Advanced Computer Networks

Fall 2013 Semester Project : SCALE-FREE TOPOLOGY GENERATION

Instructor: Dr Kamil Sarac

Team Members :
Anchal Katyal
Vasim Mansuri
Vivek Wadhwani

Project Description:

The basic aim is to generated a network topology and let the new node be connected to the initial m_0 such that it mimics the ideation of Barabasi Model.
We have create a 3 node topology for the initial network, where node 1 and 3 are connected to node 2. The nodes are denoted by M0Node1, M0Node2 & M0Node3. in the project file. Each initial m_0 has its own config file giving the node number, IP address of that node, TCP port number, UDP port number, UDP Port number for query, network max and current size and the values for m_0 nodes(m_0 = 3) and maximum edges a new node gets connected to i.e. m(m=2).
All node perform a periodic broadcast of a file containing the connectivity information of all nodes in the network, for eg. Node 1 is connected to Node 2 & Node 3 is connected to Node 2, with their corresponding IP address and TCP port numbers.

The degree information is computed by extracting the conectivity information of these nodes from the Routing information file. The routing information file gives the IP addresses and port numbers to which each of the the nodes are connected to.

When a new node arrives, it sends a join request to any of the initial m_0 in a UDP packet. The m_0 node say M0Node2 recieves this packet and identifies the new node's port and IP address. If the networkmaxsize is less than 6 it sends a UDP packet with its own TCP port number saying it can accept that node. The new node then sends a join request on this TCP port. The new node gets connected to the queried m_0 node. After this connection, the new node gets a UpdatedRoutingInformation file from m_0 node.

The findDegree method extracts the degree information of each node and stores it an array list 'degreeInformation'.

When new node calculates probability the m_0 sends UpdatedRoutingInformation file to the new node.

The findprobability runs a random function and finds cumulative probability of the network. If the generated random number is less than cumulative probability and not null, then it sends the IP address and TCP port number of that corresponding node back to the new node. The new node sends a TCP connection request on this received TCP port number to get connected to the that node. Thus, in this manner a new node gets connected to 2 other nodes in the network as we have set m=2(maximum number of edges). The network will continue to accept the requests and establish the connections until the maximum network size is reached i.e. 6.

Compilation Steps :

For running the code--->

M0Node file : M0Node1.java
M0Node2 file : M0Node2.java
M0Node3 file : M0Node3.java
NewNode file : NewNode.java
Congif1.properties
Congif2.properties
Congif3.properties
RoutingInfo.txt —> Creates and updates for each node

On M0Node machine:

```
javac M0Node.java    -->for compilation
```

On M0Node2 machine:

```
javac M0Node2.java    -->for compilation
```

On M0Node3 machine:

```
javac M0Node3.java    -->for compilation
```

On NewNode machine:

```
javac NewNode.java    -->for compilation
```

On M0Node Machine -- Start server:

```
java M0Node
```

On M0Node2 Machine -- Start server:

```
java M0Node2
```

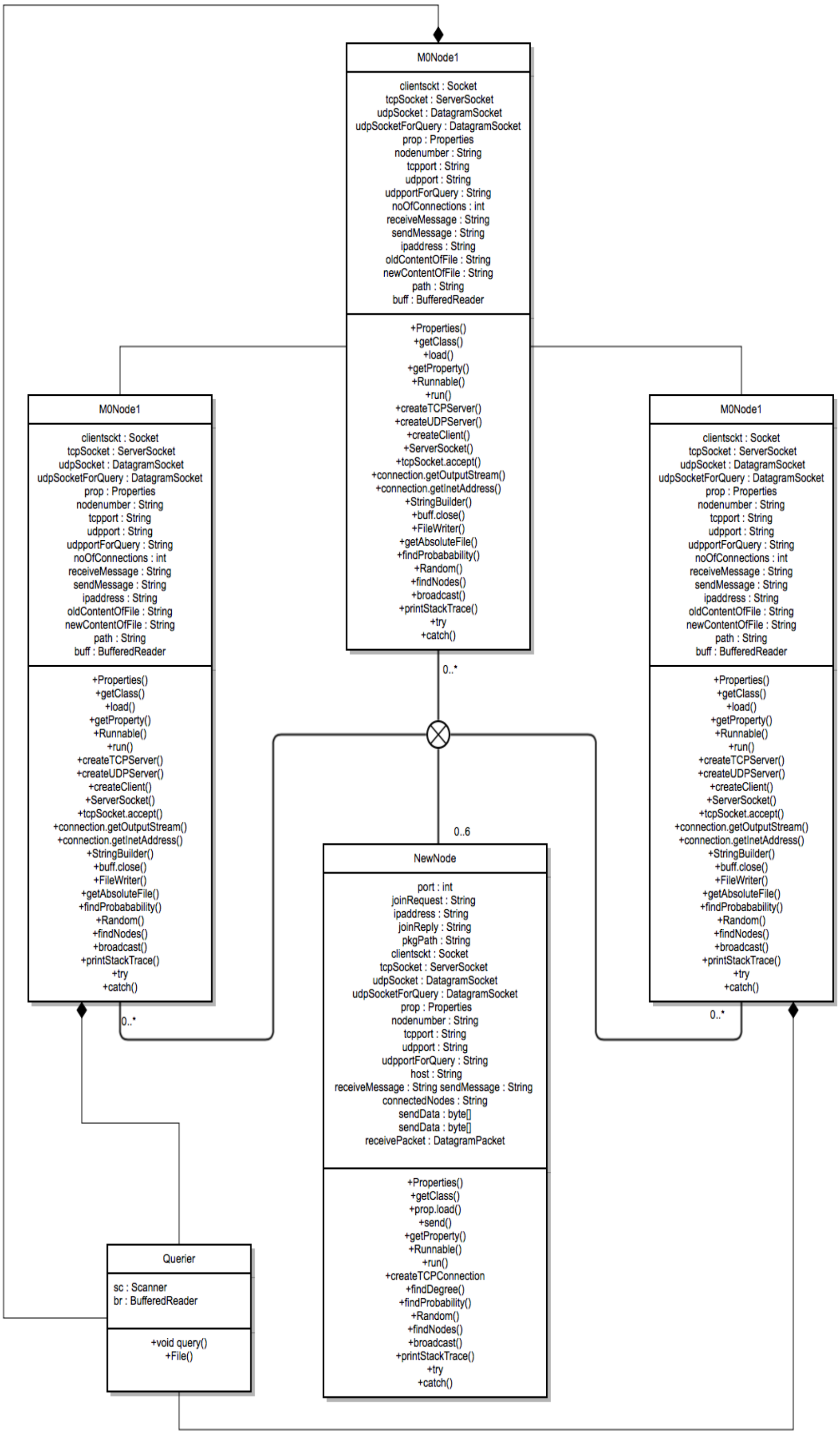
On M0Node3 Machine -- Start server:

```
java M0Node3
```

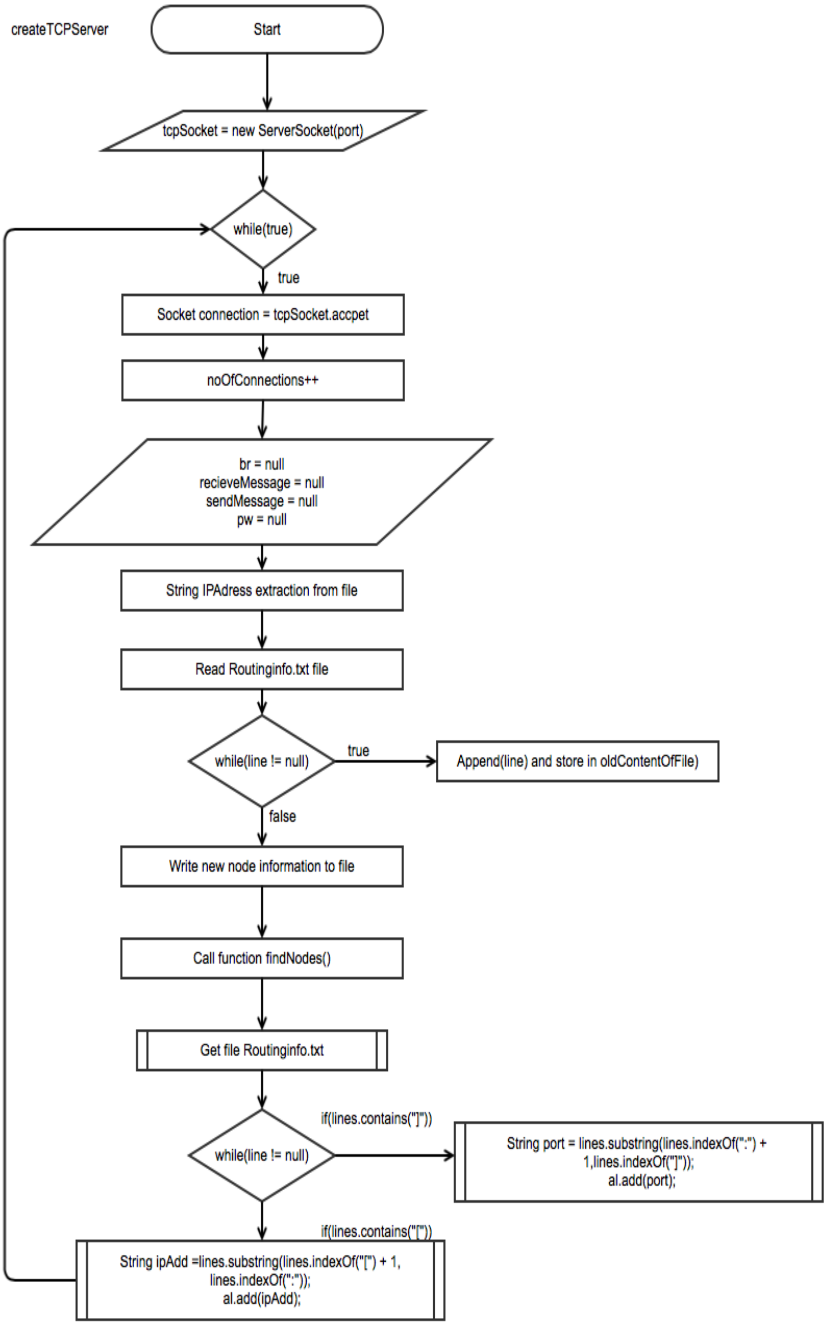
On NewNode Machine -- Start server:

```
java NewNode
```

Class Diagram :



Flowchart for createTCPServer:


createTCPServer
Start
tcpSocket = new ServerSocket(port)
while(true)
true
Socket connection = tcpSocket.accpet
noOfConnections++
br = null
recieveMessage = null
sendMessage = null
pw = null
String IPAdress extraction from file
Read Routinginfo.txt file
while(line != null)
true
Append(line) and store in oldContentOfFile)
false
Write new node information to file
Call function findNodes()
Get file Routinginfo.txt
while(line != null)
if(lines.contains("]"))
String port = lines.substring(lines.indexOf(":") + 1,lines.indexOf("]"));
al.add(port);
if(lines.contains("["))
String ipAdd =lines.substring(lines.indexOf("[") + 1, lines.indexOf(":"));
al.add(ipAdd);

Work Distribution :

	Designer	Ideation	Coder	Tester	Documentation
Anchal	✔	✔	✔		✔
Vasim	✔	✔	✔	✔	
Vivek	✔	✔	✔		✔