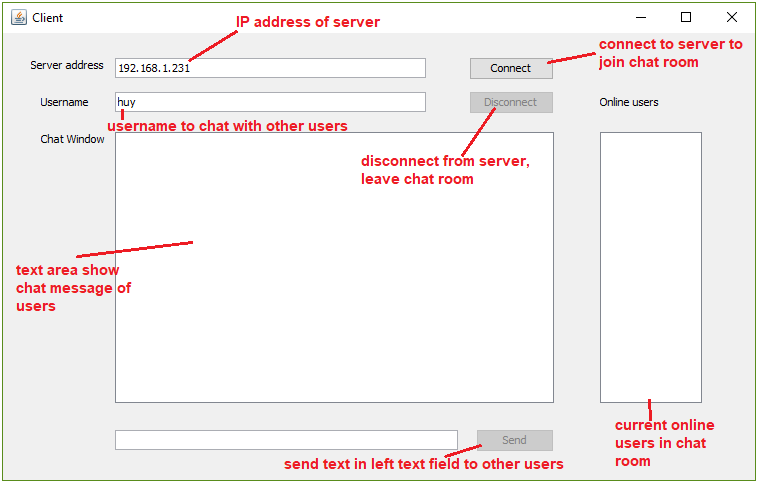


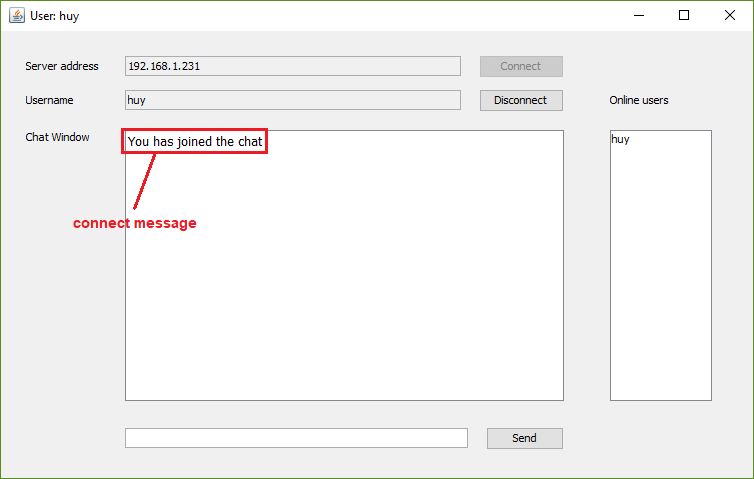
* Description:

When a client disconnect from server, above message will appear on information board.

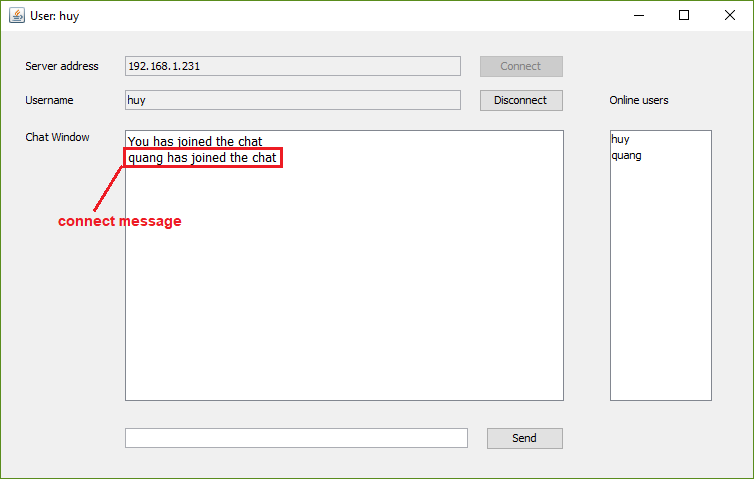
* 1. *Close the server*
* Description: server will close when admin click on ‘×’ icon at top right position of window.

### **Client application**

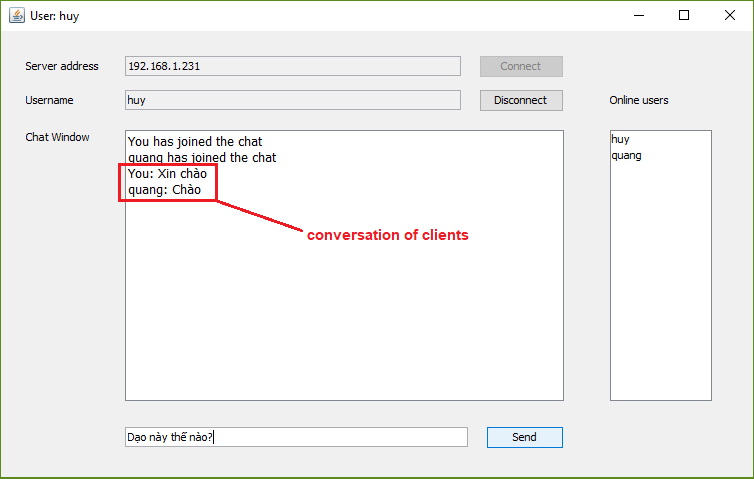
Starting interface:

* 1. *Connect to server*
* Interface
* Description:

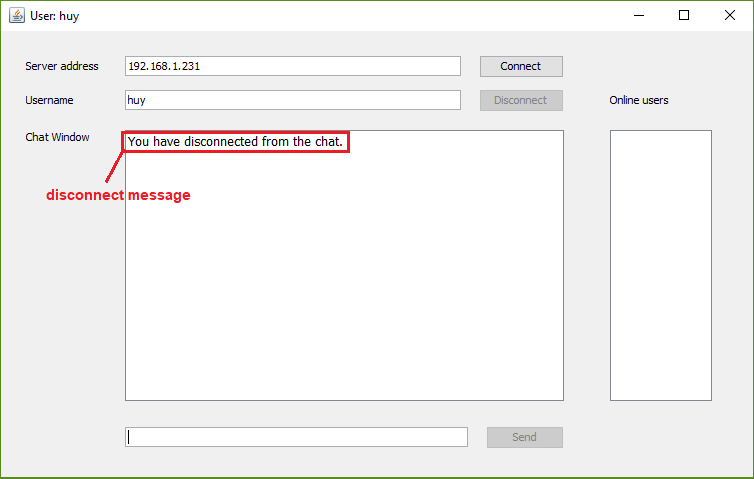
The IP address of server must be correct then client is successfully connect to server.

* 1. *When other client connect to server*
* Interface
* Description:

When other client connects to server, above message will appear on chat window, online user list will be update.

* 1. *Send message to other clients (Chatting)*
* Interface
* Description:

Enter text in chat field then click “Send”, conversation of members in chat room is shown on chat window.

* 1. *Disconnect from server*
* Interface
* Description:

After disconnect from server, above message will appear on chat window.

# network simulation

## Introduction to Network Simulator

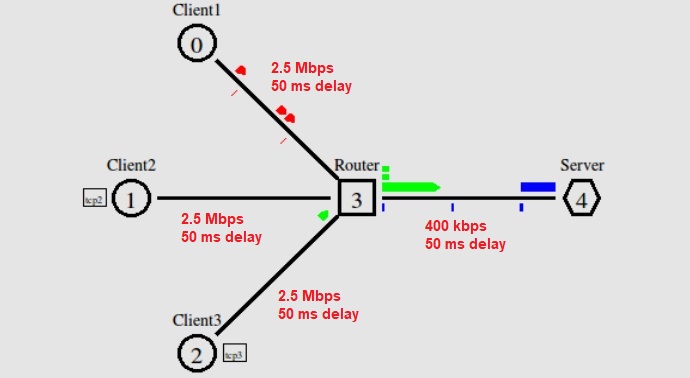
Network Simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/routers, data links, packets, etc) using mathematical formulas, or actually capturing and playing back observations from a production network. When a simulation program is used in conjunction with live applications and services in order to observe end-to-end performance to the user desktop, this technique is also referred to as network emulation.

A network simulator is a software program that imitates the working of a computer network. In simulators, the computer network is typically modeled with devices, traffic etc and the performance is analyzed. Typically, users can then customize the simulator to fulfill their specific analysis needs. Simulators typically come with support for the most popular protocols in use today, such as IPv4, IPv6, UDP, and TCP.

The notable network simulators available are NS-2 and OPNET. The most popular Open-Source Simulators available in the market are NS (also called NS-2), PDNS (Parallel/Distributed NS), GloMoSim, SSFNet (Scalable Simulation Framework Net Models), DaSSF (Dartmouth SSF), OMNET++ and others.

## Analyzing simulation

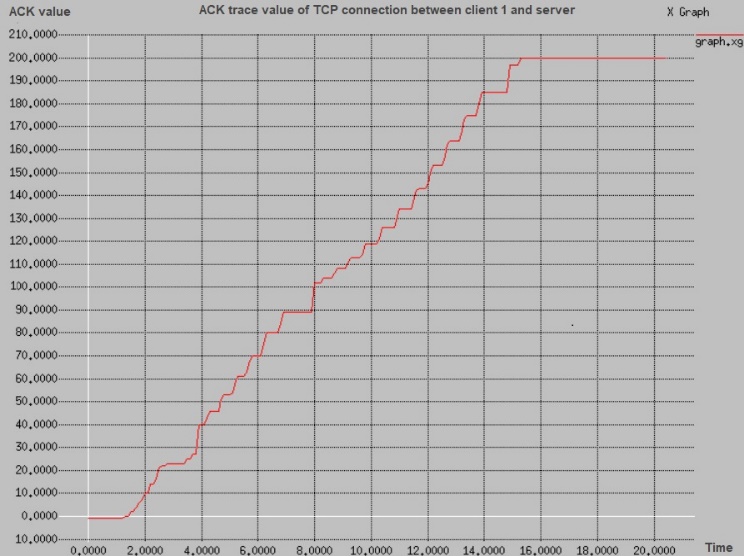
In this project, we use ns2 to simulate how packets were sent from clients to server. Here is the topology we build:



The network consists of 3 client nodes (0, 1, 2), 1 router node (3) and 1 server node (4). All duplex-links between clients and router are the same, with 2.5 Mbps of bandwidth, 50ms of delay and DropTail queue (actually useless because clients are hosts). Also, the duplex-link between router and end-server has 400kbps of bandwidth, 50ms of delay and DropTail queue. The smaller bandwidth on router-server link makes it the bottleneck. Therefore, bottleneck bandwidth is 400kbps = 50 packets/sec because the total packet size is 1000 bytes (960 bytes data as set + 20 bytes tcp header + 20 bytes ip header). We set the queue at router to contain maximum 20 packet so that packet drop event could exist 🡪 packet retransmit. TCP agents are attached to all clients with TCPSink attached to endserver in order to establish TCP connection between client and server. TCPSink agent at server sends ACK packets back to clients and also frees reveive packet. We use FTP to generate random packets and attach them on the TCP agents; all FTPs start sending packet at T = 1.0 and stop at T = 15.0. Color of packets from client 1, client 2, client 3 are red, blue, green respectively.

## Evaluation

The graph below is generate by xgraph to examine the ack values of TCP connection between client 1 (node 0) and server:



Because the maximum value of ACK packet is 200, we could conclude that the number of successful sent packets is 201 (the beginning ack is 0). Since the total number of sent packets at client 1 is about 233 (note that 50 packets/sec is bottleneck bandwidth, 14 seconds is the tranmission time – FTP start sending at T = 1.0 and stop at T = 15.0), so this is about an 82% goodput.

##### References

1. T S Pradeep Kumar, “Print the Congestion Window Graph”, <https://www.nsnam.com/2013/04/print-congestion-window-graph-in-ns2.html>, April 23, 2013.
2. Peter L Dordal, “Department of Computer Science”, 16 Network Simulations: ns-2, <http://intronetworks.cs.luc.edu/current/html/ns2.html?fbclid=IwAR3PGuZpRrWu1u_bGT31YVXHkud6wWsliqAuFNQZ8EF4z8cy3V6TEBh3HvQ#network-simulations-ns-2>.
3. Kurose Ross, “Telnet: A Case Study for Sequence and Acknowledgment Numbers”, Computer Networking A Top Down Approach – 6th edition.