Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Peer-to-Peer Communication Service and Messaging System

Nitumani Sarmah[1], Debabrat Sharma[2], Samiksha Dutta[3]
[1]*Asst. Professor, Department of Computer Science & IT, Cotton College State University*
[2]*Department of Computer Science & IT, Cotton College*
[3] *Department of Computer Science & IT, Cotton College*

Abstract - The peer-to-peer communication services[1] has entered into the public limelight over the last few years. Several researches are underway on peer-to-peer communication technologies, but no definitive conclusion is currently available. Comparing to traditional server client technology on the Internet, the peer-to-peer technology has capabilities to realize highly scalable, extensible and efficient distributed applications. Our work presents an anonymous peer-to-peer (P2P) messaging system. A P2P network consists of a large number of peers interconnected together to share all kinds of digital content. A key weakness of most existing P2P systems is the lack of anonymity. Without anonymity, it is possible for third parties to identify the participants involved. First, anonymous P2P system should make it impossible for third parties to identify the participants involved. Second, anonymous P2P system should guarantee that only the content receiver knows the content. Third, anonymous P2P system should allow the content publisher to plausibly deny that the content originated from him or her.
Keywords – peer to peer service, P2P system, client-server model.

I. INTRODUCTION

In a peer-to-peer (P2P) network, every machine plays the role of client and server at the same time. Although a P2P network has a number of advantages over the traditional client-server model in terms of efficiency and fault-tolerance, additional security threats can be introduced. Users and IT administrators need to be aware of the risks from propagation of malicious code, the legality of downloaded content, and vulnerabilities within peer-to-peer software. Security and preventative measures should be implemented to protect from any potential leakage of sensitive information and possible security breaches. Within corporate networks, system administrators need to ensure that peer-to-peer traffic complies with the corporate security policy. In addition, they should only open a minimal set of firewall ports to allow for such traffic. For end-users and/or home users, precautions must also be taken to avoid the possible spread of viruses over peer-to-peer networks. These applications are classified as peer-to-peer because of the elimination of servers to mediate between end systems on which the applications run, and their network behavior is described as an overlay network because the peer protocols form a virtualized network over the physical network.

II. OBJECTIVE AND SCOPE OF OUR WORK

P2P overlay networks which connect peers on top of physical networks like IP, and be it over a wired or over a wireless medium, are growing dramatically in their usage.
We have tried to achieve the following objectives that are critical in improving the performance file sharing P2P overlays:

1. To analyze existing protocols for location management (lookup problem) in peer-to-peer overlays and propose efficient approaches to reduce overlay maintenance cost and network congestion. The main goal is to develop lookup algorithms to optimize the decision making process at each peer by considering the locally available information at each peer like files uploaded/downloaded, bandwidth available, etc. and global information available like type of files at other peers, congestion in the network, replica availability at other peers etc.

2. To propose novel algorithms to safeguard against Sybil attacks in Peer-to-Peer[2] unstructured overlay networks. Sybil attacks are a major source of concern in any collaborative activity, and particularly in a file sharing P2P network. These attacks are mounted on a sophisticated scale where the attacker stills multiple identities and uses those to disturb the collaborative work. Using Sybil attack, attacker can spoil byzantine consensus, can drop forwarding packets to the neighbor peer, can forward the packets in a wrong route etc. Our objective in this work is to either detect these Sybil identities or to reduce their impact on the overlay services. Using JXTA[3] protocols we can, up to some extend restrict the malicious attack by hacker, crackers.

III. THEORETICAL BACKGROUND

The characteristics of our P2P approach is: network address translation (NAT) piercing ability, democracy, no outside control, full decentralization. In the next section we discuss differences with related work. After introducing the main concepts, section Protocol describes in detail the mechanism used by peers for incentive chat. Skype (Baset & Schulzrinne 2004)[4] is a P2P application based on the Kazaa architecture for voice over IP[5] (VoIP) and instant messaging (IM). It offers multiple services, such as: (a) VoIP allows two Skype users to establish two-way audio streams with each other and supports conferences of multiple users, (b) IMallows two or more Skype users to exchange small text messages in real-time, and (c) file-transfer allows a Skype user to send a file to another Skype user.

3.1 Architecture Overview

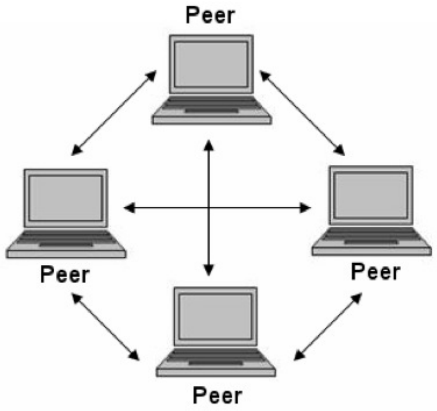

Fig. 1 Peer-to-Peer Architecture

The proposed mobile peer-to-peer architecture is shown in Fig. 1. All of the peer-to-peer communication entities that have a common set of interest and obey a common set of policies construct one peer-to-peer community. This architecture consists of the following basic components

3.1.1 Peer-to-peer node

The peer-to-peer node is an independent communication entity in the peer-to-peer network. It can be a mobile device, a PDA, a personal computer, a server or a workstation, or any of a variety of devices.

3.1.2 Mobile proxy

The mobile proxy is a function in a node, which acts as a proxy for the mobile devices with constrained capability, so that these mobile devices can join the peer-to-peer architecture. Based these basic components, the hybrid peer-to-peer network administrated by a control node (central point) and the pure peer-to-peer network without a control node, are defined.

3.1.3 Pure peer-to-peer architecture

There are only peer-to-peer nodes in the pure peer-to-peer architecture. The proposed pure peer-to-peer architecture is shown in Fig. 1 (a). The connection between peer-to-peer nodes is established on their mutual trust.

Each peer-to-peer node is an independent entity and can participate in and leave the peer-to-peer network at its convenience. Messages are sent from a peer-to-peer node to another one directly or by passing them via some intermediary peer-to-peer nodes.

Fig 1(a) : Pure P2P Architecture

3.1.4 Hybrid peer-to-peer architecture

The hybrid peer-to-peer architecture resolves the disadvantages of the pure peer-to-peer architecture such as inefficient routing, splits of network and lack of security, by introducing a control node. The proposed hybrid peer-to-peer architecture is shown Fig. 1 (b). In our architecture, the control node provides the functions for providing routing information to a destination node, discovering the first peer-to-peer node, recovering from the splitting of the peer-to-peer network, improving the network topology and security such as authentication, in order to improve the inefficiency of the pure peer-to-peer architecture.

Fig. 1(b): Hybrid P2P Architecture

3.2 Protocol Overview

Fig.2 . shows the protocol stack. The protocols have been designed over HTTP, TCP and Bluetooth in two layers. P2P Core Protocol is defined to process peer-to-peer message based on peer-to-peer communication model, six protocols realizing peer-to-peer multicast, communication with a control node and control of a peer-to-peer session and so on are defined over the P2P Core Protocol. Based on the layered approach of the protocol design, it is easy to design a new P2P application protocol based on the requirements of peer-to-peer applications.

Furthermore, the proposed protocols are defined using XML. Since XML has a capability to design general tree structured data, it is possible to design complicated protocol messages required by peer-to-peer applications, and layered protocols independently using XML Namespace. Therefore XML is well suitable to design such an application protocol.

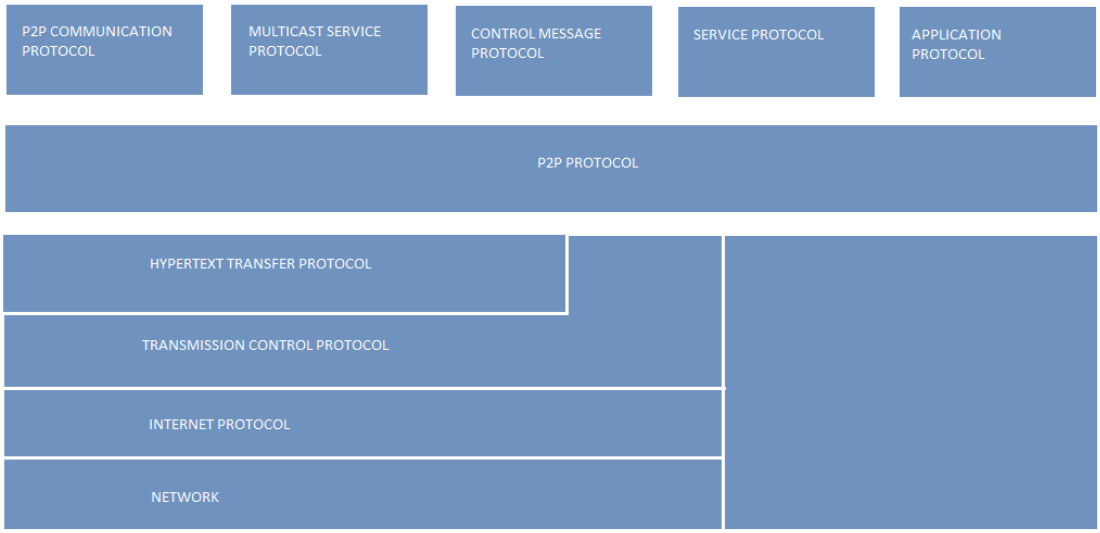

Fig. 2 : Protocol Stack

3.3 P2P Protocols

The P2P Core Protocol has been designed to process a peer to peer message according to peer-to-peer communication model. We defined three message types to realize peer-to-peer communication. Request and response messages are defined for the reactive communication mode and advertise message is defined for proactive communication mode. Additionally, we defined three communication types: unicast, multicast and broadcast.

A P2P Java platform called JXTA was released in 2001 and is widely used for a variety of P2P applications. It can be relatively easily configured for communication within groups defines as subsets of a unique group structure in charge of managing their identifiers. It also provides the needed NAT support. Various flavors of P2P platforms for social networks are provided as both open source and close source applications: Bittorrent, PeerSon.

3.4 Napster Protocol

Due to the fact that Napster is not an open source application, it was only possible to build up a similar application to reveal the Napster protocol through reverse-engineering[6]. Napster works with a central server which maintains an index of all the MP3 files of the peers. To get a file you have to send a query to this server which sends you the port and IP address of a client sharing the requested file. With the Napster application it is now possible to establish a direct connection with the host and to download a file.

The Napster protocol also uses a whole lot of different types of messages. Every state of the hosts, acting like clients towards the server, is related to the central Napster server. Thus the Napster protocol makes anonymity impossible. At first, one may think that this is a drawback, but this complex protocol actually makes a lot of services possible.

3.4.1 Napster Message Data Structures

The format of each message that flow to/from the Napster central server is shown below :

Length	Function	Payload

Where:

- Length specifies the length of the payload.
- Function defines the message type of the packet (see next paragraph)
- Payload this portion of the message is a plain ASCII string.

3.4.2 Initialisation

A registered Napster host, acting like a client, sends to the server a LOGIN message with the following format:

Hostname	Password	Port	Client_Info	Link_type

Where:

- Host & Password identifies the user
- Port is the port which the client is listening on for data transfer.
- Client_Info is a string containing the client version info.
- Link_Type is an integer indicating the client's bandwidth.

3.4.3 Client Notification of Shared File

With the Client Notification of Shared File message the client sends successively all the files it wants to share. It uses the following message notification format:

File Name	MD5	Size	Bitrate	Freq	Time

Where:

- Filename is the name of the file to be shared.
- MD5 (Message Digest 5) is the hash value of the shared file. The MD5 algorithm produces a 128-bit "fingerprint" of any file. It is nearly computationally infeasible to produce two messages having the same hash value. The MD5 algorithm is intended to provide any user the possibility to secure the origin of his shared file, even if the file is laying on drives of other Napster users.
- Size is the file size in bytes
- Bit-rate is the bit rate of the MP3 in kbps (kilobits per second)
- Frequency is the sample rate of the mp3 in Hz (Hertz)
- Time is the duration of the music file in seconds

3.4.4 File Request

The downloading client will first issue either a Search or Browse. The first search message has the following format:

Artist Name	Title	Bit rate	Max Result	Line Type	Freq

Where:

- Artist Name is the name of the artist of the MP3 song.
- Title is the title of the MP3 song.
- Bit-rate is the range of bit-rates to be used.
- Max Results is the maximum number of results.
- Link-Type is the range of link-types.

- Frequency Range is the range of sample frequencies in Hz.

3.4.5 Response and Browse Response

The server answers respectively with a Search Response or a Browse Response with the formats given below:

File Name	MD5	Size	Bit Rate	Freq	Time	User	IP	Link Type

Where:

- Filename is the name of the file that is found.
- MD5 is the hash value of the requested file
- Size is the file size in bytes
- Bit-rate is the bit rate of the MP3 in kbps
- Frequency is the sample rate of the MP3 in Hz
- Time specifies the length of the file
- User is to identify the user who shares the file
- IP is a 4 Byte integer representing the IP address of the user with the file.
- Link-Type refers to the Login Message.

IV. OUR SYSTEM DESIGN

In our system we have design a centralized peer to peer system. So we have create a simulated environment of server –client where a peering between the server and the client will be established.

4.1 Designing and coding process :

Establishing a simple server in java require three steps –

1. Create a server socket object. A call to the ServerSocket constructor such as

   ServerSocket s = new ServerSocket(Port, Queuelength);

   where it will register an available port upto no 65536.

   Queuelength is the number of clients that can request connection to the server.

   Socket connection = s.accept();

2. We create socket to the server to connect to the server

   Socket connection = new Socket (ServerAddress, port);

3.We create socket to connect to the server

   Socket connection = new Socket (ServerAddress, port);

V. SYSTEM ANALYSIS AND DESIGN

Open-source peer-to-peer software development takes many forms, ranging from obscure hobbyist projects with only a few users to software programs that have been downloaded tens of millions of times. Many of the most influential peer-to-peer ideas, such as the hydra-headed decentralization of Gnutella[7] or the speedy "swarming"

downloads of BitTorrent, have come from this community. At one end of the spectrum are projects like Shareaza, which are wholly decentralized, without a controlling entity at all that bears legal or financial responsibility. On the other are companies like LimeWire and MetaMachine, both of which have open-source development products, but have revenue-generating businesses based on selling software and associated products. Somewhere in the middle are younger companies such as Aelitis, the French company recently formed by developers of the Azureus BitTorrent client, the most popular piece of software tracked by Sourceforge, a well-trafficked open-source software development hub.

In our system we have design a centralized peer to peer system. So we have created a simulated environment of server –client where a peering between the server and the client will be established.

VI. METHODOLOGY AND PROCESS INVOLVED

There are two major P2P network topologies: centralized topology, and decentralized topology.

In a centralized topology, network functionality depends largely on a central server and each peer accesses the central server to upload and download information.

In a decentralized P2P topology, there is no central server and peers are normally organized in an unstructured fashion.

Our system is a centralized one, where each peer can only communicate directly with a few neighbour peers through a control node; however a peer can communicate with the rest of network through hop-by-hop message propagation. The methodology of the system:

1. A P2P network will have an addressing scheme to identify each individual peer. Each peer in the network is identified by a randomly generated virtual address. Non-static addresses are very useful for achieving anonymity since such addresses can effectively hide peer identity.
2. A new peer can easily discover existing peers via host listing or host caching. Host listing uses a centralized server to maintain a list of active peers.
3. When a new peer wants to join the network, it first publishes in the network.
4. In our centralized P2P networks, queries are propagated hop-by-hop.
5. In order to prevent excessive query flooding, each query has a time-to-live (TTL) field, where the TTL is the number of peers the query is allowed to visit. Each time the query passes from one peer to another, the TTL is decremented by 1. When the TTL reaches 0, the request will be dropped.
6. Most P2P file sharing applications publish file content along with a key. Usually the key is a hash value of the file name or file description, and it is used by queries to search for the published file.
7. Dividing a file into chunks makes it possible for content receivers to speed up the retrieval process by retrieving various parts of the file from multiple peers simultaneously.

8. A storage is managed as an LRU (Least Recently Used) cache. When a new file arrives and causes the storage to exceed its capacity, the least recently used files are removed.

9. File querying is supported by a keyword search engine. Keywords usually consist of filenames or file descriptions.

10. File retrieval can be accomplished by using traditional file transfer protocols, such as FTP or HTTP and JXTA.

11. An anonymous P2P file sharing system also includes receivers and senders. A receiver is a peer that receives published files from the system while a sender is a peer that can send published files to the receiver.

REFFERENCE

[1]Lutz Seidenfaden, Platz der Göttinger Sieben, Björn Ortelbach, Matthias Schumann,"A PEER-TO-PEER APPLICATION SYSTEM FOR THE SCHOLARLY COMMUNICATION" - The 15th European Conference on Information Systems, University of St.Gallen, Switzerland, 7-9 June 2007

[1]PEER Behavioural Research: Authors and Users vis-à-vis Journals and Repositories, Baseline report - Jenny Fry, Charles Oppenheim, Steve Probets, Claire Creaser, Helen Greenwood, Valérie Spezi, Sonya White, September 2009

[2]FANG Qun (Department of Computer Science,Anhui Normal University,Wuhu 241000,China);Developing Method for JXTA-Based P2P Applications[J];Computer Knowledge and Technology;2006-20

[3] DOUCEUR, J. The Sybil Attack. In 1st Intl. Workshop on Peer-to-Peer Systems (2002).

[4] S. Baset and H. Schulzrinne., "An analysis of the skype peer-to-peer Internet telephony protocol" Columbia University Technical Report CUCS-039-04, September 2004.

[5] Salman A. Baset and Henning G. Schulzrinne :"An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol".

[6] Choon Hoong Ding, Sarana Nutanong, and Rajkumar Buyya, "Peer-to-Peer Networks for Content Sharing".

[7] Gayatri Tribhuvan , "A BRIEF INTRODUCTION AND ANALYSIS OF THE GNUTELLA PROTOCOL"

