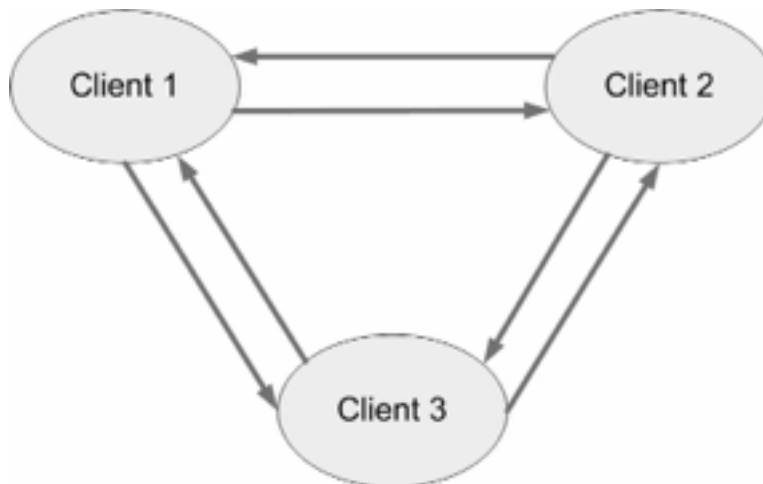


Experiment 4

Peer to Peer Application development

The task is to build a server-less peer-to-peer system, presented in the following figure, in which client-to-client file exchange takes place.



Implementation must Linux socket programming using C (TCP Socket). Your application should work as follows:

A client communicates with other clients to send and receive files from them. Of course, the client application also presents a user interface to the user. For receiving requests from other clients it is required that each client listens to some port (say 3300).

When a client starts, it broadcasts a packet of type 1. In this packet the user name of the user who comes online is specified. All other online clients should respond to this client with a packet of type 2, which contains their username. This allows the client to build a list of online users. This process has to be repeated every 60 seconds to maintain an up to date user list.

A packet of type 3 is broadcasted as soon as a user is looking for a file to download. This packet encapsulates the filename the user is looking for. Each client that is sharing this file should respond to the requesting client with a packet of type 4. Consequently, the requesting client can present the user with a list of usernames that share the queried file.

The user can now select the username from which to download the requested file. In order to achieve file transfer, a packet of type 5 is sent to the client from whom the file should be downloaded. The client that receives this request responds with a packet of type 6. This packet contains the TCP port on which the file is being offered and the file size in bytes. This enables the downloading client to connect to this TCP port and receive the file. It must be possible to provide simultaneous full-duplex file transfers.

Packet formats

Packet type 1: Request for online users



NLEN – length of USERNAME
 USERNAME – name of the sender who comes online

Packet type 2: Online user information



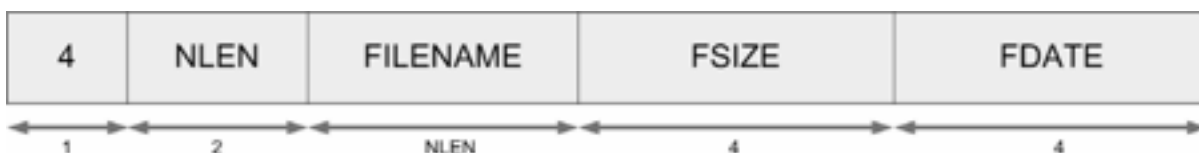
NLEN - length of USERNAME
 USERNAME - name of the sender who is online

Packet type 3: Search for file



NLEN - length of FILENAME
 FILENAME - name of the file the sender is looking for

Packet type 4: Sharing acknowledgement for file



NLEN - length of FILENAME
 FILENAME - name of the file the sender shares
 FSIZE - size of the file the sender shares
 FDATE – creation (or modification) date of the file the sender shares

Packet type 5: Request for file transfer



NLEN - length of FILENAME
 FILENAME – name of the file which the sender wants to receive

Packet type 6: Offering file transfer



NLEN - length of FILENAME

FILENAME – name of the file which the sender is offering

FSIZE – size of the file which the sender is offering

PORT – TCP port number on which the sender is offering the file