0. Before starting this program

This Program need a configuration file except for the source code of this program. The file ‘configuration.xml’ contains server ID, server IP, server Port (Basically the IP address are set to localhost). Also, there should be the shared directory to transfer or download from other neighbors.

1. Main driver - (Gnutella.java)

Firstly, this main driver reads the network topology and store the information of servers into running program so that the program can be ready to participate in the file sharing network. Secondly, there is a String variable ‘sharedDirectoryPath’ in this driver that stores the path of the shared directory. When this program starts, user is supposed to enter the server id. After entering the server id, the shared directory’s name become ‘sharedDrectoryPath+ServerID’ (for example, when the server id is 1 then shared directory is ~/incoming1). Then, this driver gets the neighbor numbers by console input to connect to the neighbor of this servent. Now all the requirement for starting file sharing is set. Finally, this driver starts the client thread and server thread at the same time so that this program can obtain a file and react to other messages concurrently.

2. Client - (ServentClient.java)

The client is implemented with single thread for getting user input and requests from other servents concurrently. Even though the server is running, user can request to search or obtain file in the network without any disturb. Once this client thread starts, it prints the menu on the console. After selecting a menu, it prints the menu iteratively. With this menu, the user can search a file the file, obtain the file and exit this program. When searching is selected, ‘Query’ method is executed to make a new query for searching the file. After making an initial query, this client passes the query to server to broadcast the query into entire network (result of searching will be described at server design). When obtaining is selected, this client part prints the list of servent that are holding the file. Then user can select the servent to download. After that, this client make a new message for downloading consisted of ‘obtain, filename, requestorServentIP, requestorServentPort’. When the server gets this message it starts upload threads that connect to this server.

3. Server - (ServentServer.java)

The serve is also implemented with thread but for the multiple connection between neighbors, threads is created when the connection is established from any neighbors.
1) Waits connection requests from single or multiple neighbors. And make thread when connection is established. This server has an inner thread that is called ServerProcessor. When this server’s serversocket gets connection, it passes the accepted socket to the server processor

then the server processor processes every request. There are 3 types of messages query(broadcast), queryhit, and download request. Each message is consisted of information of specific server information that are proper to operations (such as query, query hit, obtaining).

Query Message :
SeqNo/IP(where the query made)/Message Sender(where the message was sent)/filename/TTL

Queryhit Message :
SeqNo/IP(that has the file)/List of Sender(for traceback)/IP(where the query made)/ servent id of sender/filename/TTL/hit

Obtain Message :
obtain, filename, serverip, serverport

When the server gets query message, the server looks up the local file list. If file exists, the server makes new query hit message and trace back to the original servent. If file not exists, the IP address of the servent(hop) is added to the sender list and broadcasted until the file is found or TTL is expired. And when the server getrs query hit message, the server compares the last item of sender list and current server information. When the information is same, the original sender recognizes that it is the message that indicates specific servent is hoilding the requested file. When the server gets obtain message then the server starts file uploader with the requestor server information.

4. Data Structure - (MessageID.java ServerInfo.java)
ServerInfo stores information of server information that are given in the configuration file. When the program need to get information about network or other servers, the program will access to the ServerInfo to get specific information.MessageId store information of messages like sequence number, original requestor, ipAddress and port number when they are required.

5. File transfer - (FileDownloader.java FileUploader.java)
File downloader and file uploader were implemented with thread so that multiple request and response are done concurrently. So the requests or responses don't needed to be queued or stored to be processed later.

6. XML Parser
XML Parser gets network information from 'configuration.xml' so that the driver can use it.

7. Improvement and extensions.
If the network is much huge, ping-pong function must be implemented so that the servent really can distribute the burden of tasks. Also, there can be a tracker which will function storing multiple servents when a servent download a big file. If the tracker list would be implemented, we should consider the balancing on downloading a big file from multiple server.