

Swinburne University of Technology
Faculty of Information and Communication Technologies
Data Communications & Security
Assignment 2 – 10%
Chat Network

One or Two

This assignment may be done alone or with ONE other student.

If you intend to work in a pair, you **MUST** register your partnership with your tutor before week 8 by sending your tutor an email.

Pairs must be in the same lab.

Overview of System

You are to develop a chat and file sharing program. It **must be a console application**, written in Java and executed from the command line.

All hosts will have **the same software**.

- Your program must initially investigate the state of the network. Your IP address, network address, host number. (see print out for further detail)
- When a peer starts running it will search for other hosts on the same subnet.
- Next, each host will be tested to see if it is part of your chat network. It must recognize a unique code assigned to your group.
- For those in your chat network, their names and IP must be displayed on the screen.
- The user may type any message and it will be sent to all other peers.
- The user may type the word <File,IP,filename> and this will send the file to the nominated peer.
- A peer that receives a message will display the message along with the name of the sender and their IP.
- A peer that receives a file will store it in a folder named FilesReceived.
- A peer must tell other peers it is leaving the network by sending a message <BYE>
- Bonus marks (20%) will be awarded to students who encrypt the file before sending and then decrypt the file at the receiving end.

Technologies & knowledge areas

- Writing a peer to peer network application
- Use of TCP using java sockets
- Use of UDP using java sockets
- Encryption of files using java security API's (For those going for a Bonus)
- Peer to Peer architectures
- Client Server architectures
- Use of Threads

Key Parts

- Peer - Discovers who is on line
- Peer - Learns about subnet
- Peer responds to requests to check they are online
- Peer sends chat to all other clients (multicast socket ok)
- Peer can send file to a particular peer
- Peer can receive message and display message
- Peer can receive file
- Bonus 20% for encryption of file
- Information gathered about network at time program starts
- New peers are added as they join the network

Some Rules

- File exchange must use TCP
- All other exchanges must use UDP
- Your solution must use multiple threads.
- A GUI interface **is not** allowed.
- Must use port 4000-4010
- All projects must be run on the real networks in the DCS labs (EN305/EN310)
- Only class libraries that are installed on the lab machines may be used.
(The ones that come with the standard install)
- The program must be written in java.
- Your program **must** run from the command line
(Make sure you know what this means)
- There may be only one main method in your application, but it may create one or more threads.
- The peer search is to be limited to 10 peers either side of your own host number.
- All chatters in your network must share and know of a unique code that allows them to participate in the network of chatters.

Suggestions

- Write a UDPMessageServer Class (sends messages)
 - It will be a Thread
 - Include two methods

sendMessage(sendTo:String, message:String)

Its job is to create a datagram packet and send it to the nominated IP.

- Write a MessageProcessServer Class (receives messages)
 - It will be a thread
 - It will have constructor that receives a port, a code, a name and an array of known connected peers
 - It will have a run method
 - It will open a socket waiting for UDP datagrams.

Its job is to receive datagrams from other peers and process the data received.

- Define your protocol early, and discuss with your tutor, it will help you build the program

Define user input

Define what will be displayed

Define what messages will be sent and the format they will be in

Define what the recipient of a message will do, what will be displayed and what will its reply be, if any?

The protocol will be discussed further in lectures.

- Provide a ChatProgram class (is the User Interface)

Has a public static void main

Reviews current network

Discovers chatting peers

Starts any servers needed

Manages chat from user

Displays messages received

- Write a FileServerClass that receives files & is a thread (receives files)
- Write a fileSender class that is a thread (sends files)

Qualities

This is a code writing assignment and it is expected that a significant amount of the code you write will be sourced from the work of others (Dr. Google). The work sourced from others **MUST BE acknowledged** by the use of inline comments in your source code.

You may not submit other students work.

You may not ask someone to write part or all of the code or pay someone to write part or all of the code.

The main criteria to pass are;

1. Your program works, does everything described here (all parts)
2. Your program does not hang or crash.
3. You provide code that is meticulously document using Javadoc
4. You reference your sources.
5. You show you have a very good understanding of the code you have submitted.
6. You successfully demonstrate the code in class to your tutor on the lab computers.
7. You can make a minor change to the code during the demonstration.

Submission Requirements:

To ESP (Code Section), by 5pm on 23rd October 2016

Java source code and all files needed to run at the command line

All .class files (in a single zip file)

<https://esp.ict.swin.edu.au/>

If you do the work as a pair, both students must submit a copy to ESP.

Submission must include source code for each part, with detailed comments that explain the code

Each student must **independently write their own inline comments** explaining how the code works.

.

To be demonstrated in your lab class in week 12.