UBS/UFR SS1 – M1 SAIM

Distributed computing
Java TCP/UDP networking

Lab 3: UDP – gossiping

The objective of this lab is to develop a peer-to-peer application allowing users to exchange files with each others.

The application allows two users to exchange the files they hold through TCP connections, by first gossiping (through UDP messages) about the files they could provide to the remote user and the files they are interested in or not amongst those proposed by the remote user. The messages exchanged also carry various informations such as attributes about the UDP/TCP sockets to be used to exchange messages and/or transfer files.

Each user has a unique name composed of n ascii characters, and has a directory to store the files to be sent and the files received. This directory contains subdirectories having the names of the different users of the application. The application will write the files received in the directory having the name of the current user and look in the other directories contain for files to send to remote users. Those directories are filled in by an external application (for this lab, the files will be added "manually" or using the mailer and SMTP server used in the previous lab).

The messages exchanged between two users user1 – that starts the gossiping – and user2 – that waits for messages – are the following[1],[2]:

- offer message (code: 1): sent by user1 to propose a file he holds to user2, and inform him that the reply (request or delete) should be sent to "IP_addr:udp_port"

code	version	IP_addr	udp_port	n1	user2_name	n2	file_name
1	1	4 or 16	2	1	n1	1	n2

- request message (code: 2): sent by user2 to request the file, and ask him to send the file at "IP_addr:tcp_port"

code	version	IP_addr	tcp_port	n1	user2_name	n2	file_name
1	1	4 or 16	2	1	n1	1	n2

- delete message (code: 3): sent by user2 to ask user1 to delete the file

code	n1	user2_name	n2	file_name
1	1	n1	1	n2

When starting, each peer gossiping program has as arguments:

- the local user name,
- the path to the base directory that contains the subdirectories where the files sent/received will be stored,
- a UDP port number on which it should wait for remote gossiping messages,
- a TCP port number where the files will be received and stored in the subdirectory having the local user name.

While running, the program asks for informations about remote users to start gossiping with (user name, host name and UDP port number).

Implement the peer program according to the adopted protocol. You should use the code developped in the previous lab to implement the file transmission over TCP

Assignment: give an executable jar file with the Java sources. The program should display a help message describing the expected command line arguments when it is launched with the -h option

[1] the value under each field of the messages represents the field's length in bytes. n1 and n2 are the length of user2_name and file_name

[2] the version field represents the IP version (4 or 6) and determines the address length (4 or 16)