Optimized File Sharing Application within Local Network

By

A.K.M. Nazmul Islam – 09301011

Md. Nigran Islam Mizan – 11201001

Under the guidance of

Ms. Sadia Hamid Kazi

Assistant Professor

BRAC University

Statement:

We hereby proclaim that this thesis is based on the results we found by our hard work. Contents of the work found by other researcher(s) are motioned by references. This thesis has never been previously submitted for any degree neither in whole nor in part.

Signature of Supervisor:

Ms. Sadia Hamid Kazi

Signature of Authors:

A.K.M. Nazmul Islam

[Student ID: 09301011]

Md. Nigran Islam Mizan

[Student ID: 11201001]

Acknowledgment:

We would like to thank almighty ALLAH for giving us the opportunity and aptitude to complete a valuable amount of knowledge in this life span and with His blessings; we shall continue to pursue all the good things in our life ahead.

We also thank our supervisor, Ms. Sadia Hamid Kazi heartily for giving us the opportunity to work under her and also for her kind support in making this research successful.

We thank BRAC University for giving us the opportunity to be a part of it and all the faculty members who have taught and motivated us to think more deeply.

Finally, we would like to thank all the seniors and juniors who have helped in minute tasks. We may miss names so we thank you all heartily.

Table of Contents page

List of figures

Abstract

File sharing is the practice of distributing or providing access to digitally stored information, such as computer programs, multimedia (audio, images and video), documents or electronic books. It may be implemented through a variety of ways. Either by using client server or by peer to peer methodology file sharing is conducted. Nowadays sharing files is a major part of the network usage. Files are being shared between multiple of users and thus creating a load on the network. Thus optimizing file sharing is important for the network to be efficient. The objective of this thesis was to study different file sharing algorithms and chose one that would create an optimized file sharing solution for a local area network using peer to peer method. Among different optimizing algorithms Ant Colony Optimization (ACO) was chosen and compared other available algorithms. Based on that, we have tried to develop a file sharing application within a local area network using the concept of peer to peer file sharing methodology named FAaNS. Thus this research gives an overview of the ACO algorithm, reasons behind choosing ACO and then finally the development of the file sharing application which will detect the nodes that have the files for sharing and will share the files optimally between nodes.

Part 1: Introduction

1.1: Objective:

Nowadays sharing files is a major part of the network usage. Files are being shared between multiple of users and thus creating a load on the network. Thus optimizing file sharing is important for the network to be efficient.

The main purpose of this project is to study file sharing algorithms and its optimization. Then after review several we have chosen Ant Colony Optimization (ACO), and then we did a comparison with other available algorithms. Then a file sharing application named FAaNS was developed based on that. Through this project, we can get an overview of ACO algorithm, reasons behind chosen ACO, literature review, methodology and our application FAaNS with interface and simulation.

We have developed a file sharing application with user friendly interface named File Allocation Amongst Native Systems (FAaNS). This application has connection types (LAN or WLAN) to share the files. It has also available option to detect the devices of a certain network. We use JAVA language to build up the interface of application. This application gives few statements and warning messages into “Notes” option depending on exception. There is also a reset button to reset the application. We had tried to use the concept of BitTorrent to send the file from source to destination.

1.2: Outline of the Report:

This report is divided into five parts. Part 1 discusses brief description, goal of the project and outline of the project. Part 2 consists of the descriptions of file sharing and its classification, optimization, descriptive overview of ACO algorithm and reasons behind chosen ACO. This part also contains literature review. Part 3 emphasizes methodology of the project. Part 4 clarifies the application named FAaNS with screenshot and its simulation. And part 5 concludes this project along with the future prospect of this project in indicated at the end.

Part 2: Theory

2.1: File sharing:

The perform of sharing or offering entrance to digital data, information or resources including documents, multimedia (audio/video), graphics, computer programs, images and e-books is called file sharing. There is either the private or the public sharing of information or resources within a network with different levels of sharing privileges. To transmit data so easily and quickly, file sharing is executed on computer networks. As an example, a user may share a video from his computer that is connected to a corporate network permitting all other employees to access the video. [1][2]

Figure 1: File sharing

2.2: Classification of file sharing:

File sharing has three major types. In a network, one file-sharing method or any combination of these three can be used. They are as below:

i. System-native

Our operating system is constructed by this type of file sharing. It is usually intended for sharing files in our local network with other users. Sometimes it allows printers also. But it is not intended for sharing files with others across the internet. The most commonly used examples of this model are My Network Places in Microsoft Windows and Bonjour in the past called AppleTalk in OS X.

ii. Client- server

Client-server file sharing model covers different types of fields such as web pages, email, FTP, cloud storage services and so on. Sometimes the data is stored in a central location. It is called the server. Often the data is sent separately to each user who requests it. They are called the clients. This model is mainly used for file sharing through internet.

iii. **Peer-to-peer**

This model of file sharing doesn't rely on a single central server. Any two users can exchange data, information or resources with each other directly where one user can upload files to the server and another user can download files from the server. [3]

We are using client-server file sharing type and the concept of peer-to-peer within local network.

2.3: Optimization:

Optimization means to find an alternative option with the most cost effective or maximum feasible performance under the specified limitations by taking advantage of desired features and reducing undesired ones. The lack of full information and the lack of time to estimate what information being available are constrained by practice of optimization. Optimization is usually achieved by using linear programming techniques of operations research in computer simulation or modeling of business problems. [4]

2.4: Ant Colony Optimization (ACO):

Ant Colony Optimization (ACO) studies artificial systems to take idea from the actions of real ant colonies which are used to solve discrete optimization problems. ACO is a population-based algorithm where several artificial ants find out good solutions. Every ant puts up a solution phase by phase in that way going through several decisions until a solution is found. Ants that found a good solution mark their paths through the decision space by

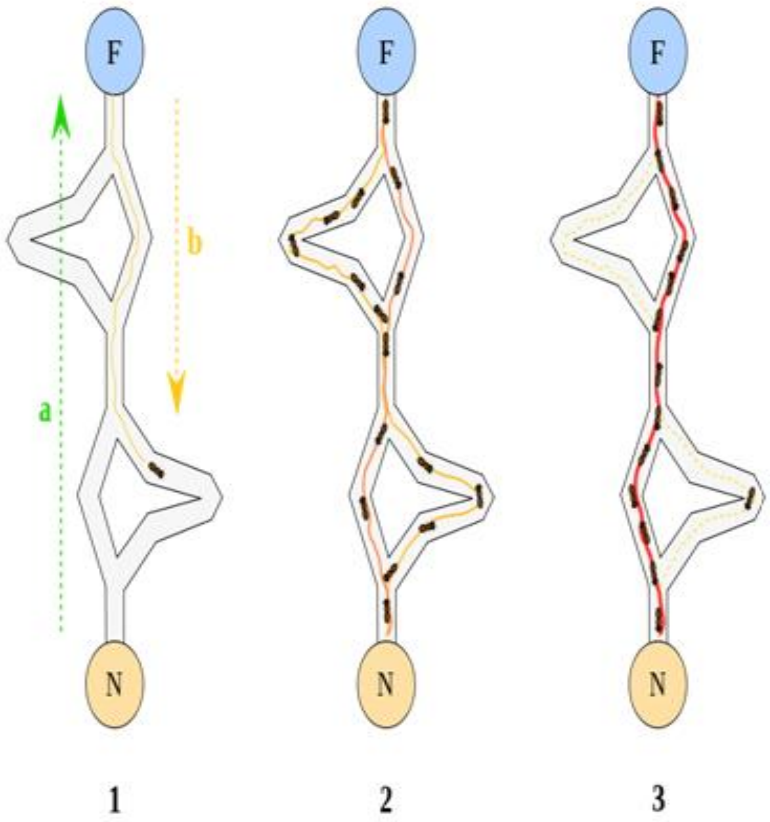

Figure 2: Ant Colony Optimization

putting some amount of pheromone on the edges of the path. The following ants of the next generation are attracted by the pheromone so that they will search in the solution space near good solutions. [5]

2.5: Ant Colony Algorithm:

According to Ant-based load balancing in telecommunication networks (Schoonderwoerd, R. -1996):

- Network has n nodes.
- Each node has its Routing Table (pheromone table)
- Initialize: equilibrium Routing table (all nodes have the same value or normalized random values)
- Each node lunches {n-1} ants (agents) each to different destination.
- Each ant select its next hop node proportionally to goodness of each neighbor or node
- Routing table of the node that just the ant arrived to is updated. [6]

2.6: Why ACO chosen:

Ant colony algorithm has a number of qualitative and quantitative advantages over Dijkstra's algorithm as well:

i. Consuming network resources

An ant hardly requires any bandwidth on the network: It only holds its age, and its source and destination identifiers. In Dijkstra, there are relatively large tables and therefore require much more bandwidth than ant colony algorithm. Congestion dependent ants temporarily reduce traffic at the congested nodes.

ii. Robustness

Malfunctioning in the system might cause a process such as an ant or to crash. If an ant crashes, this will not have a significant effect on the performance of the algorithm at all but if a node fails in Djikstra algorithm all the tables have to be modified.

iii. Computational issues

Ants are likely to require more computation on the nodes of the network, due to the extensive use of random generators. Further, with Ant based control, nodes need to allocate more space for their pheromone tables than is needed when normal routing tables are used. However, these issues do not affect and width or switching capacity, which is our main concern.

iv. Bidirectional routes

During the simulations of the ant controlled system, the route from one node to another tends most of the time to be the same that in the opposite direction. This is probably due to our mechanism of trail lying, where ants from complementary source and destination nodes mutually reinforce one anther's trails. At first sight, this property might seem to be disadvantageous for good load balancing, but we believe that this will only make a significant difference in small networks. [7]

2.7: Literature Review:

We are motivated to do our project from few projects and researches which are related to our project. We spent times to improve our project based on those resources. Here is given a short description few of them:

i. Ant based load balancing telecommunications networks

This paper is based on a new method of achieving load balancing in telecommunications networks. A simulated network models a typical distribution of calls between nodes which carrying an excess of traffic and can become congested. As a result, causing calls to be lost. So, the network supports a population of simple mobile agents which following actions of ants. Calls between nodes are routed as a function of the pheromone distributions at each intermediate node. The performance of the network is calculated by the proportion of calls which are lost. The results compared with those achieved by using fixed shortest-path routes. [8]

ii. AntNet: A Mobile Agents Approach to Adaptive Routing

This paper describes a new routing algorithm for communications networks named AntNet. This is an adaptive, distributed, mobile-agents-based algorithm on the ant colony metaphor. AntNet to a datagram network and compare it with both static and adaptive state-of-the-art routing algorithms. It has good performance and robustness under all the experimental conditions with respect to its competitors. [9]

iii. Simulation of Route Optimization with load balancing Using AntNet System

This paper explains analysis of the performance of load balancing and route optimization in computerized networks. This model shows packet distribution between nodes and due to traffic packet to be failed if congestion occurs. To simulate the network, this model use AntNet system. For the load balancing of the network, the simulation runs on the ant's behavior. This is also an adaptive system, the nodes are removed from the network and system finds the alternative chosen paths without system down and controls the performance of routing. [10]

iv. Load Balancing in P2P networks using DHT based systems and Ant based systems: A Comparison

The paper based on peer to peer (P2P) networks. It has its advantage as well as disadvantage. This topic describes about amount of storage, computing power, connectivity, bandwidth etc. in a cost effective behavior of networked computers. To balance these systems in a cost-effective way so many approaches are available. Here focus on two main approaches: the Distributed Hash Table (DHT) based systems and Ant based systems. Both approaches are having own advantages and limitations. [11]

v. Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm

This paper is based on ACO which has been established a successful method and applied to a number of combinatorial optimization problems and is taken as one of the high performance computing methods for Traveling salesman problem (TSP). ACO has very good search capability for optimization problems, but it still ACO algorithm costs too much time to

convergence and traps in local optima in order to find an optimal solution for TSP problems. This paper highlights firstly to adapt rapid convergence speed and secondly a dynamic updating rule to improve the performance in solving TSP. [12][13]

Part 3: Methodology

We emphasize methodology of our project within three sections. In project design section, we will describe how we designed our project. In project tools section, we will explain how we utilized tools related our project. In data analysis section, we will clarify how we assembled data from resources and how we analyzed them.

3.1: Project design:

We design our project into two ways. For theory part, we collect our data, information and resources from IEEE materials and BRAC library journals. We gather articles, journals from those fields which are related to our project. We also collect data and information from trusted website written by certified authors. To design our application, we use java language and its library. Also, we were always active along with our supervisor to improve our project day by day.

3.2: Project tools:

In order to collect the most accurate and relevant data, this project relates different project and research tools. We use Visual Studio 2010 to simulate the project code of "Artificial intelligence network load balancing using Ant Colony Optimization" By Lawrence Botly, 9 Apr 2007 and also use dotnetcharting to observe its output graphically to evaluate them. We use NetBeans IDE (version 7.4) to build up our application. We also use JDK (version 8) and JRE (version 8) as built-in tools for NetBeans IDE.

3.3: Data analysis:

We assemble articles, journals which are related to our project from IEEE materials and BRAC library journals and acquire the concept from them what we required. Literature review portion is an example of it. We also collect information from trusted website written by certified authors. We describe file sharing and its classification on the basis of it. We explain ACO algorithm and compare it with available another algorithm called Dijkstra as well. We take concept to sort out the process of methodology of our project from our university courses like software engineering, system analysis and design etc. Our supervisor also helps us a lot to decorate this portion as well. Thus, we assemble data from resources and analyze them.

Part 4: Application

For the development of the application we followed the whole structured procedure or the flow of the working process is as same as used to structure, plan, and control the process of developing a software system globally. The working process includes **Essential activities**, **Methodologies** and T**ools**.

4.1: Essential Activities:

i. Requirements

- Share big files along with directory reference through a local area network.
- A host can work both in sender or receiver mode.
- Work with any type of connection LAN or WLAN.
- Find active hosts within the network and list them.
- Form user friendly and easily understandable interface.
- Increases file transfer speed along with balancing load in the network and server host
- Work in cross platform.

ii. Design

The application works in two different modes as File Sender or as a File Receiver. While working in a mode the other one disabled for that moment on that host.

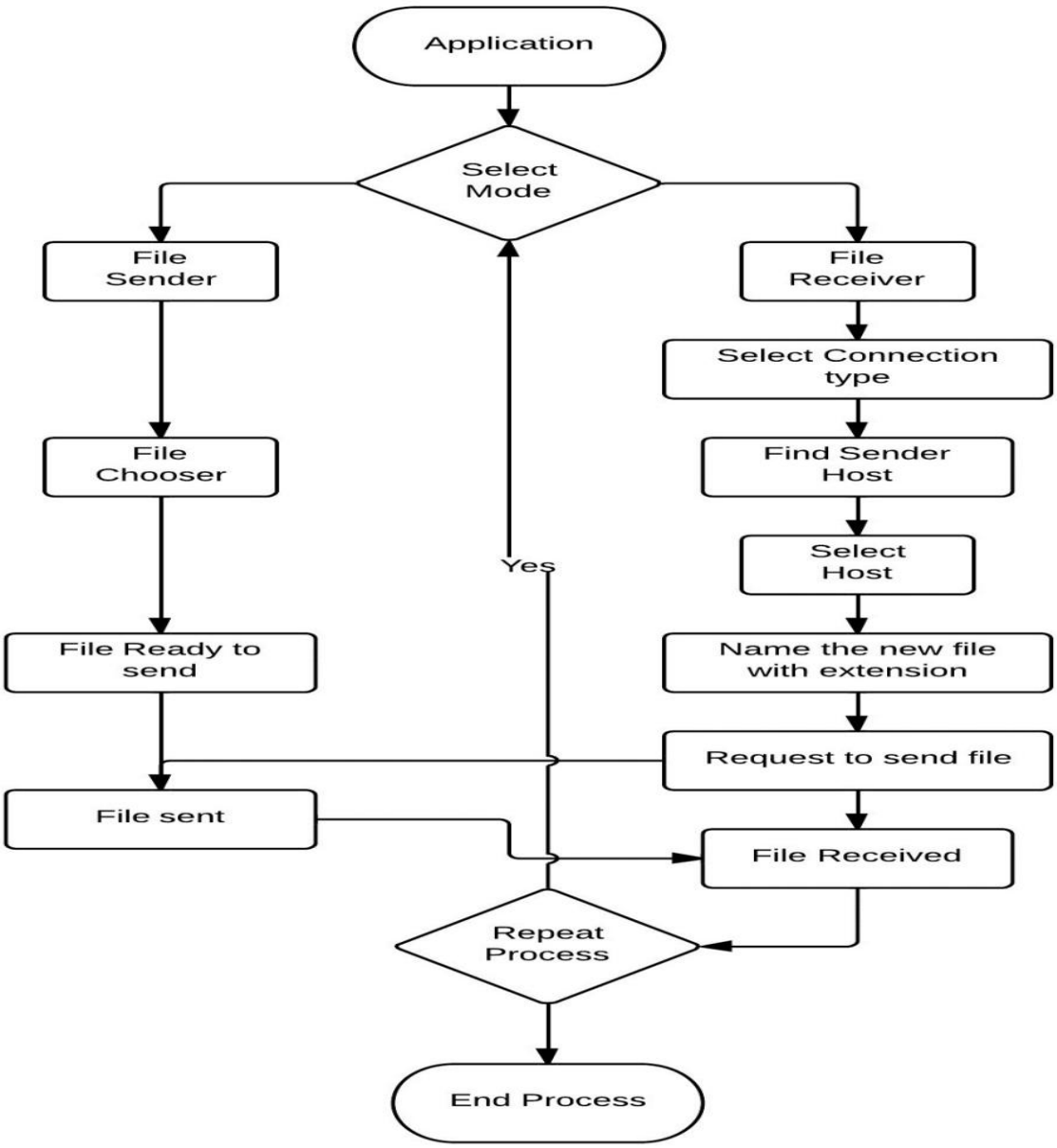

Figure 3: Application Flowchart

In File Sender mode by selecting file that needs to be sent, the sender becomes ready to send the file and waiting for request from receiver.

At **File Receiver** end after selecting the connection type application will search for the active host among the network and list them and show sender IP address as well. Name of the file with extension is given which is going to be received. Then from device list by choosing one

(The Sender host) the receiver request for the file and file sending process is initiated. Finally gives a successful message if the procedure worked perfectly and file transferred between the hosts.

iii. Testing

Because of this application is build up for project purpose we emphasize on its ability to work rather than its performance. We test the application using both black box and white box method.

Launch the application

The application does not require any kind of installation (except some pre installation such as updated version of JRE and JDK are mandatory to run). It has been built in such way that it can work like plug and play. After launching the application there is model selection, whether the host works as a sender or receiver. At the beginning phase application will create a folder named "**FAaNSfolder"** at C: drive in the host's disk drive.

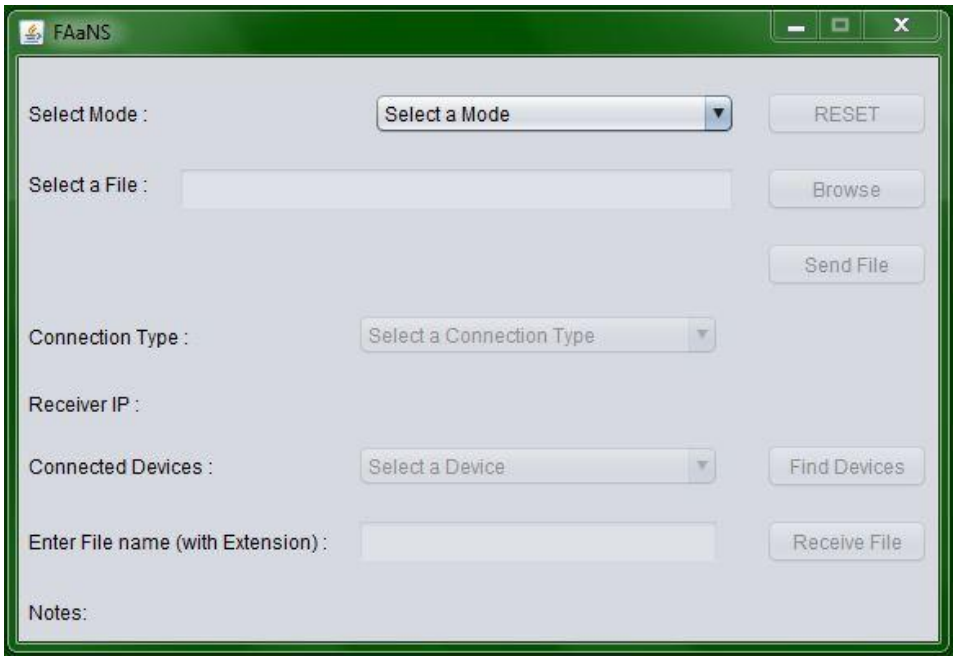

Figure 4: Startup Application interface

File Sender Mode

Selecting sender mode enables sender portion of the application. Then select the file need to send to the receiver. If no file is selected then a warning message will be given in the Notes field.

Figure 5: File Sender Mode

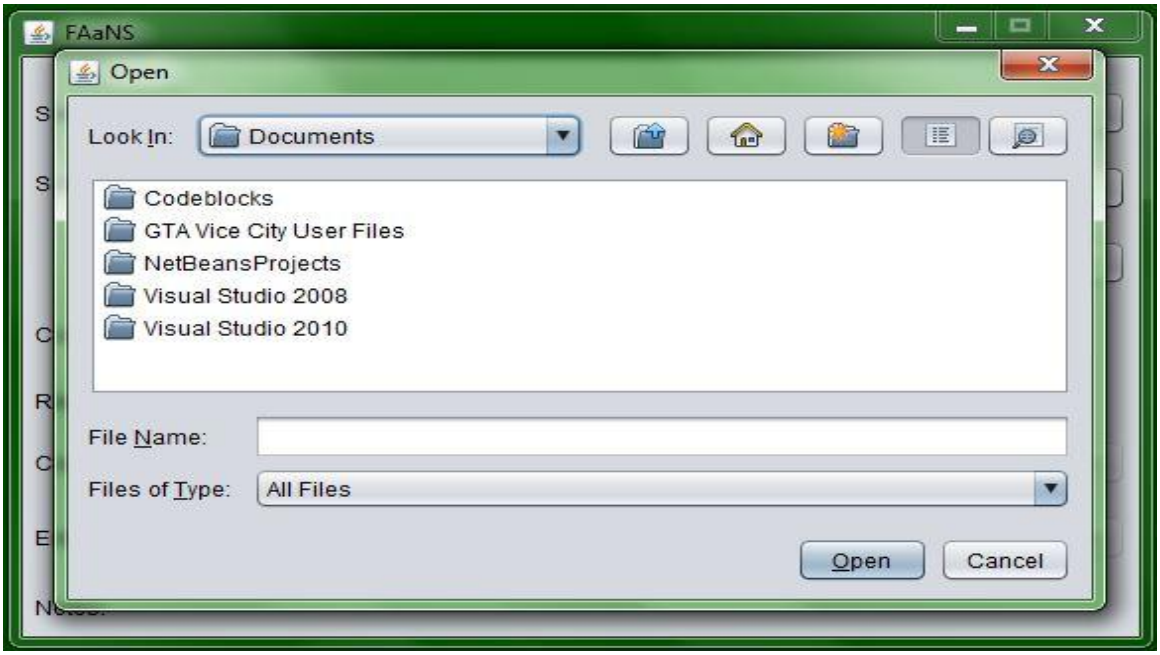

Figure 6: Selecting File

After selecting the file, sender is ready to send the file and wait for receiver's request to be made.

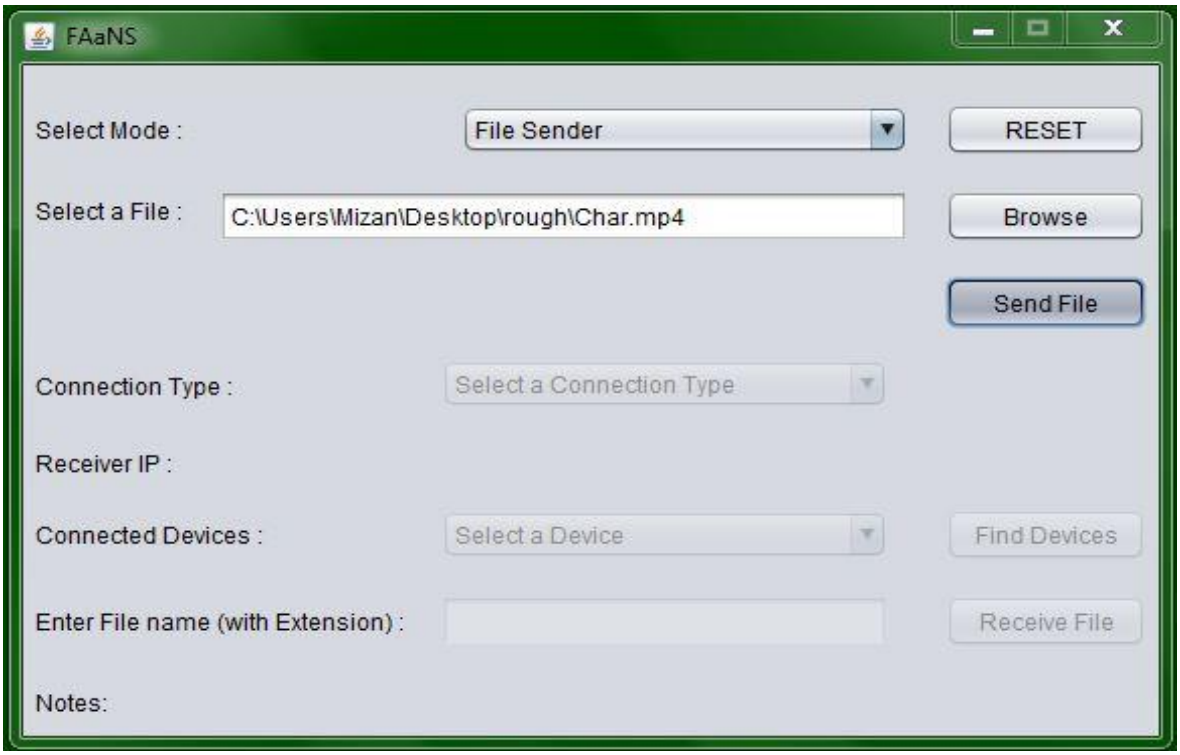

Figure 7: Waiting for Sending File

File Receiver Mode

In receiver mode after selecting connection type, searching for sender hosts within local area network is being initiated and lists the hosts as connected devices.

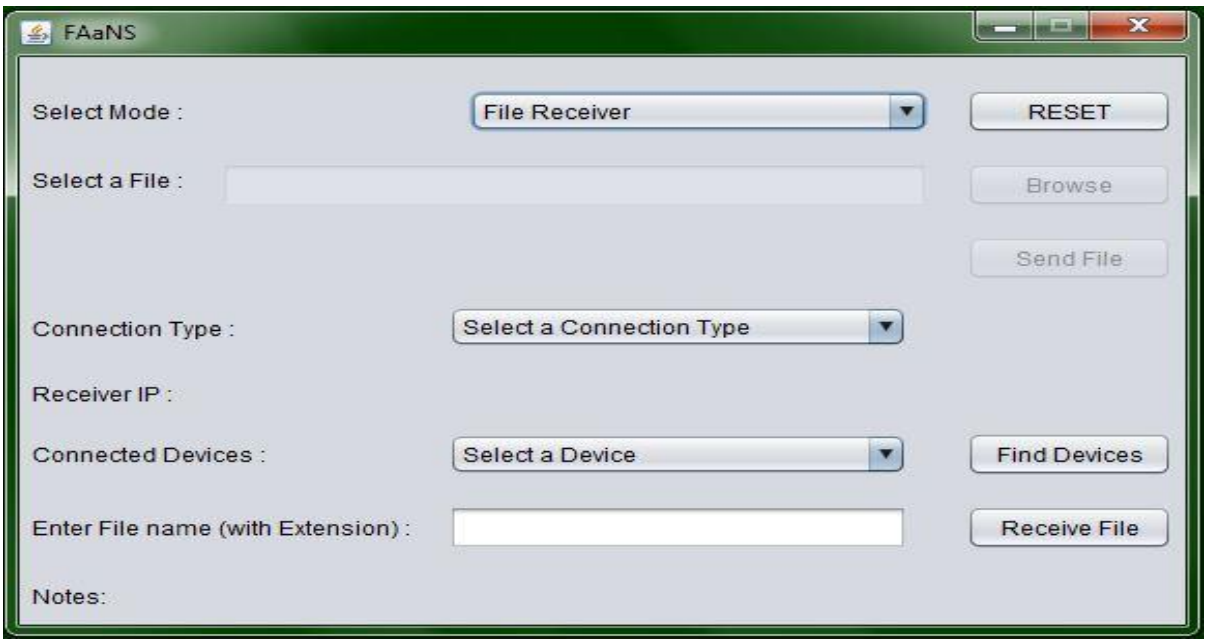

Figure 8: File Receiver Mode

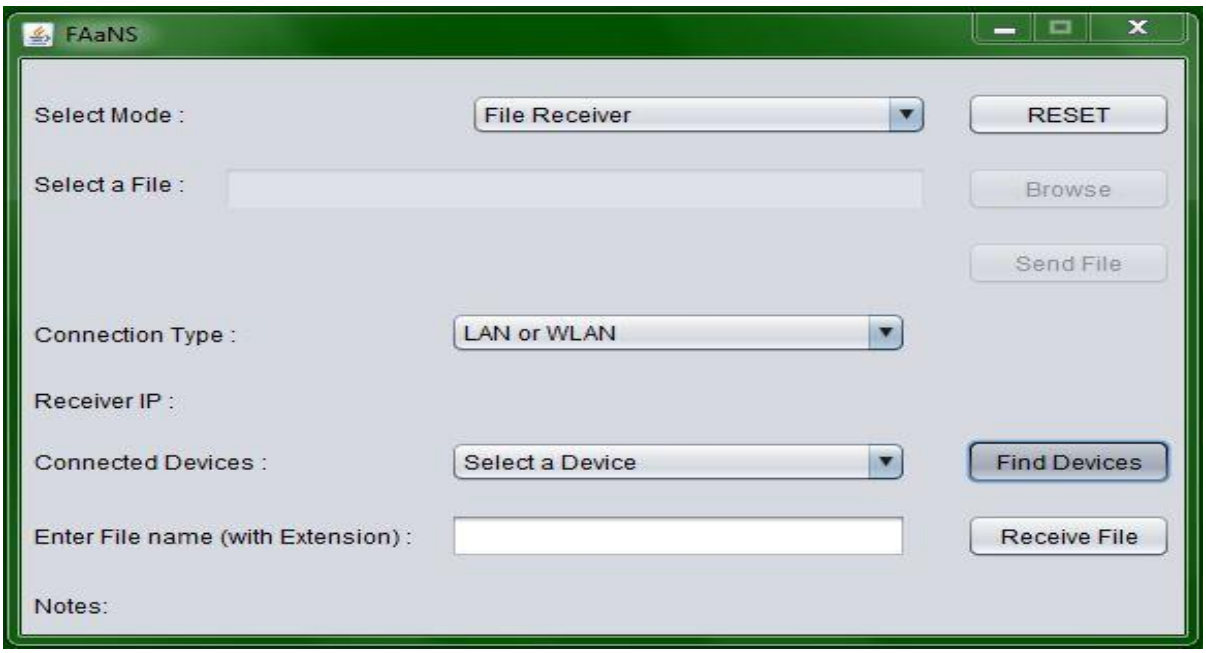

Figure 9: Searching for Devices

When the searching is finished the list of all active hosts will be in the connected device menu and the IP of the receiver host will be shown in the label.

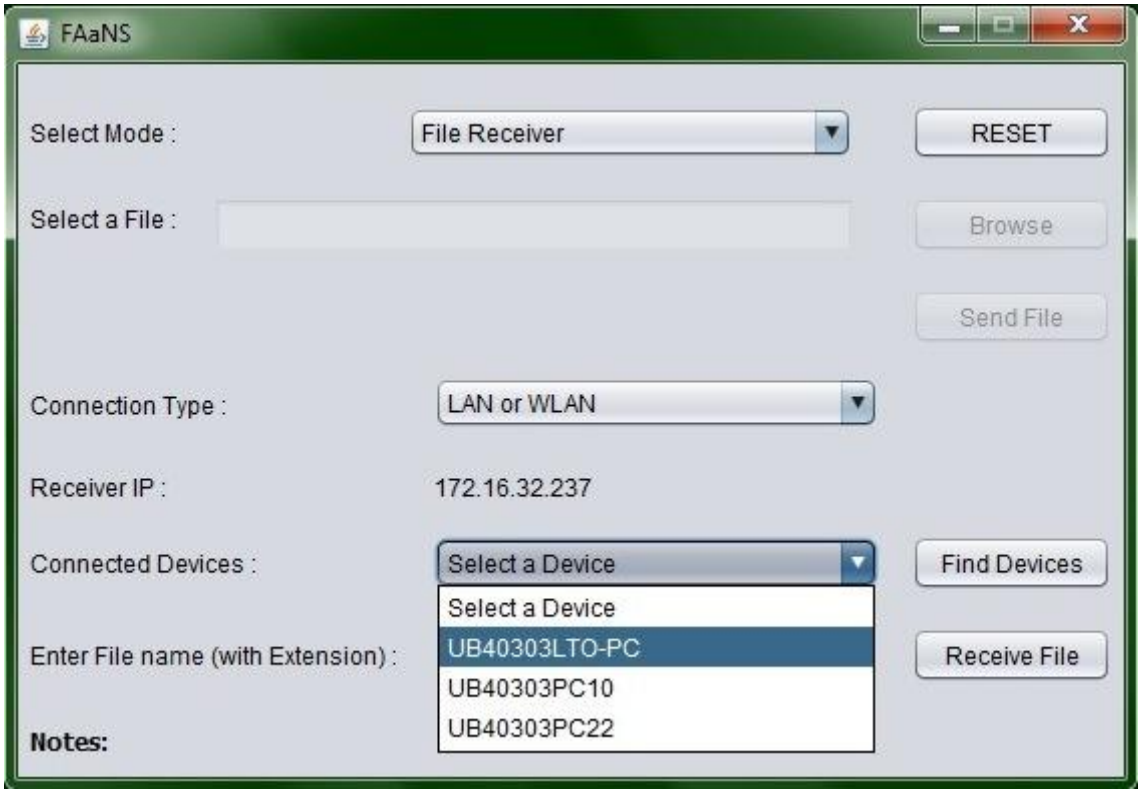

Figure 10: Showing Connected Device List

Consequently, the receiver host provides the name of the file with extension and selects the sender host. Request is been made for the file to the sender by pressing receive file button.

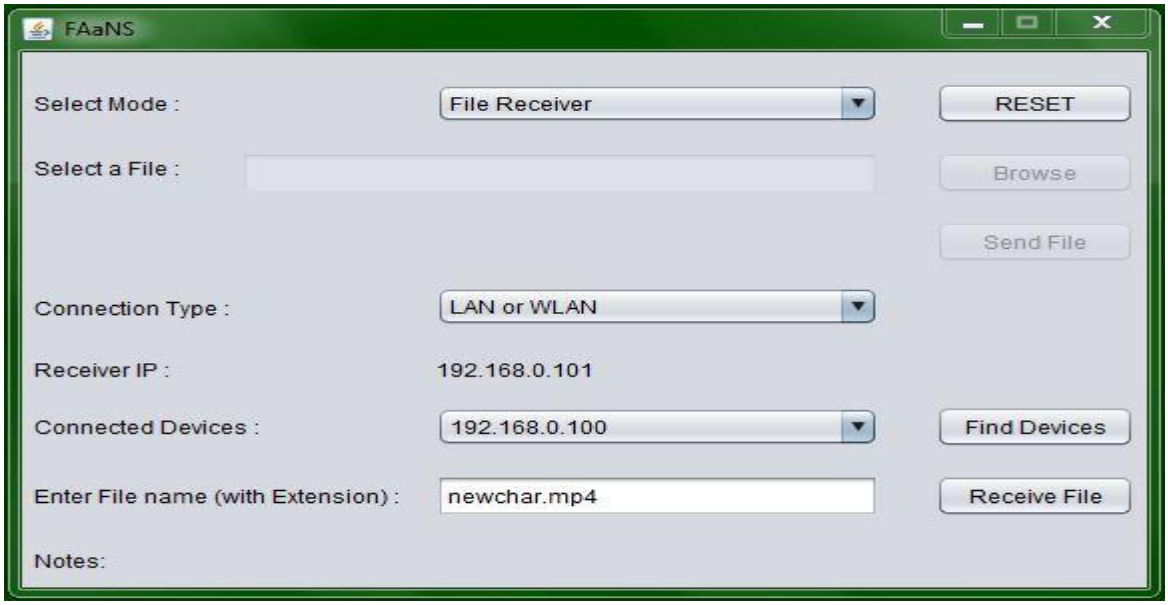

Figure 11: Request to Receive file

The file sending process started and if there is no exception occurred the sending process will be completed with confirmation message.

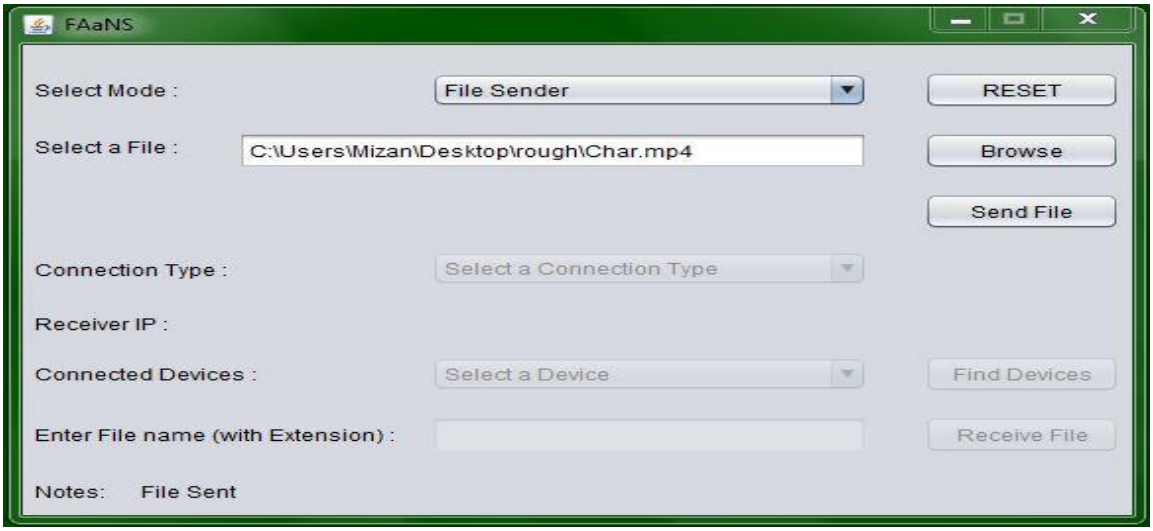

Figure 12: File Sent Notification

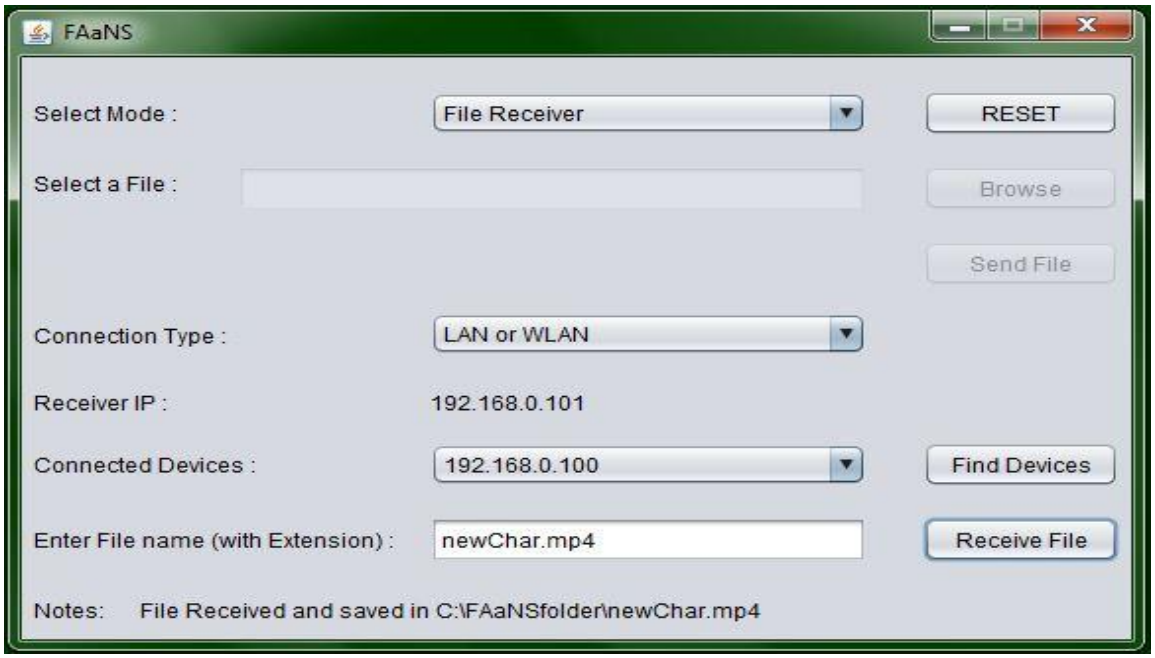

Figure 13: File Received Notification

Finally there is also a reset button with which we can reset the fields to work again or repeat the process.

4.2: Methodology:

To develop this application we followed agile development method more precisely XP (Extreme Programming) methodology.

As our requirement for the application changed while developing we change our application which is incremental planning of our project.

After configuring the software for few times we met with our advisor for feedback and then update, so we can consider them as different version with small releases.

Before including some new features in our project we test that before by implement that functionality differently and if that works perfectly we merge those within our project to fulfill the requirement of our project.

Here we also work in a very popular and effective way for software development which is pair programming. For more than one developer involving a project surely lacks the problems of a project and solves the problems more quickly.

In XP customer is a part of the team. We and our advisor play that role in this aspect thus it also fulfills a principle of agile development.

Hence, we can conclude with the word that our project building process fulfill the benchmarks of XP methodology.

4.3: Tools:

The whole program is designed and defined with Netbeans IDE. Java complier included in JDK version 8 is used to compile the program. As for Debugger a rich debugger bundled with Netbeans is utilized. Java swing package is used to design the graphical interface of the application. For network connection and features Net package and IO package is utilized for manipulating with file input output and further processing.

4.4: Limitations:

Though this app can work in cross platform but overall this application is slow to search devices. Besides, providing a specific extension name was problematic due to the file naming system of application (user have to know exact file extension name of the file due to run the file after transferring). The file size is also limited to 60MB. So, these all limitation rose while we started implementing the application.

Part 5: Conclusion

In conclusion, we may state that there is nothing called the "perfect" of file sharing application and its algorithm. Every application and algorithm has its advantages and disadvantages.

5.1: Concluding the Project:

The main focus of our project was to analyze file sharing, to study of ACO algorithm and to develop our file sharing application. We learned the different types of file sharing classes. We studied over ACO algorithm, reasons behind chosen this algorithm from others like Dijkstra etc. We also learned a lot from articles, journals and took the concept from them what we required. From methodology, we learned how to design a project, how to use project tools and also how to analyze data. It's a nice experience to learn so many things related with our project.

5.2: Future work:

We propose that this file sharing method can balance the load on a host for sending big files to more than one host at a time and consequently on the local area network as well. And it can also overcome the problems of FTP (Such as share the file along with directory) and surely becomes user friendly as we build so far. The limitation of application which was that the search was slow and file size had a limit which we hope to solve in the future. Later following this concept file transferring process can be implemented in multiple networks too. We also hope to increase its performance and establish it in such way thus it can be used globally.

References:

[1] Janssen C. (n.d.). File Sharing. Retrieved from

http://www.techopedia.com/definition/16256/file-sharing

[2] File sharing (n.d.). Retrieved from

http://www.computerhope.com/jargon/f/fileshar.htm

[3] Elements of Networking: Data and File Sharing (May 29, 2013). Retrieved from

http://www.cites.illinois.edu/network/concepts/elements-sharing.html

[4] Optimization (n.d.). Retrieved from

http://www.businessdictionary.com/definition/optimization.html

[5] Ant Colony Optimization (n.d.). Retrieved from

http://neo.lcc.uma.es/software/mallba/aco.php

[6] Schoonderwoerd, R. (1996) "Ant-based load balancing in telecommunication networks" Adapt. Behav. 5, 169-207

[7] C. Detrain, J.-L. Denebourg, and J. Pasteels. Decision-making in foraging by social insects. In C. Detrain, J.-L. Denebourg, and J. Pasteels, editors, *Information processing in social insects*, pages 331–354. Birkhauser, Basel, CH, 1999.

[8] Appleby, S., Steward, S. (1994). Mobile software agents for control in telecommunication networks. In *BT Technology Journal, Vol. 12, No.2*.

[9] ANTNet: A Mobile Agents Approach to Adaptive Routing. Gianni Di Caro and Marco Dorigo. "This work was supported by a Madame Curie Fellowship awarded to Gianni Di Caro
(CEC-TMR Contract N. ERBFMBICT 961153). Marco Dorigo is a Research Associate with the FNRS."

[10] IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 4 (May. - Jun. 2013), PP 01-07 doi: www.iosrjournals.org

[11] G. T. Chavan., M. A. Mahajan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 2, March -April 2013, pp.356-359 doi: www.ijera.com

[12] C. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on System, Man, and Cybernetics*, Part B, vol.26, pp. 29-41, 1996.

[13] *International Journal of Information and Education Technology, Vol. 1, No. 5, December 2011*

= = = = = = = = = = = = = = = = = = The End = = = = = = = = = = = = = = = = = =