T-110.5150 – Application and Services in Internet

Assignment 1 Report: Peer-to-Peer Network

Group 20
Bhorkar Gaurav (545691)
Thapa Manish (534398)

1. INTRODUCTION

This report describes the development assignment of a peer-to-peer application. The report consists of description about list of features implemented along with its development and usage procedure. The source code along with readme file and assignment report is available in latest commit in the GIT repository provided to us.

The application is developed in C. It is based on given protocol specification which originated from Gnutella 0.6.

We started the project by getting insights of socket programming in Linux using C. Initially getting clear understanding about working of sockets was bit complex. Once we got full view of the application and its features we started development and conquered features one by one by keeping a checklist. Final status of checklist is in section 2.

2. CHECKLIST OF FEATURES IMPLEMENTED

Following functions are implemented and tested properly. Functioning of program and execution of features are logged into file ***log.txt*** created dynamically in the same source folder of the application. The implementation details in brief and usage of each function are explained in detail in section 3.

2.1 Basic Features

No.	Features	Checklist
1	Send Join Request message in order to join the network	Implemented
2	Handle Join Request message	Implemented
3	Send Query message with a search key and handle Query Hit messages	Implemented
4	Publish a key/value pair in the network and respond to a Query message when its search key equals your published key	Implemented

Table 1: Basic Features Checklist Table

2.2 Advanced Features

No	Features	Checklist
5	Join multiple peers to expand your network	Implemented
6	Send Type A Ping message and handle Type A Pong message to check if your neighbors are alive.	Implemented
7	Respond to Type A Ping message with a Type A Pong message. Tell your neighbor that you are still alive.	Implemented
8	Send Type B Ping message and handle Type B Pong message to find more node information in the network	Implemented
9	Respond to Type B Ping message with a Type B Pong message. Tell your neighbor who else you know.	Implemented
10	Forward Query messages with loop avoidance to help other node to find a resource	Not Implemented
11	Forward Query Hit message by following the reverse path of its corresponding Query message	Not Implemented

Table 2: Advanced Features Checklist Table

3. IMPLEMENTATION

We designed the application layout in such a way that it is interactive and supports command line UI. Usage of application with description of menu is available in README file in the same submission package. We modeled the control flow in a way that after opening a listener socket on the user specified port, the main function waits infinitely until a command is given to the application by the user or some activity is done on listener socket or connected peer sockets. To implement this flow we used select () function which waits for any activity on file descriptor of a socket. This way we were able to have a single process and single threaded application. We created a peer handler which handles message received on a peer socket. Based on message received corresponding message handlers are executed which are created as separate functions that implement the required feature in the list.

Once we had control flow logic ready we started coding in pair. Before implementing any feature we first discussed the protocol specification for that feature and constructed the message handler depending on type of message. Debugging was done together and was easier when working together. Figuring some weird errors in socket programming was sometimes so time consuming that our development process was paused. We spent approximately following times each.

Background study and understanding the given protocol	7 hrs
Discussing and deciding the control flow for application	5 hrs
Coding (approx) including discussion and implementation of each feature	20 hrs
Report writing and final touch up (miscellaneous)	3 hrs

Table 3: Time spent during the project

In total we spent around 35 hours for each member to develop the project.

4. COMMENTS ON ASSIGNMENT

Overall we felt that the assignment is bit complex, especially for the students who do not have prior experience in socket programming. We think that few exercise sessions regarding socket programming might be very helpful for students.

5. CONCLUSION

This assignment helped us learn socket programming and understand how peer-to-peer networks actually work. It also helped us to know the various implementation approaches for peer-to-peer network application development.