

PEER-PEER FILE DISTRIBUTION SYSTEM

(Windows implementation)

By

Irfan Haider Jaffery (MCS - 244011-005)



Supervised by
Dr. MAHMOOD ANWAR KHAN

Department of Computer Science
Bahria Institute of Management and Computer Science
Bahria University, Islamabad

PEER-PEER FILE DISTRIBUTION SYSTEM

(Windows implementation)

By

Irfan Haider Jaffrey (MCS - 244011-005)



Supervised by
Dr. MAHMOOD ANWAR KHAN

A report has been submitted to the department of Computer Science,
Bahria Institute of Management and Computer Science, Islamabad
In partial fulfillment of requirement for the degree of MCS.

Department of Computer Science
Bahria Institute of Management and Computer Science
Bahria University, Islamabad



In the name of Allah Most Gracious Most Merciful

Dedicated

To

Our Parents

For all the support and guidance they have provided to us throughout our lives

&

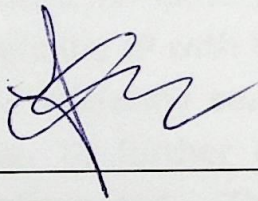
To

Our Teachers

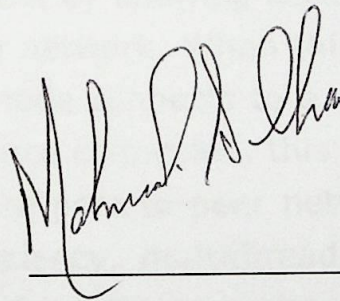
For their guidance in laying the framework for our careers

Certificate

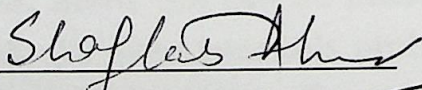
We accept the work confirming to the requirement standard for the partial fulfillment of degree of MCS in the subject of Computer Science.



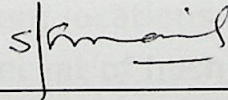
Head of Department
Prof. Fazal Wahab



Supervisor
M. A. Khan



Internal Examiner
Mr. Shaftab Ahmed



External Examiner
Dr. Syed Ismail Shah

ABSTRACT

Peer-to-Peer file distribution system was developed for a private intranetwork. The objective of this software was to efficiently allow access to company files over a LAN, as well as Internet, while at the same time reducing the load on the server. Initially, all company files were placed on a central server, and access to them was via the server only. With over a million files, and growing, this central access methodology greatly increased traffic to and from the server, and decreased efficiency.

The P2P software overcame this problem by allowing access to desired files from other nodes present on the network. When this software is run on a client computer, this client node connects to a **super node**, and registers its IP with this node. Once connected, this software can be used to efficiently search the entire peer-to-peer network for any desired file. To further enhance efficiency, multithreading has also been implemented. Thus, a single file can be downloaded simultaneously from a number of different nodes. Also, before downloading each node is checked for bandwidth efficiency. A file is downloaded from a node only if the latter fulfills the bandwidth criteria. Same file can be present at different locations under different names. To overcome this problem, the concept of hash code has been introduced into this software. That is a unique 128-bit hash code is generated for each file on the basis of file contents. Thus, two files have the same contents, but different names, will have same hash codes. On the other hand, if the contents of the files are changed, their hash codes will also change.

The concept of fallback server is also introduced in this software. The name fallback server has been given to the central server that originally hosted all the files. The idea behind this is that in case a file cannot be found on the peer-to-peer network, the last resort will be to download it from the fallback server. This server will always hold a copy of all files present on the network.

ACKNOWLEDGEMENT

First and foremost, we would like to thank the **Almighty Allah** who helped us accomplished yet another task in our lives. And who has provided us with so many precious gifts that we don't usually find the time to thank him.

Secondly, thanks goes out to our parents who bore with us during our tough studies, and encouraged us while making this project.

We would also like to thank you supervisor, **Mr. Mahmood Anwar Khan**, for his guidance, help and co-operation during the course of this project.

Irfan Haider Jaffery

List Of Figures

<u>Figure #</u>	<u>Description</u>	<u>Page #</u>
Figure 1.1	Servent-to-Servent Communication	11
Figure 2.1	Super Node	21
Figure 3.1	Sequence Diagram	33
Figure 3.2	Class Diagram	35
Figure 3.3	Flow Diagram	37
Figure 4.1	System level Diagram	40
Figure 5.1	Use Cases	54
Figure U-1	Connect Button	65
Figure U-2	Connection Establish	66
Figure U-3	Disabling The Share Folder	67
Figure U-4	Enabling The Share Folder	68
Figure U-5	View Options	69
Figure U-6	Display of Files in Shared Folder	70
Figure U-7	Search Option	71
Figure U- 8	Search File Display	72
Figure U- 9	Bandwidth Status	73
Figure U- 10	Bandwidth Output	74

Table of Contents

<i>Content</i>	<i>Page #</i>
Abstract	i
Acknowledgement	ii
List Of Figures	iii
1 Network and File Sharing History	1
1.1 Introduction	2
1.2 Peer To Peer File Sharing	2
1.2.1 Introduction	3
1.2.2 P2P Overview	4
1.2.3 File Sharing Historical Overview	5
1.2.4 Problems and Issues	5
1.3 Gnutella Architecture	6
1.3.1 Terminology	6
1.3.1.1 Terminology	6
1.3.1.2 Searching	7
1.3.1.3 Downloading	8
1.3.1.4 Time To Live (TTL)	8
1.3.1.5 Difference From Napster	9
1.3.1.6 Similarities With Napster	9
1.3.1.7 Servent-to-Servent Communications	10
1.3.1.8 The Packet Header	12
1.3.1.9 Functions	12
1.3.1.10 Packet Routing	13
1.3.1.11 An important Note on Anonymity and Tracking	14
1.3.1.12 Search Query Proofing	15
1.3.1.13 Search Result Proofing	15
1.3.2 Examples Software available	16
1.3.2.1 Kaaza	16
1.3.2.2 Imesh	16
1.3.2.3 Napster	17
2 Requirement Analysis	19
2.1 Introduction	20
2.2 Scope Of the Project	20
2.2.1 Fall Back Server	20
2.2.2 Super Node	20
2.2.3 Bandwidth Check	20
2.2.4 MD 5	20
2.2.4.1 Terminology and Notation	22
2.2.4.2 MD5 Algorithm Description	23
2.2.4.2.1 Append Padding Bits	23
2.2.4.2.2 Append Length	24

2.2.4.2.3	Initialize MD Buffer	24
2.2.4.2.4	Process message in 16-Word Blocks	24 25
2.2.4.2.5	Output	28
2.2.4.2.6	Summary	28
2.2.4.2.7	Search By File Name and Hash Code	29
2.2.4.3	Multithreading	29
2.2.4.4	Node Traversal	29
3	System Design	30
3.1	Introduction	31
3.1.1	Object Oriented Design	31
3.1.2	The Unified Modeling Language	31
3.2	Sequence Diagrams	32
3.2.1	Overview	32
3.3	Class Diagrams	33
3.3.1	Overview	34
3.4	Flow Chart	36
3.4.1	Flow Diagram Peer to Peer Application	36
4	Implementation	39
4.1	Introduction	40
4.2	High Level System Diagram	40
4.3	Tools Used	41
4.4	Implementation of MD5 Class	41
4.5	Structures defined	42
4.6	Global structure definitions	44
4.7	Implementation of class Node	45
4.8	Implementation of Server	46
4.9	Implementation of Client	50
5	Testing and Debugging	52
5.1	Introduction	53
5.2	Use Case Diagram	53
5.3	Use Cases Description	55
5.3.1	Get File	55
5.3.2	Connect	55
5.3.3	ShareFile	55
5.3.4	Find File	55
5.3.5	Download File	55
5.3.6	VerifyFile	58
5.3.7	Delete File	59
5.3.8	Get Bandwidth Stats	59
5.3.9	Disconnect	60
5.4	Test Plan	61
5.4.1	White Box Testing	61
5.4.2	Black Box Testing	61

5.5	System Acceptance Testing	62
5.6	Debugging	62
5.6.1	Syntax Errors	62
5.6.2	Logical Errors	62
5.7	Testing Strategy	63
5.8	Stress Testing	63
5.9	Validation	63
5.10	Verification	63
5.11	System Evaluation	63
Appendix A User Manual (Windows Application)		64
References		75