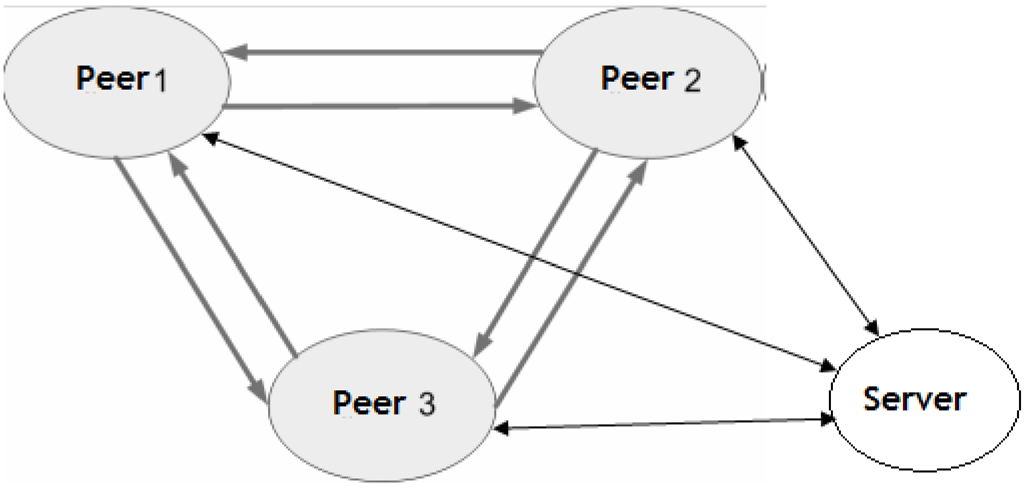
Experiment 4

Peer to Peer Application development

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



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

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

When a peer starts, it sends a packet of type 1 to the directory server . In this packet the user port number of the user who comes online is specified along with the names of files with the peer.

Packet type 1: Request for online users

1	PORT NO	File1	....	FileN

A packet of type 2 is send to server as soon as a user is looking for a file to download. This packet encapsulates the filename the user is looking for. The server responds to the requesting peer with a packet of type 3.

2	FILENAME

3	Peer1	Peer1 FileNames		PeerN	PeerN FileNames

Here peer number should be replaced by port numbers of the corresponding peers

Consequently, the requesting peer should make a connection the peer having the queried file and sent a type 2 packet to request the file. The server should not be involved for file transfer. The server should maintain a data structure of connected peers port numbers and the files with each of such peers.

You must show the concurrent file transfer with a particular peer where the peer is acting as both client and server. That is the peer is question (say peer 1) will be acting as server for peer2 at the same time it is receiving file from peer3 as a client. This part will require concurrent connection mechanism. Make sure that your program is not terminated immediately after one transfer or so (use a infinite looping option before the connection stage).