

## Assignment 3 - Documentation

Hamidreza Tavakoli

February 25, 2012

# 1 Summary

In this project the P2P Distributed Hash Table application is implemented in Java. This program is created using Object oriented design. Documentations are also provided in the code for a better understanding of how the application works. During development of this application, GIT source version controlling system was used and the code is hosted on Github: [git://github.com/hmrtk/p2p-dht-simple-protocol-implementation.git](https://github.com/hmrtk/p2p-dht-simple-protocol-implementation.git) .

# 2 Design

This program consists of 6 separate class objects as follow:

## **peer**

Contains the main method, creates an instance of PeerNode, processes the arguments that are passed to it from the command line through a method in the PeerNode instance, and passes the PeerNode object to Serverthread class.

## **ServerThread**

Implements the Server part of the application. This server side of the application is implemented using multi threading design. It further uses methods inside the PeerNode object to processes the messages received from client and reply to them.

## **PeerNode**

Implements a peer node in the network, it has all the required properties that defined in the assignment. It uses two objects Request and Response and Settings. The client part of the application that interacts with the server is implemented int this class inside a method. This class also takes care of all the conditions in the protocol when a message is being sent and received.

## **Request**

Simulates a message that is sent to a server. When server gets the message the message is translated into this object, that makes it easier to further process the message through the application.

## **Response**

Response class is similar to Request class with the only difference that it simulates messages that are sent from server to the client and is processed through client section of application.

## **Settings**

This class has properties that are used as setting in the program. As an example the protocol version is defined here and can be changed if necessary.

## **Installation**

This program was developed using Eclipse IDE. The compiled files are located in the /bin folder and source files are located in the /src folder.

The peer.java file is the starting point of the application that creates a peer node in the network. The switches that can be passed to this file are:

- i n — the ID for this peer
- h x — host name of the computer that this program is running on
- p n — port on which this peer will listen for other peers
- m n — the maximum ID value
- r x — host name of another peer in the system
- s n — port number the other peer in the system

-f — present if this is the first host in the system

Here is how this application can run from a unix operating system:

For the first peer the -f switch is passed to indicate that this peer is the only peer in the network.  
java peer -i 15 -h ubuntu -p 2112 -m 32 -f

Examples for other peers:

```
java peer -i 10 -h ubuntu -p 2113 -m 32 -r ubuntu -s 2112
java peer -i 20 -h ubuntu -p 2114 -m 32 -r ubuntu -s 2112
```

The communications between each peer is logged in server.p2plog under the /bin folder.

To communicate with each node, the following command can be used:  
telnet hostname port

i.e telnet ubuntu 2112

These are the messages that can be sent to any servers using telnet:

ID — Determine the peer in the system that is responsible for a specific ID. i.e ID 3171<sub>a</sub>3/1.0 0 13CRLF  
ADD — Add a string to be stored at the peer. i.e ADD 3171<sub>a</sub>3/1.0 1CRLFwhat time is it?CRLF  
QUERY — Determine whether or not this peer is storing a specific string. i.e QUERY 3171<sub>a</sub>3/1.0 1CRLFkumquatCRLF  
PULL — Retrieve any strings at this host that lie at or above the ID value of the requesting peer. That peer is getting ready to take over those strings. i.e PULL 3171<sub>a</sub>3/1.0 1 27CRLFhector.cs.dal.ca 3000CRLF  
NEXT — Return the information for the next peer in the system. i.e NEXT 3171<sub>a</sub>3/1.0 0CRLF  
DONE — Indicate to a peer that the calling peer will now take responsibility for answering all queries about strings at or beyond the sending peer's ID value. i.e DONE 3171<sub>a</sub>3/1.0 0 27CRLF

All messages have the following basic format where < sp > represents a space:

operation < sp > version < sp > number of following lines < sp > optional peer ID CRLF  
0 or more following lines, each ending with CRLF

All responses have the following message format:

version < sp > operation < sp > number of lines < sp > response code < sp > response string CRLF  
0 or more following lines, each ending with CRLF

## Testing

The server.p2plog contains the last Test cases that has been done on this program. In this section the test case for 3 peers in the network is included:

```
java peer -i 15 -h ubuntu -p 2112 -m 32 -f
Server 15 Started!
```

```
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
```

```

Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
quitHostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 i
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin
Hostname: ubuntu Port: 2112 ID: 15 NextHostname: ubuntu NextPort: 2114 NextID: 20 MaxID: 32 isFin

java peer -i 10 -h ubuntu -p 2113 -m 32 -r ubuntu -s 2112
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Server 10 Started!
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin
Hostname: ubuntu Port: 2113 ID: 10 NextHostname: ubuntu NextPort: 2112 NextID: 15 MaxID: 32 isFin

java peer -i 20 -h ubuntu -p 2114 -m 32 -r ubuntu -s 2112
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Server 20 Started!
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin
Hostname: ubuntu Port: 2114 ID: 20 NextHostname: ubuntu NextPort: 2113 NextID: 10 MaxID: 32 isFin

-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] ID 3171_a3/1.0 0 10CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 ID 0 200 okCRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] NEXT 3171_a3/1.0 0 0CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 NEXT 1 200 okCRLFubuntu 2112 15CRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] PULL 3171_a3/1.0 1 10CRLFubuntu 2113CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 PULL 1 200 okCRLFblahCRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] DONE 3171_a3/1.0 0CRLF
[Sent - PeerID: 15 Portnum: 2112]
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] ID 3171_a3/1.0 0 20CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 ID 0 200 okCRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] NEXT 3171_a3/1.0 0 0CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 NEXT 1 200 okCRLFubuntu 2113 10CRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] PULL 3171_a3/1.0 1 20CRLFubuntu 2114CRLF
[Sent - PeerID: 15 Portnum: 2112] 3171_a3/1.0 PULL 1 200 okCRLFblahCRLF
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] DONE 3171_a3/1.0 0CRLF
[Sent - PeerID: 15 Portnum: 2112]
-----> peer 15 server logging <-----
[Received - PeerID: 15 Portnum: 2112] ADD 3171_a3/1.0 1CRLFblah blah blahCRLF

```

```

-----> peer 20 server logging <-----
[Received - PeerID: 20 Portnum: 2114] ADD 3171_a3/1.0 0 15CRLFblah blah blahCRLF
[Sent - PeerID: 20 Portnum: 2114]      3171_a3/1.0 ADD 0 200 okCRLF
[Sent - PeerID: 15 Portnum: 2112]      3171_a3/1.0 ADD 0 400 NotResponsibleCRLFblah blah blahCRLF
[Received - PeerID: 15 Portnum: 2112] ADD 3171_a3/1.0 1CRLFthis is assignmentCRLF
-----> peer 20 server logging <-----
[Received - PeerID: 20 Portnum: 2114] ADD 3171_a3/1.0 0 15CRLFthis is assignmentCRLF
-----> peer 10 server logging <-----
[Received - PeerID: 10 Portnum: 2113] ADD 3171_a3/1.0 0 20CRLFthis is assignmentCRLF
[Sent - PeerID: 10 Portnum: 2113]      3171_a3/1.0 ADD 0 200 okCRLF
[Sent - PeerID: 20 Portnum: 2114]      3171_a3/1.0 ADD 0 400 NotResponsibleCRLFthis is assignmentCRLF
[Sent - PeerID: 15 Portnum: 2112]      3171_a3/1.0 ADD 0 400 NotResponsibleCRLFthis is assignmentCRLF
[Received - PeerID: 15 Portnum: 2112] QUERY 3171_a3/1.0 1CRLFthis is assignmentCRLF
[Sent - PeerID: 15 Portnum: 2112]      3171_a3/1.0 ADD 0 400 NotResponsibleCRLFthis is assignmentCRLF
-----> peer 10 server logging <-----
[Received - PeerID: 10 Portnum: 2113] QUERY 3171_a3/1.0 1CRLFthis is assignmentCRLF
[Sent - PeerID: 10 Portnum: 2113]      3171_a3/1.0 QUERY 0 200 OKCRLF

```

```

telnet ubuntu 2112
Trying 127.0.1.1...
Connected to ubuntu.
Escape character is '^'.
ADD 3171_a3/1.0 1CRLFblah blah blahCRLF
3171_a3/1.0 ADD 0 400 NotResponsibleCRLFblah blah blahCRLF
ADD 3171_a3/1.0 1CRLFthis is assignmentCRLF
3171_a3/1.0 ADD 0 400 NotResponsibleCRLFthis is assignmentCRLF
QUERY 3171_a3/1.0 1CRLFthis is assignmentCRLF
3171_a3/1.0 ADD 0 400 NotResponsibleCRLFthis is assignmentCRLF

```

```

telnet ubuntu 2113
Trying 127.0.1.1...
Connected to ubuntu.
Escape character is '^'.
QUERY 3171_a3/1.0 1CRLFthis is assignmentCRLF
3171_a3/1.0 QUERY 0 200 OKCRLF

```