Improved Gnutella Algorithm

Martin Minovski, University of Aberdeen

I. Introduction

The purpose of this project is to propose, describe and implement an improved solution for querying and retrieving files from a Peer-To-Peer network based on the popular Gnutella protocol.

II. Brief description of the system

The proposed peer-to-peer system introduces 3 different types of peer roles:

a. Host caches – they hold information about current peers in the network
b. Super-nodes – they maintain a list of directly connected normal nodes and answer directly to queries without querying the normal nodes.
c. Normal nodes – they stay connected to a super-peer

III. Node behaviour

1. Pseudocode and notes

Important notes:

- Instead of IP and Port, an ID that combines those in itself also including Name is used throughout the pseudocode.
- If supernode A has supernode B as neighbour, B is not required to add A back as neighbour. This is done so that supernodes with higher bandwidth can be referenced as neighbour by more than 4 supernodes. The host cache determines which supernode to pick as first neighbour for a newly connected supernode. That way, a better load balancing can be achieved in the network.
- Host caches use bandwidth - connected nodes ratio. The more available bandwidth a supernode has (which depends on total bandwidth and currently connected nodes / neighbouring supernodes), the bigger the chance that a host cache will pick that supernode for someone else's neighbour or as a "parenting" supernode for a normal node.

Host Caches

Overall behaviour of a Host Cache:
Listens for messages and reacts to requests made by nodes.
Determines whether a node is a supernode or normal node
Introduces the node to the network and sends back:
a) A supernode to connect to if the requesting node is classified as a normal node
OR
b) Another supernode to connect to as neighbour if the requesting node is classified as a supernode
Keeps track of the current nodes in the network
Updates other host caches when new nodes enter the network

Overall Algorithm

```
begin
      IntroduceToNetwork (add listener)
      UpdateRecordFromOtherHostCache (add listener)
end
```

Algorithm IntroduceToNetwork

```
begin
      Nodes ← {}
      SuperNodes ← {}
      while true
      begin
            receive (request, Sender, MyID, {'CON_REQ', BANDWIDTH} )
            NeighbourSuperNode ← selectSome(SuperNodes)
            if  % check if it's a “returning customer” - check if it exists in Nodes or Supernodes with online flag = 0
                  set this flag back to 1
            else if BW > 2Mb then
                  SuperNodes ← SuperNodes + {(ID,BW,NeighbourSuperNode)}
            else
                  Nodes ← Nodes + {(ID,BW,NeighbourSuperNode)}  // Here NeighbourSuperNode is actually the "parent" SuperNode for that node
            send (inform,HC,Sender,(SomeSuperNodes)
            UpdateOtherHostCaches(Sender, BADNWIDTH, NeighbourSuperNode)
      end
end
```

Algorithm to select a super node to ping / register to as normal node

Algorithm selectSome(SuperNodes)

```
begin
      SuperNode <- the supernode from SuperNodes with highest bandwidth/already connected nodes ratio
      return SuperNode
end
```

Algorithm to update other host caches when a new node is connected

Algorithm UpdateOtherHostCaches

```
inputs ID, BW, NeighbourSuperNode
begin
      HostCaches ← { {HC1, ID}, {HC2, ID}, ... }          Those are hardcoded in every node
      foreach HC ∈ HostCaches
            send(inform, MyID, HC, {NEW_NODE, ID, BW, NeighbourSuperNode})
      end
end
```

```
Algorithm to update the HC's records if a node has connected via another HC
Similar to IntroduceToNetwork, but needed to avoid HC inform flood
```

Algorithm UpdateRecordFromOtherHostCache

```
begin
	Nodes ← {}
	SuperNodes ← {}
	while true
	begin
		receive (request, Sender, MyID, {'NEW_NODE', NewNodeID, Bandwidth, NeighbourID} )
		if  % check if it's a “returning customer” - check if it exists in Nodes or
Supernodes with online flag = 0
			set this flag back to 1
		else if BW > 2Mb then
			SuperNodes ← SuperNodes + {(NewNodeID,Bandwidth,NeighbourID)}
		else
			Nodes ← Nodes + {(NewNodeID,Bandwidth,NeighbourID)}
	end
end
```

Normal Nodes

Overall behaviour of normal nodes:

1. Connect to a HostCache and get a SuperNode to connect to
2. Register with a HostCache and submit the files available to it
3. Request the files it’s looking for and listen for file hits; store files in table
4. Continuously provide access to the files it has

Overall algorithm for normal nodes:

```
begin
	FilesToRequest <- { File1, File2, File3, …}  Those are set according to args
	MyFiles <- { MyFile1, MyFile2, MyFile3, … }  Ditto
	MySuperNode <- RegisterWithSuperNode( TryConnect() ) *
	RequestFiles(MySuperNode, FilesToRequest)
	while true
		receiveB (request, Sender, OwnID, {‘FILE_REQ’, Filename, {Path} } )
		if Filename exists in MyFiles
			File <- file data (in this case, just the filename)
			send (inform, OwnID, Sender, {‘FILE_HIT’, File, {Path}} )
		end
	end
end
```

* The chained call RegisterWithSuperNode(TryConnect()) ensures that it can pick another “parent” SuperNode if the one selected by the HC in TryConnect is busy.

Algorithm to try and connect to a supernode

Algorithm TryConnect

```
begin
	HostCaches <- { {HC1, HC1ID}, {HC2, HC2ID}, {HC3, HC3ID}, ... }		Those are
hardcoded
	foreach HC in HostCaches
		send(request, OwnID, HCID, {OwnBandwidth})
	end
	receiveB (inform, Sender, OwnID, {'CON_ACK', 'SER', SuperNodeID)
	return SuperNodeID
end
```

Algorithm to register with the SuperNode and send list of files

Algorithm RegisterWithSuperNode

```
inputs SuperNodeId
begin
	send ( request, OwnID, SuperNodeID, {'REG', {Files}})
	receive ( inform, SuperNodeID, OwnID, {MSG} )
	if MSG is "NACK" or timeout
		return RegisterWithSuperNode( TryConnect() )  //call TryConnect again to pick
another SuperNode
	else
		return SuperNodeID
	end
end
```

Algorithm to request all files to the supernode and listen for file hits

Algorithm RequestFiles

```
inputs SuperNodeID, {FileList}
begin
	ReceivedFiles <- {}
	TTL <- 3		Defined by runtime args
	foreach File in {FileList}
		send ( request, OwnId, SuperNodeID, { 'FILE_REQ', Filename, TTL, {Path: OwnID} }
		receiveNB ( request, SuperNodeID, OwnId, {'FILE_HIT', File, {Path: OwnID} }
			Path containing only OwnID means that file hit is for this node
		if received
			add File to ReceivedFiles to be added to MyFiles list
	end
end
```

SuperNodes

Overall behaviour:

1. Connect to HostCache and obtain a supernode to connect to
2. Try to connect to a supernode
3. If successful, send ping, if not, refer back to HostCache
4. Listen for pongs coming back from the supernode and then connect directly to ponging supernodes
5. Listen for registering nodes
6. Start searching for files by sending flooded requests
7. Listen for file hits and add to file table
8. Listen for pings/pongs/file requests/file hits that are not for self.

NOTE:

Algorithm OverallBehaviour

```
begin
	FilesToRequest <- { File1, File2, File3, …}  // Those are set according to args
	MyFiles <- { MyFile1, MyFile2, MyFile3, … }  // Ditto
	NormalNodesFiles <- {}	// This is populated as normal nodes connect to this supernode
		// NormalNodesFiles is data structure in format {node: {files}}
	Neighbour <- TryConnect()
	AllNeighbours <- {}
	Ping( Neighbour )
	ListenForPings(AllNeighbours)
	ListenForPongs(AllNeighbours)
	wait some time for pongs to arrive
	ListenForRegisteringNodes(NormalNodesFiles)
	add RequestFiles(AllNeighbours, FilesToRequest) to MyFiles
	ListenForFileRequests(MyFiles, NormalNodesFiles, AllNeighbours)
	ListenForFileHits	// This listens for file hits not intended for this supernode

end
```

Algorithm TryConnect

```
begin
	HostCaches <- { {HC1, HC1ID}, {HC2, HC2ID}, {HC3, HC3ID}, ... }	Those are hardcoded
	foreach HC in HostCaches
		send(request, OwnID, HCID, {OwnBandwidth})
	end
	receiveB (inform, Sender, OwnID, {'CON_ACK', 'SUP', NeighbourID)
	return NeighbourID
end
```

Algorithm Ping

```
inputs NodeToPing
begin
	send ( request, OwnID, NodeToPing, {‘PING’, TTL=3, PATH={OwnID} } )
end
```

Algorithm ListenForPings

```
inputs Neighbours
begin
	while true
		receiveNB (request, Sender, OwnID, {‘PING’, TTL, PATH = {ORIGINATOR,{HOPS} } } )
		send ( inform, OwnID, Sender, {‘PONG’, SENDER=OwnID, PATH} )
		TTL = TTL - 1
		if TTL is not 0
			foreach Neighbour in Neighbours excluding Sender
				send ( request, OwnID, Neighbour,
					{‘PING’, TTL, {ORIGINATOR, {HOPS + OwnID} } } )
				// Broadcast the ping to neighbours
			end
		end
end
```

Algorithm ListenForPongs

```
inputs pointer to AllNeighbours
begin
	while true
		receiveNB ( inform, Sender, OwnID, {‘PONG’, SENDER, PATH= {ORIGINATOR, {HOPS, LASTHOP}}} )
		if ORIGINATOR = OwnID // then pong is intended for me
			add SENDER to AllNeighbours
		else
			// forward the pong to the last hop in chain:
			send ( inform, OwnID, LASTHOP, {‘PONG’, SENDER, PATH= {ORIGINATOR, {HOPS}}} )
		end
	end
end
```

Algorithm RequestFiles

```
inputs {Neighbours}, {FileList}
begin
	ReceivedFiles <- {}
	TTL <- 3		Defined by runtime args
	foreach File in {FileList}
		foreach Neighbour in Neighbours
			send ( request, OwnId, Neighbour, { 'FILE_REQ', Filename, TTL, {Path: OwnID} }
			receiveNB ( request, Neighbour, OwnId, {'FILE_HIT', File, {Path: OwnID} }
				Path containing only OwnID means that file hit is for this node
			if received
				add File to ReceivedFiles to be added to MyFiles list
	end
	return ReceivedFiles
end
```

Algorithm ListenForFileRequests

```
inputs MyFiles, NormalNodesFiles, Neighbours
begin
	while true
		receiveNB ( request, Sender, OwnID, {‘FILE_REQ’, FILENAME, TTL, PATH= {ORIGINATOR, {HOPS, LASTHOP}}} )
		if FILENAME is in MyFiles
			FILE <- FILENAME // In reality should be the file itself
			send ( inform, OwnID, Sender, {‘FILE_HIT’, FILE, PATH} )
		else
			// Now check if file is one of the directly connected normal nodes
			if FILENAME is in {NormalNode:{Files}} of NormalNodesFiles
				send ( request, OwnID, NormalNode, {‘FILE_REQ’, FILENAME, PATH + OwnID } )
			else
				foreach Neighbour in Neighbours except for Sender
					send ( request, OwnID, Neighbour, {‘FILE_REQ’, FILENAME, PATH + OwnID } )
				end
			end
		end
	end
end
```

Algorithm ListenForFileHits

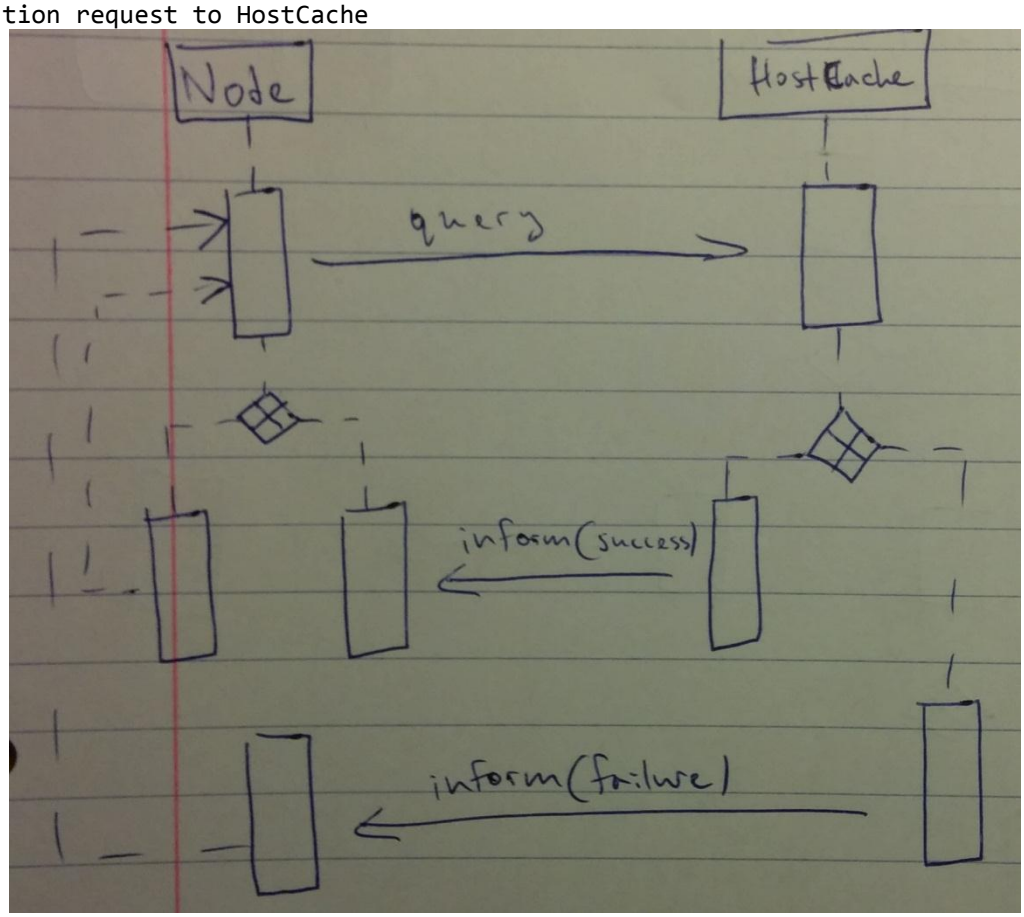
```
begin
	while true
		receiveNB ( inform, Sender, OwnID, {‘FILE_HIT’, FILE, PATH= {ORIGINATOR, {HOPS, LASTHOP}}} )
		// NOTE: File hits intended for this supernode are handled in RequestFiles;
		// This algorithm is only for forwarding file hits
		send ( inform, OwnID, LASTHOP, {‘FILE_HIT’, FILE, PATH= {ORIGINATOR, {HOPS}}} )
	end
end
```

Algorithm ListenForRegisteringNodes

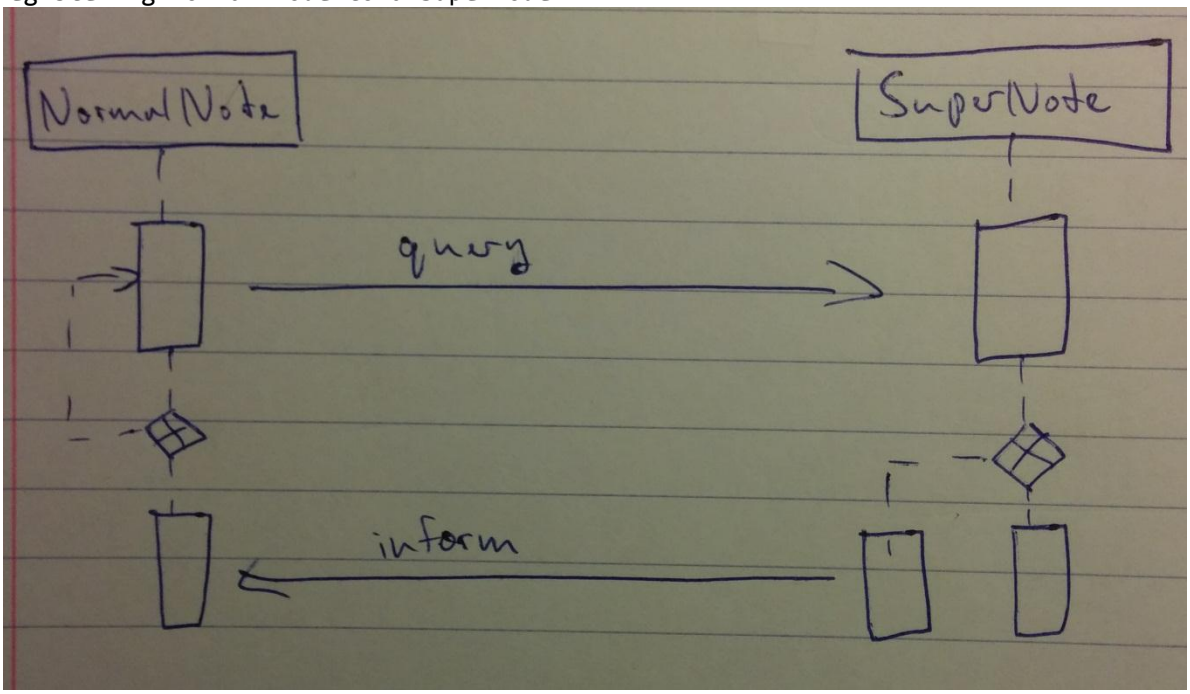
```
inputs pointer to NormalNodeFiles
begin
	while true
	receiveNB( inform, Sender, OwnID, {‘REG’, FILES }
	add {Sender:{FILES}} to NormalNodeFiles
end
```

Agent UML

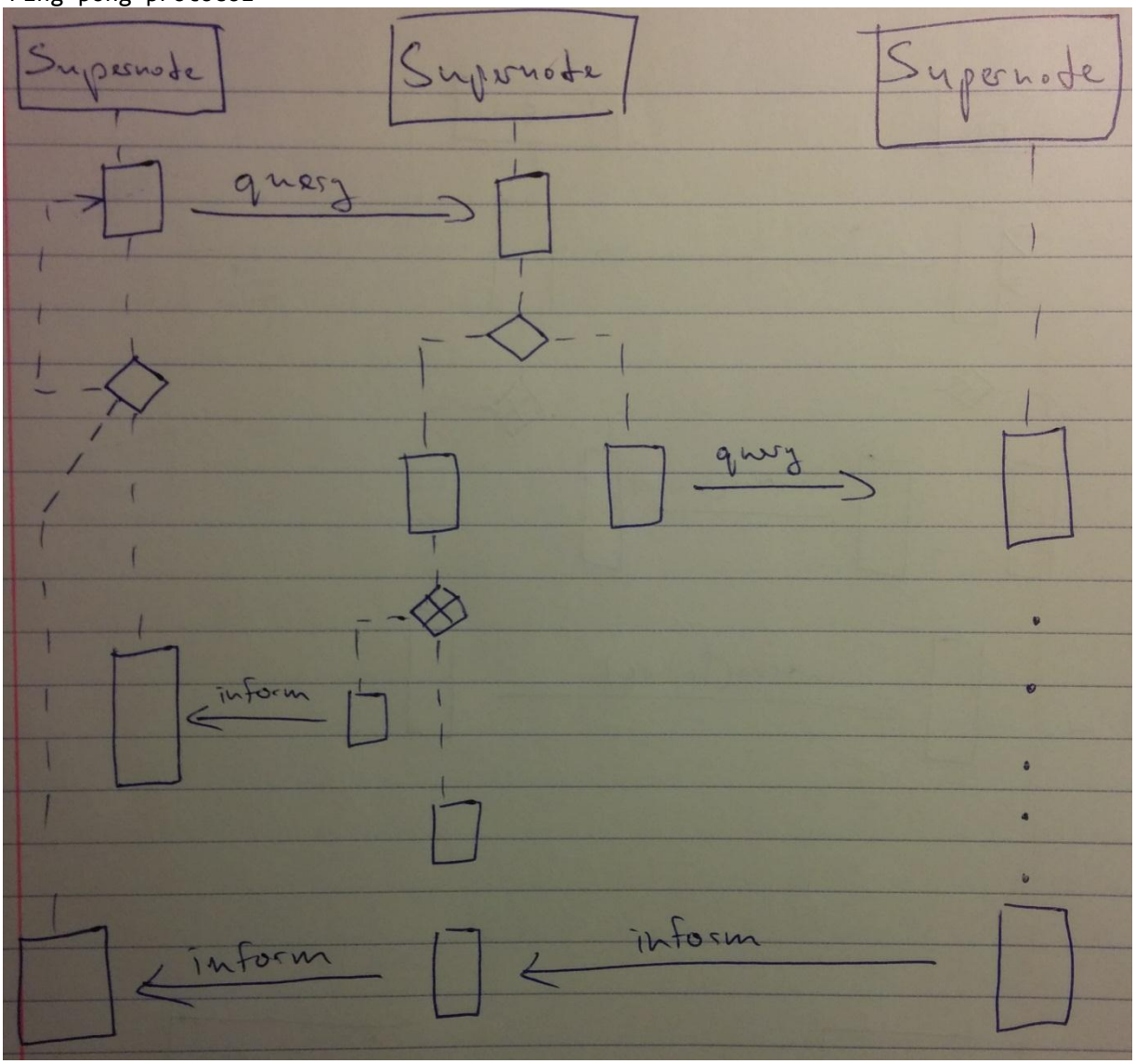
1. Connection request to HostCache



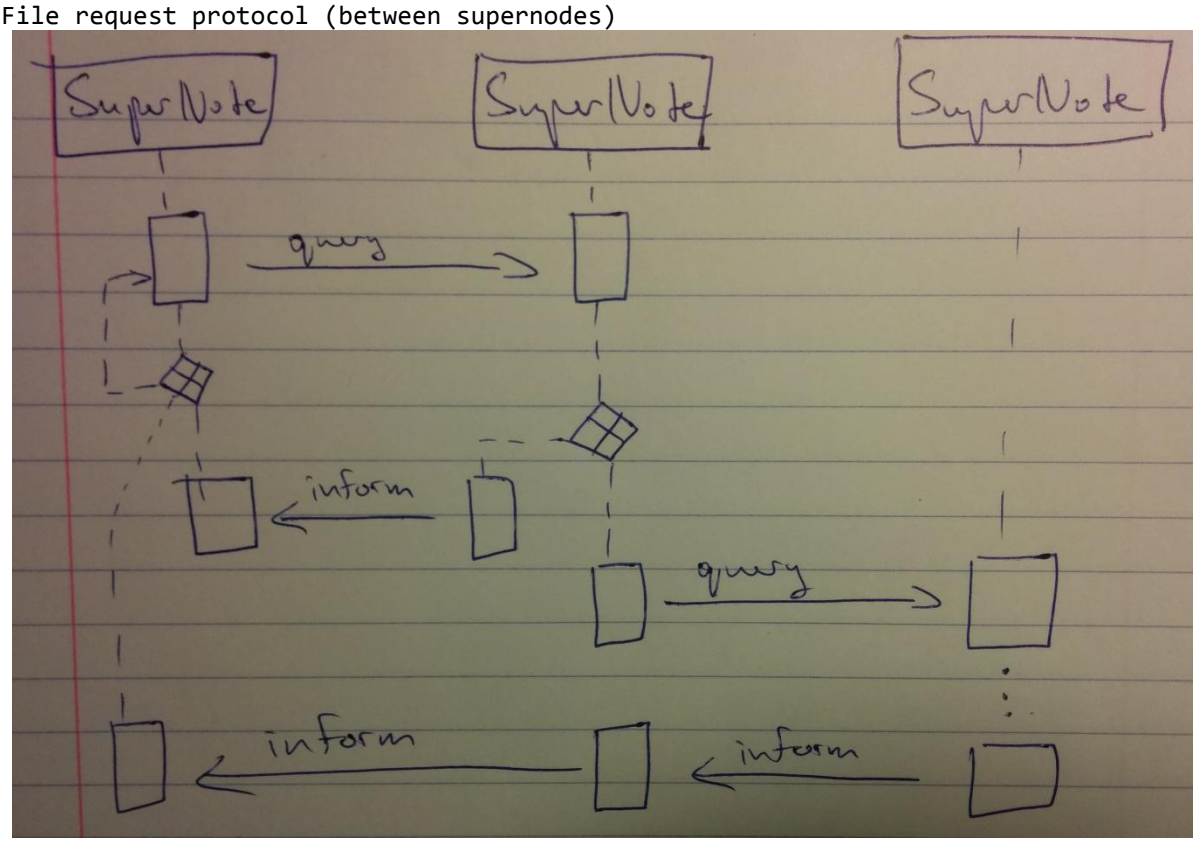
2. Registering normal node to a SuperNode



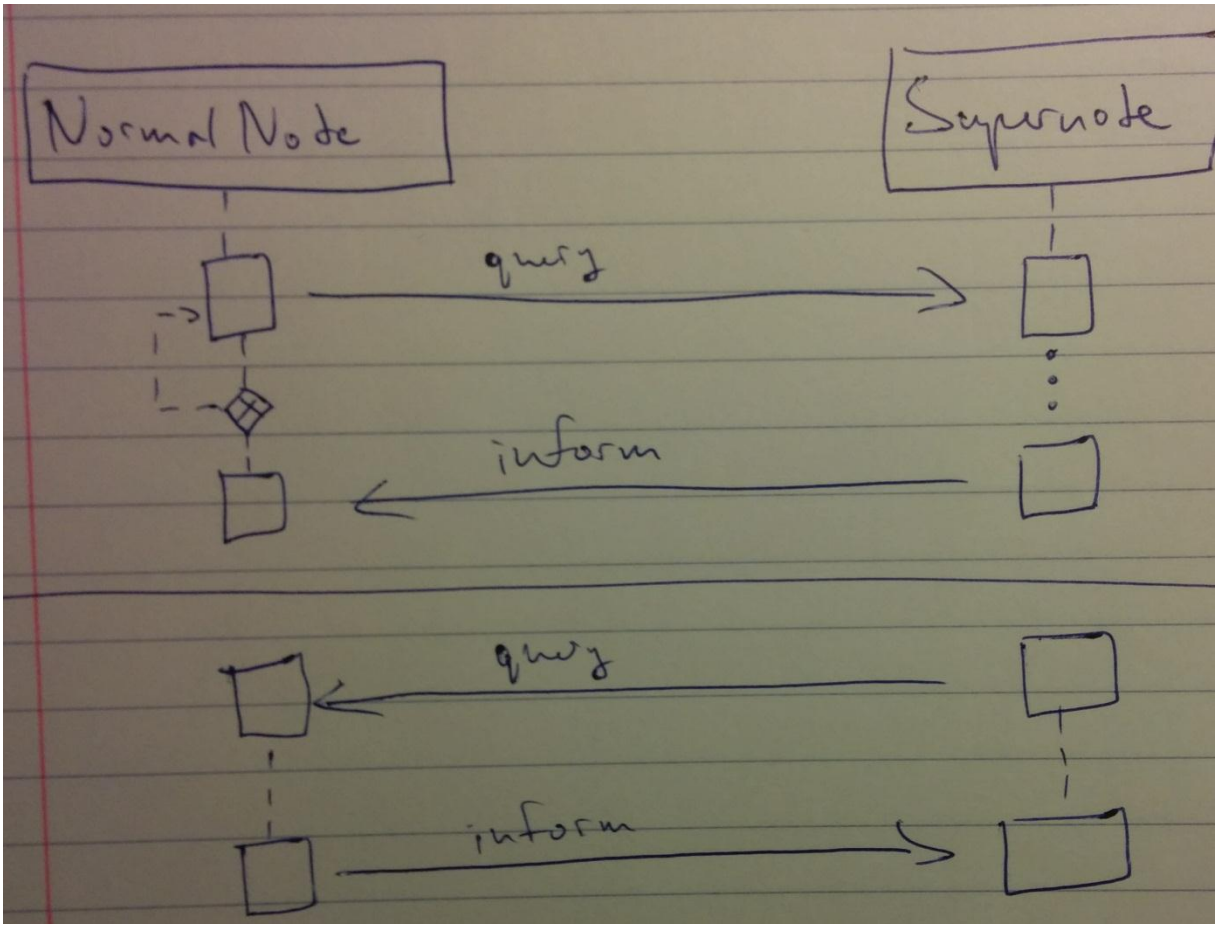
3. Ping-pong protocol



4. File request protocol (between supernodes)



5. File request between supernode and node



^ Here 2 simple cases are combined in 1 AUML diagram – when normal node requests a file and super node initiates the request.

Message format

General message format is:
Directive, Sender, Recipient, {'ACTION', {ARGUMENTS} }

Connect request (query):
- Sent from any Node to HostCache
- **ACTION**: CON_REQ; **ARGUMENTS**: {BANDWIDTH}

Connect response (inform):
- Sent from HostCache to any Node
- When connection has been acknowledged:
  ACTION: CON_ACK; **ARGUMENTS**:
  i. {'SER', SuperNodeID}
  ii. {'SUP, SuperNodeID}
  iii. {'ONLY_SUP'}
- When connection has been declined:
  ACTION: CON_REF; **ARGUMENTS**: {'NO_SUP'}

Network update message (inform):
- Sent between HostCaches
- When newly connected node is SuperNode:
  ACTION: NEW_NODE; **ARGUMENTS**: {NodeID, 'SUP', NeighbourID}
- When newly connected node is Normal Node (a.k.a. servant):
  ACTION: NEW_NODE; **ARGUMENTS**: {NodeID, 'SER', ParentID}

Normal node registering to supernode message (query)
- **ACTION**: REG; **ARGUMENTS**: {Files={file1,file2,file3…}}

Ping message (query):
- Sent between supernodes
- **ACTION**: PING; **ARGUMENTS**: {TTL, PATH={ORIGINATOR,HOP,HOP…} }

Pong message (inform):
- Sent between supernodes as response to Ping
- **ACTION**: PONG; **ARGUMENTS**: {PONGSENDER, PATH={ORIGINATOR,HOP,HOP..} }
- PONGSENDER or SENDER as written in pseudocode is needed to keep track of the node who sends the pong

File request (query):
- Sent to and from Normal nodes and Super nodes
- **ACTION**: FILE_REQ; **ARGUMENTS**: {FILENAME, TTL, PATH={ORIGINATOR, {HOP,HOP…} }

File hit (inform):
- Sent to and from Normal nodes and Super nodes
- **ACTION**: FILE_HIT; **ARGUMENTS**: {FILENAME, PATH={ORIGINATOR, {HOP,HOP…} }

Quit message from normal node to supernode (inform);
- **ACTION**: EXIT; **NO ARGUMENTS**

Implementation

The implementation of this agent-based system is a Java program that consists of three packages – agents, frames and services.

- The agents package contains 3 types of agents – ManagerNode (the operating agent that creates all other agents), HostCache and ClientNode (a hybrid ClientNode that takes the form of either NormalNode or SuperNode depending on response from the HostCache it is connecting to).
- The frames package is created for the purpose of visualising the final adjacency graph after running the simulation.
- The services package provides ready-made functionality to agents for ease of use such as message sending.

The program uses the JADE framework for agent communication. The usual agent behaviour used is Cyclic behaviour with blocking receive. In JADE this is not the same as the term blocking receive in agent theory where agents do not receive anything while blocked. Instead, JADE keeps a queue that keeps all messages that arrive while blocked and allows the agent to process them in the same order as they arrive. This is very helpful as using non-blocking receive would require us to isolate blocks of code and make the implementation much more complicated. For more complex tasks, non-blocking receive may be more useful, but the JADE version of blocking receive is a good choice for this assignment.

Instructions on how to run the program:

The JAR file needed to run the program is located in the “dist” folder. Run from command line:

Improved-Gnutella-Algorithm.jar –gui ManagerNode:agents.ManagerNode("input.txt","config.txt")

The files input.txt and config.txt must be in the same directory of the JAR file.

“input.txt” contains information about the peers in the network in the following format:
Name;Bandwidth ; file1,file2,file3 ; fileLookingFor1,fileLookingfor2
(tabs are added for clarity).

“config.txt” is the configuration of the system in the following format:

logHostCacheConv - Logs the messages sent to host caches as well as among them (for network update)
logDetailConversation - Logs the detailed conversation of pings/pongs between supernodes
logNeighbourTables - Logs the updated neighbour tables as nodes receive pongs
logFileExchange - Logs the file exchange messages
logFileReceive - Logs file receive messages (i.e. when nodes get the file they're looking for)
pingTTL - TTL value of pings
fileRequestTTL - TTL value of file requests
maxNeighbours - Max neighbour value for nodes
delayBetweenNodeCreate - Delay between node creation by the ManagerNode
displayIndividualStatistics - Display individual statistics of each node after simulation run

After the simulation, the topology of the network will be shown.
Also, the file output.txt will be created with the simulation results.
NOTE: The user must wait for the topology window to appear. After this, the output file is ready.

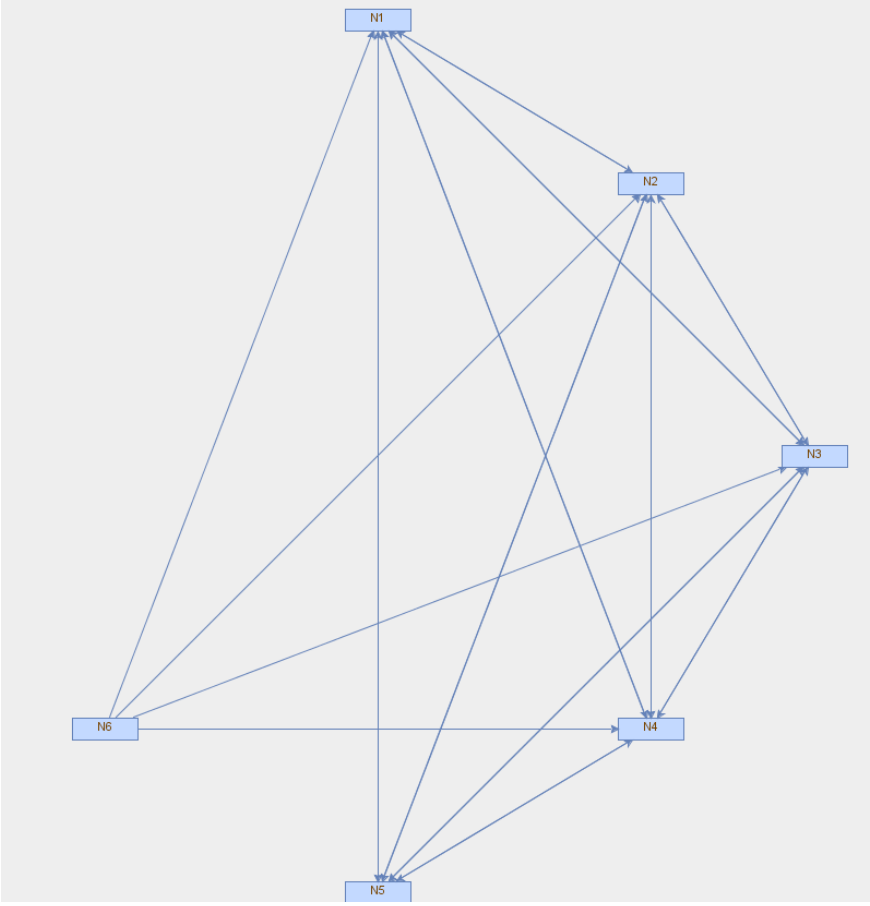
Simulation results

1. **Scenario 1 – one servent is the sole provider of a file which is wanted/needed by all other servents**

Run:
P2PAssignment.jar -gui ManagerNode:agents.ManagerNode("scen1.txt","config.txt")

In this scenario, s20sFile3 is held by S20, but requested by all other nodes.
S20 is a normal node and its supernode is N4.
From the output, we can observe that N1 obtains the file via N4, which is correct.
After this, N2 obtains it directly from N1 as it already has it from S20.
This way, no other node gets the file from S20 as it would be much more efficient to get it straight from the nodes that have obtained it previously in order to save bandwidth.
We can see that all nodes have received the file they were looking for without obtaining it from the same node every time.

Results:



Output:

```
N1: Received file s20sFile3. Hooray! Path:
N1@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE,S20@192.168.1.2:1099/JADE
N2: Received file s20sFile3. Hooray! Path: N2@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
N3: Received file s20sFile3. Hooray! Path: N3@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
N4: Received file s20sFile3. Hooray! Path: N4@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
N5: Received file s20sFile3. Hooray! Path: N5@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
N6: Received file s20sFile3. Hooray! Path: N6@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
S1: Received file s20sFile3. Hooray! Path: S1@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S2: Received file s20sFile3. Hooray! Path: S2@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
S3: Received file s20sFile3. Hooray! Path: S3@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
S4: Received file s20sFile3. Hooray! Path: S4@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE
S5: Received file s20sFile3. Hooray! Path: S5@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
```

```
S6: Received file s20sFile3. Hooray! Path: S6@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
S7: Received file s20sFile3. Hooray! Path: S7@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
S8: Received file s20sFile3. Hooray! Path: S8@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE
S9: Received file s20sFile3. Hooray! Path: S9@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE
S10: Received file s20sFile3. Hooray! Path: S10@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
S11: Received file s20sFile3. Hooray! Path: S11@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
S12: Received file s20sFile3. Hooray! Path: S12@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S13: Received file s20sFile3. Hooray! Path: S13@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S14: Received file s20sFile3. Hooray! Path: S14@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE
S15: Received file s20sFile3. Hooray! Path: S15@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S16: Received file s20sFile3. Hooray! Path: S16@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
S17: Received file s20sFile3. Hooray! Path: S17@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE
S18: Received file s20sFile3. Hooray! Path: S18@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
S19: Received file s20sFile3. Hooray! Path: S19@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
S20: Received file s6sFile1. Hooray! Path:
S20@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE,S6@192.168.1.2:1099
/JADE

N1 Statistics :
       13 pings received,
       10 pings forwarded,
       21 piongs forwarded,
       4 file requests initiated,
       19 file requests forwarded,
       4 file hits forwarded,
       15 file hits received.
       Total messages received: 95
N3 Statistics :
       9 pings received,
       9 pings forwarded,
       4 piongs forwarded,
       4 file requests initiated,
       7 file requests forwarded,
       19 file hits forwarded,
       6 file hits received.
       Total messages received: 70
N2 Statistics :
       11 pings received,
       10 pings forwarded,
       6 piongs forwarded,
       4 file requests initiated,
       11 file requests forwarded,
       10 file hits forwarded,
       12 file hits received.
       Total messages received: 72
S1 Statistics (my supernode is N5@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S3 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S6 Statistics (my supernode is N1@192.168.1.2:1099/JADE):
       0 pings received,
```

```
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       4 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 8
S5 Statistics (my supernode is N2@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S8 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S11 Statistics (my supernode is N1@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S16 Statistics (my supernode is N1@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S13 Statistics (my supernode is N5@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S10 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S14 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
```

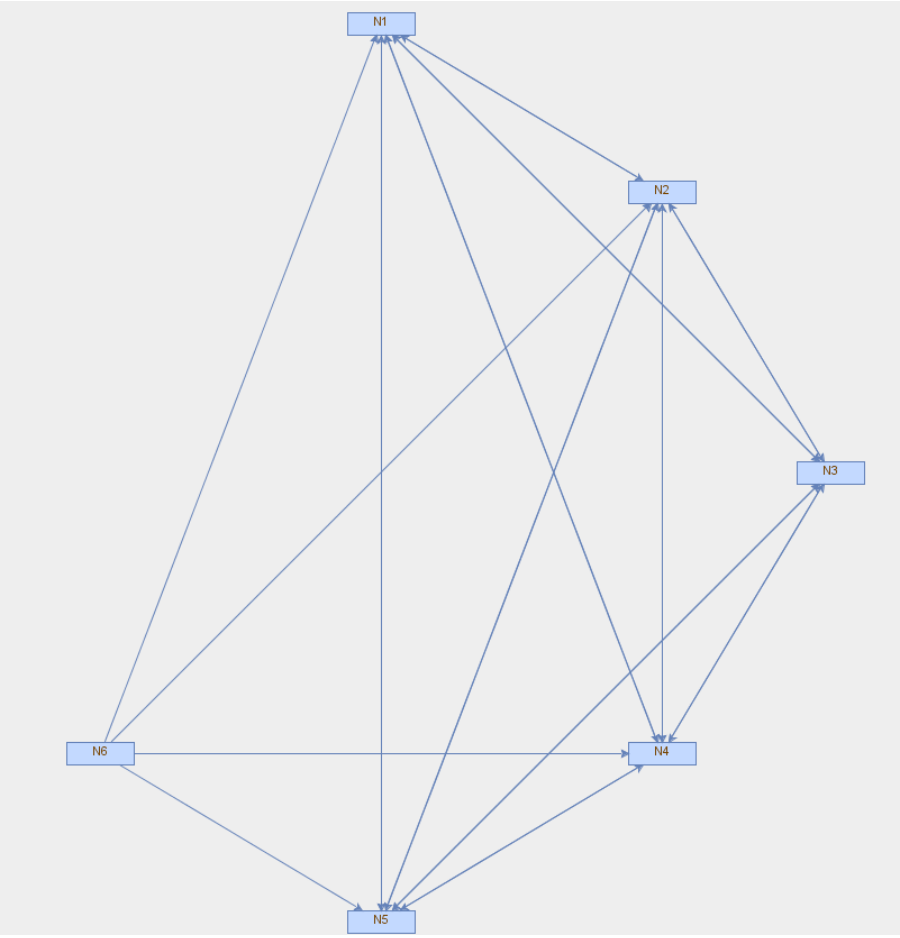
```
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S18 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S2 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S17 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S20 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       17 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
       Total messages received: 24
S4 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
S12 Statistics (my supernode is N5@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 4
```

2. **Scenario 2 – half the servants provide the files wanted by the other servants**

Run:
Run:
P2PAssignment.jar -gui ManagerNode:agents.ManagerNode("scen2.txt","config.txt")

Again, very good file hit rate can be observed from the results.
Average length of file hit path is significantly longer as we are dealing with more files compared to the previous experiment. Also, number of messages passed is also much bigger.

Results:



Output:

```
N1: Received file s7sFile3. Hooray! Path:
N1@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE,S7@192.168.1.2:1099/JADE
N2: Received file s8sFile3. Hooray! Path:
N2@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,S8@192.168.1.2:1099/JADE
N3: Received file s9sFile3. Hooray! Path:
N3@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE,S9@192.168.1.2:1099/JADE
N4: Received file s10sFile3. Hooray! Path:
N4@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE,S10@192.168.1.2:1099/JADE
N5: Received file s11sFile3. Hooray! Path:
N5@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE,S11@192.168.1.2:1099/JADE
N6: Received file s12sFile3. Hooray! Path:
N6@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE,S12@192.168.1.2:1099/JADE
S1: Received file s13sFile3. Hooray! Path:
S1@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE,S13@192.168.1.2:1099
/JADE
S2: Received file s14sFile3. Hooray! Path:
S2@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE,S14@192.168.1.2:1099
/JADE
```

```
S4: Received file s16sFile3. Hooray! Path:
S4@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,S16@192.168.1.2:1099/JADE
S6: Received file s18sFile3. Hooray! Path:
S6@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,S18@192.168.1.2:1099
/JADE
S7: Received file n1sFile3. Hooray! Path:
S7@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
S8: Received file n2sFile3. Hooray! Path:
S8@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
S9: Received file n3sFile3. Hooray! Path:
S9@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE
S10: Received file n4sFile3. Hooray! Path:
S10@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE,N4@192.168.1.2:1099/JADE
S11: Received file n5sFile3. Hooray! Path:
S11@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S14: Received file s2sFile3. Hooray! Path:
S14@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,S2@192.168.1.2:1099
/JADE
S15: Received file s3sFile3. Hooray! Path:
S15@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE,N3@192.168.1.2:1099
/JADE,S3@192.168.1.2:1099/JADE
S16: Received file s4sFile3. Hooray! Path:
S16@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,S4@192.168.1.2:1099/JADE
S17: Received file s5sFile3. Hooray! Path:
S17@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE,S5@192.168.1.2:1099
/JADE
S18: Received file n5sFile3. Hooray! Path:
S18@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,N5@192.168.1.2:1099/JADE
S19: Received file n2sFile3. Hooray! Path:
S19@192.168.1.2:1099/JADE,N3@192.168.1.2:1099/JADE,N2@192.168.1.2:1099/JADE
S20: Received file n1sFile3. Hooray! Path:
S20@192.168.1.2:1099/JADE,N6@192.168.1.2:1099/JADE,N1@192.168.1.2:1099/JADE
N2 Statistics :
       11 pings received,
       10 pings forwarded,
       5 piongs forwarded,
       4 file requests initiated,
       32 file requests forwarded,
       100 file hits forwarded,
       22 file hits received.
       Total messages received: 371
N3 Statistics :
       9 pings received,
       9 pings forwarded,
       4 piongs forwarded,
       4 file requests initiated,
       16 file requests forwarded,
       118 file hits forwarded,
       16 file hits received.
       Total messages received: 382
N4 Statistics :
       7 pings received,
       7 pings forwarded,
       4 piongs forwarded,
       4 file requests initiated,
       22 file requests forwarded,
       111 file hits forwarded,
       22 file hits received.
       Total messages received: 370
N5 Statistics :
       5 pings received,
       4 pings forwarded,
       20 piongs forwarded,
       4 file requests initiated,
       30 file requests forwarded,
```

```
       110 file hits forwarded,
       20 file hits received.
       Total messages received: 376
N6 Statistics :
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       4 file requests initiated,
       0 file requests forwarded,
       27 file hits forwarded,
       13 file hits received.
       Total messages received: 76
S1 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
       Total messages received: 7
S2 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       4 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
       Total messages received: 11
S3 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       4 file requests forwarded,
       0 file hits forwarded,
       0 file hits received.
       Total messages received: 7
S4 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       1 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 5
S5 Statistics (my supernode is N2@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       4 file requests forwarded,
       0 file hits forwarded,
       0 file hits received.
       Total messages received: 7
S6 Statistics (my supernode is N5@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
```

```
      Total messages received: 7
S7 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      10 file requests forwarded,
      0 file hits forwarded,
      10 file hits received.
      Total messages received: 23
S8 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      10 file requests forwarded,
      0 file hits forwarded,
      10 file hits received.
      Total messages received: 23
S9 Statistics (my supernode is N1@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      10 file requests forwarded,
      0 file hits forwarded,
      10 file hits received.
      Total messages received: 23
S10 Statistics (my supernode is N2@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      10 file requests forwarded,
      0 file hits forwarded,
      10 file hits received.
      Total messages received: 23
S11 Statistics (my supernode is N2@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      10 file requests forwarded,
      0 file hits forwarded,
      10 file hits received.
      Total messages received: 23
S12 Statistics (my supernode is N4@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      13 file requests forwarded,
      0 file hits forwarded,
      0 file hits received.
      Total messages received: 16
S14 Statistics (my supernode is N5@192.168.1.2:1099/JADE):
      0 pings received,
      0 pings forwarded,
      0 piongs forwarded,
      1 file requests initiated,
      4 file requests forwarded,
      0 file hits forwarded,
      4 file hits received.
      Total messages received: 11
S15 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
```

```
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
       Total messages received: 7
S16 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       1 file requests forwarded,
       0 file hits forwarded,
       1 file hits received.
       Total messages received: 5
S17 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       4 file hits received.
       Total messages received: 7
S18 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       4 file requests forwarded,
       0 file hits forwarded,
       10 file hits received.
       Total messages received: 17
S19 Statistics (my supernode is N3@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       10 file hits received.
       Total messages received: 13
S20 Statistics (my supernode is N6@192.168.1.2:1099/JADE):
       0 pings received,
       0 pings forwarded,
       0 piongs forwarded,
       1 file requests initiated,
       0 file requests forwarded,
       0 file hits forwarded,
       13 file hits received.
       Total messages received: 16
```

References:

JADE - http://jade.tilab.com/

JGraphX - https://github.com/jgraph/jgraphx