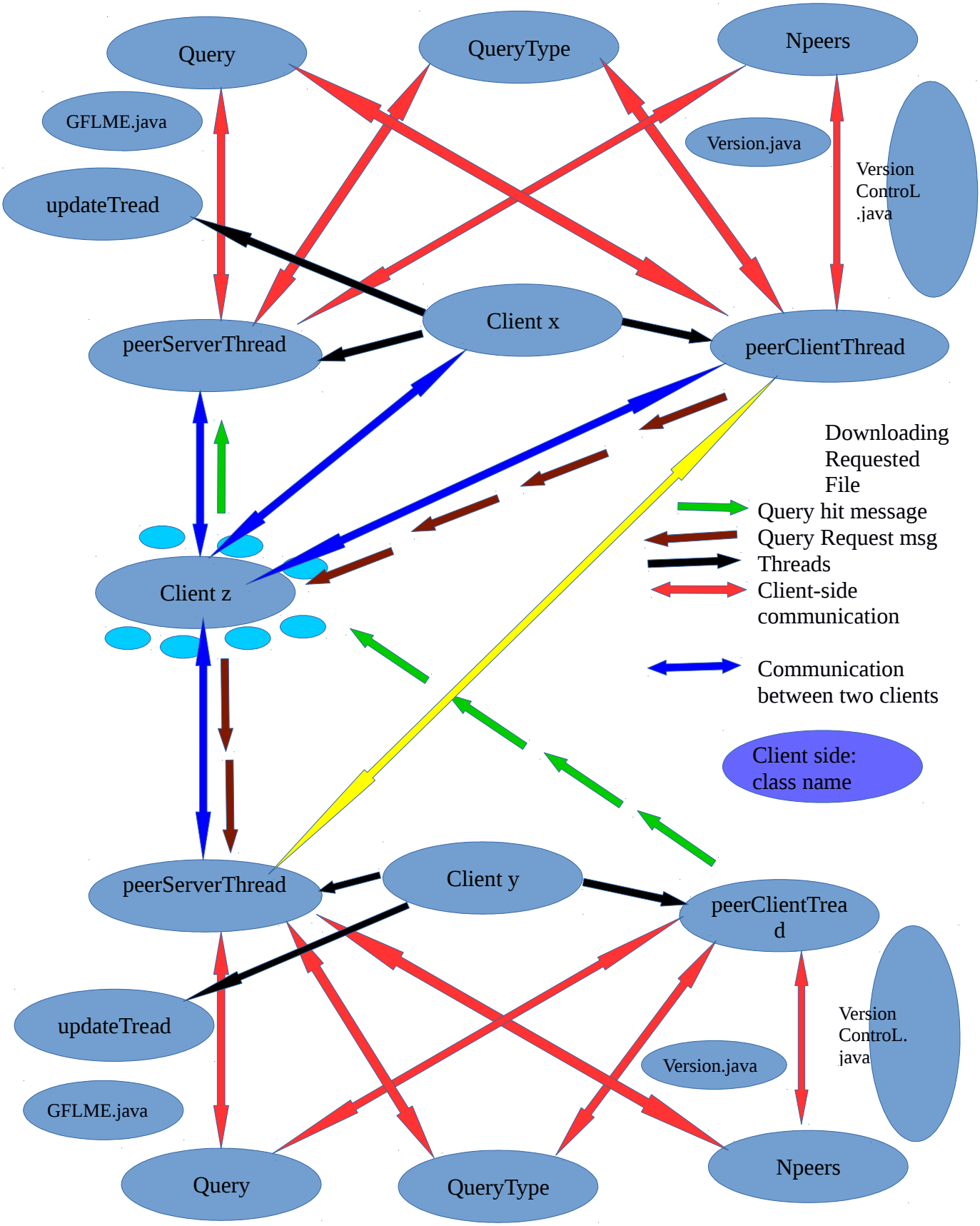
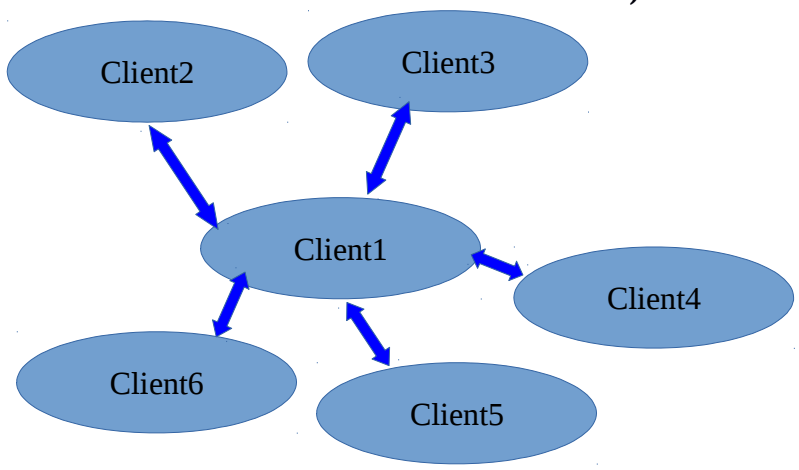
# Our Peer to Peer File Sharing System Design.

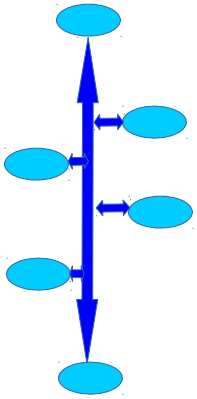


**TOPOLOGIES:**

**1)STAR :**

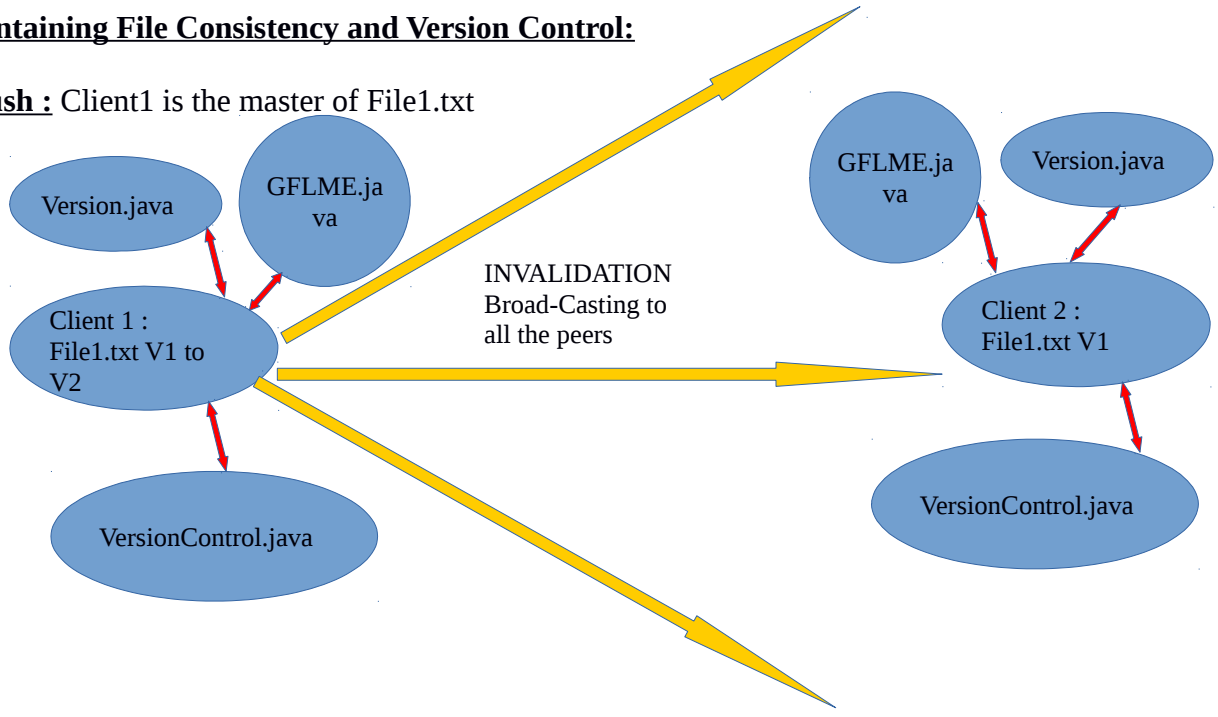



**2) BUS :**



**Maintaining File Consistency and Version Control:**

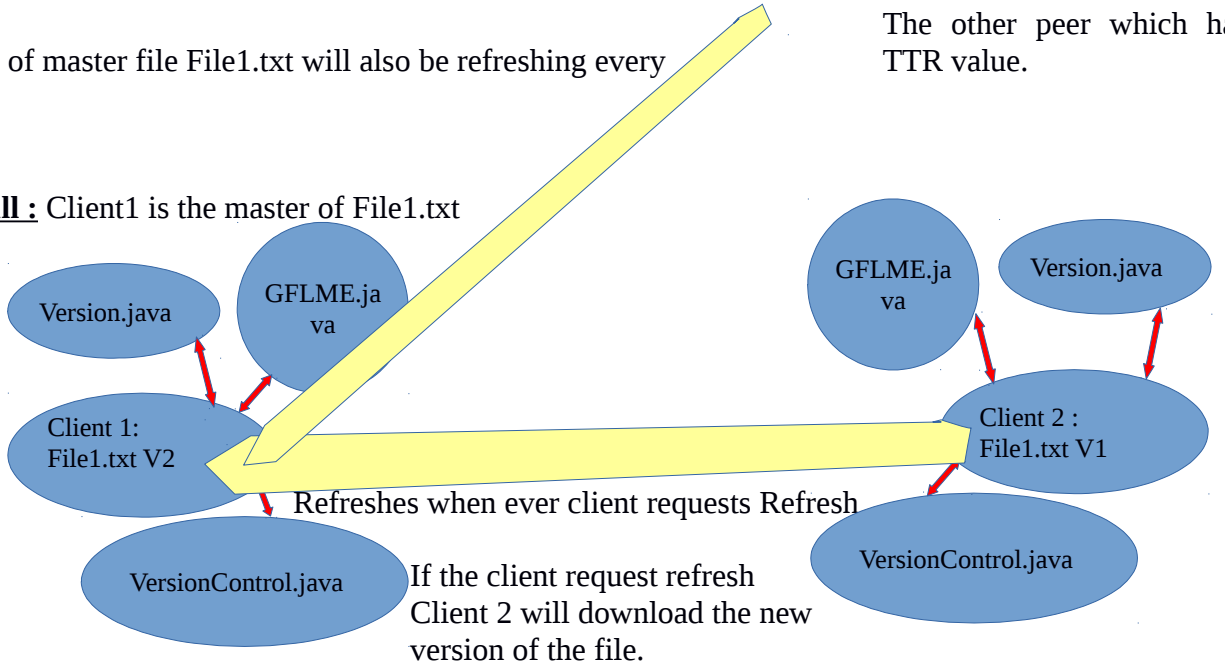
**1)Push :** Client1 is the master of File1.txt




When Client1 make any changes to the master copy of the File File1.txt then it broadcast INVALIDATION notification to all the peers which has a copy of the file. On receiving the INVALIDATION notification the peers make the copy of the file File1.txt to be invalid.

copy of master file File1.txt will also be refreshing every The other peer which have TTR value.

**2)Pull :** Client1 is the master of File1.txt




A peer will periodically checks its local store and poll the server for those copies whose TTR expired. And the server in reply will send a new TTR if the file is not modified since. Or a negative reply if the file has been modified. The client will mark its copy invalid if the server reply is negative. Otherwise the client updates the file's TTR and proceeds. The client side should provide users an option to "refresh" an outdated file, which downloads a new copy of the file.

**New implementation :**

Already existing file got modified and we added three new java files i.e. Version.java, versionControl.java, and GetFileLastModifiedExample.java (GFLME.java)

**Flow logic: The above diagram demostrate the flow logic**

- First we create number of clients/peers and assign them some port numbers, then we invoke peerClientThread thread which act as a client which initiate a query request, for searching for a particular file, from client x using Query.java class object and send to all client x neighbors.
- Query request message is propagated from peer to peer until is finds the peer containing the requested file.
- All peers containing the file requested by peer client x respond with Query hit message, in the above diagram client y has the requested file, it generate the Query object and response with back tracking using query hit message.
- When the requesting client receives the query hit message from a peer or set of peers(in this case it select the peer from the list of peer containing the same file) then it establish a connection with the particular peer which has the file and download the file.
- We manage size of this associative array our peer needs to maintain, and flush out old entries at appropriate times(user defined) using the updateThread thread.

**Client side :**

**QueryType.java :** enum QueryType having some values related to the type of query(Request, Response, invalid, Refress or Download) each Peer initiates.

**Client.java :** This class has the Client side main() function, which intern class updateThread Class object to keep checking the status of the files present at the Client associated and update automatically when there is a change in the file system. It also create peerClientThread and peerServerThread class object to take and process the client request.

- Loadfile() : This function will load the file acoording to the version number of a particular file.

**Query.java :** This class have all information for a query request, response and download. This class implements JAVA serializable and it has following functions :-

- public final String getIpAddr() : Returns IP-Address InetAddress object which is set by void setIpAddr(InetAddress ipAddr) function.
- public final void setIpAddr(String ipAddr) : Set InetAddress object IP-Address using the passed parameter ipAddr.
- public final int getPort() : Returns the port number.
- public final void setPort(int port) : Sets the port number to the port value passed to the function.
- public final int getMessageID() : Returns the MessageID.
- public final void setMessageID(int messageID) : Sets the MessageID.
- public final String getFileName() : Return the name of the filename set by void setFileName(String fileName) function.
- public final void setFileName(String fileName) : Set fileName
- public final QueryType getQueryType() : Return RequestType enum value identifying the type of request made by peers.
- public final void setQueryType(QueryType queryType) : Set the RequestType enum value identifying the type of request made by peers.
- public Query( Query query) : Constructor to initialize local variable
- double getVersion() : Return the version number of a particular file
- void setVersion(double Version) : sets the version number of a particular file
- int getTTR() : Returns the Time to Refresh value.
- void setTTR(int tTR): Sets Time to Refresh value
- String getAuthor() : Return the Author of a particular file
- void setAuthor(String author): Set the Author of a particular file

**PeerClientThread.java :** This class take the client requestType and calls related functions :-

- run() : This function accept the client request and call the respective functions.
- void RequestPeerList() : This function send request to all peers asking for a peer list which contain a particular file send with the request.
- void DownloadFile():This function download a particular requested file from other peers.
- void sendObject2AllPeer(Query query): This function send Query object to all the peers.
- void sendObject2Peer(Query query): This function send Query object to a particular peers.
- int receiveFile(Query query): This function receive the file and its content from the client which has the requested file.

**PeerServerThread.java :** This class is used to make client act as a server by communicating with other client member in the connected cluster. It has following functions :-

- run() : This function accept the filename requested from requesting client and then process on the file depending upon the requestType parameter.
- sendRequestFile(Query query): This function return the file to the requesting client. It takes Query object to find the QueryType and related information
- sendResponse(Query query) : This function return Query object with QueryType Response if the requested file is present in a particular client handling request.
- void sendFile(File file): This function reads the file locality and send the file after converting it in to bytes to the requesting client-side.
- void readRequestObject() : This function read the object related to the communication between the requesting client and processing client.
- sendObject2Peer(Query query) : This function sends the response with the Query object to the requesting Peer
- void refress(Query query1) : This function check the version number of a file and download the file from the master if the file is invalid
- void inValidate(Query query) : This function marks a file invalid if required.

**Npeers.java :** This class have all information for Request. It has following functions :-

- String getIpAddr(): This function returns IP-Address InetAddress object which is set by void setIpAddr(InetAddress ipAddr) function.
- void setIpAddr(String ipAddr): This function set InetAddress object IP-Address using the passed parameter ipAddr.
- int getPort(): Returns the port number.
- void setPort(int port) : Sets the port number to the port value passed to the function.

**UpdateThread.java :** This class is used to flush out old entries at appropriate times from a associated array.

**GetFileLastModifiedExample:** This class checks and prints when a particular file was late modified.

**Version.Java:** This class have all information for Version Control Request. It implements Serializable and has following functions :-

- getIpAddr(): Returns IP-Address InetAddress object which is set by void setIpAddr(InetAddress ipAddr) function.
- setIpAddr(String ipAddr) : et InetAddress object IP-Address using the passed parameter ipAddr.
- int getPort() : Returns the port number.
- void setPort(int port) : Sets the port number to the port value passed to the function.
- String getAuthor() : Return the author of a particular file
- void setAuthor(String author) : Set author of a particular filename
- String getFileName() : Return the name of the filename set by void setFileName(String fileName) function.
- void setFileName(String fileName) : Set fileName
- public Version( Version version) : Parameterized Constructor
- int getCTTR(): Returns counter for TTR.
- void setCTTR(int Cttr) : Sets counter for TTR
- int getTTR() : Returns TTL in sec

- void setTTR(int tTR) : Sets TTL in sec
- long getTime() : Returns last modified time of a file.
- void setTime(long time) : Sets last modified time of a file.
- long getValid() : Returns 0 or 1 indicating whether a particular file is valid or not .
- void setValid(int valid) : Sets 0 or 1 indicating whether a particular file is valid or not.
- double getVersion() : Return the version number of a particular file.
- void setVersion(double version) : Set the version number of the file when modified.

**versionControl.Java:** This class invoke its constructor when its object is created, a value one second is passed for checking the status of the file at the client side whether it has undergone any change. If it has undergone any change then it update the file status at the server side as well. This class has following functions:-

- RemindTask2 extends TimerTask : void run(): this class runs in every second set by client
- void vRefress() : The functions checks the files and there's contents to refresh the version number of the file and the file it self if the particular client as the invalid file.
- void vControl() : This functions checks the version of the files and update it when required.
- void invalidateSlaves(String fileName,double ver) : It invalidate the file refered by the filename parameter and invalidate it if it has a old version of the file.

**Files :**

- **Input_files :** This folder contains all the documented files at the client side and this folder is accessible by the Java classes at the client side
- **Output_files :** This folder contains the requested file after receiving it from other peers.