COMPARATIVE STUDY OF PEER TO PEER NETWORKS AND ITS APPLICATIONS

Jayeeta Biswas[1]
[2]Dept. of IT, Tripura University, Agartala, India

ABSTRACT: In the recent years Peer to Peer applications has received a lot of attention and popularity due to the ongoing legal battle with the music and movie industry. Despite many beliefs it is not a new concept but, in its simplest form, has existed for over four decades and its origin can be traced back to the original implementation of the Internet. Napster, the first p2p based music sharing application has an immense contribution in making the peer to peer network a popular one. In this report, after a detail study of various P2P based application, we have tried to make a comparison among them based on their architecture and how they perform their work. We have also tried to show the application's advantages and disadvantages and their contribution in the advancement of today's Internet.

***Index Terms*:** *Bit-Torrent, Napster, Gnutella, eDonkey.*

1. INTRODUCTION

Peer to peer network is a configuration where two or more computers are connected to transfer and receive various files and folders. A P2P network can also be configured by connecting two or more computers via universal serial bus to transfer files. A P2P network can also establish direct relationship among users by setting permanent infrastructure as well as a large network. [1]

In P2P networks decentralized model is used. In decentralized model each machine is referred to as peer, function as a client with its own layer of server functionality. The role of a client and server is played by the peer alone at the same time. That is, the peer can initiate requests to other peers, and at the same time respond to incoming requests from other peers on the network. It is an advantage over the traditional client-server model as in this model a client can only send requests to a server and then wait for the server's response.

The overall performance of the network decline when there is a increase in request of service by client from the server. In P2P network, network performance improved with the increase in number of peers. These peers can organize themselves into ad-hoc groups as they communicate, collaborate and share bandwidth with each other to complete the tasks at hand (e.g. file sharing). Each peer can upload and download at the same time, and in a process like this, new peers can join the group while old peers leave at any time. This dynamic re-organization of group peer members is transparent to end-users.

P2P network is famous for its robustness. In P2P network, if a peer gets disconnected or goes down then it will not affect the network, the P2P application will continue by using other peers. For example, in a Bit Torrent system, any clients downloading a certain file are also serving as servers. When a client finds one of the peers is not responding, it searches for other peers, picks up parts of the file where the old peer was, and continues the download process while in client-server approach, if the server goes down the entire system will go down.

P2P networks can be classified into two types-

1. Pure P2P network.
2. Hybrid P2P network.

1.1. Pure P2P network

In a pure P2P network, the system does not rely on a central mainstream server to help in controlling, coordinating or managing the exchange of information among the peers. In this P2P network, all participating peers are equal and each peer plays the role of both client and server. all participating peers are equal, and each peer plays both the role of client and of server. Examples of a pure P2P network are Gnutella and Free Net.

1.2. Hybrid P2P network

In a hybrid P2P network, to facilitate the P2P service some administrative functions are perform by a central server. For example, in Napster, a server helps peers to "search for particular files and initiate a direct transfer between the clients". Only a catalogue of available files is kept on the server, while the actual files are scattered across the peers on the network. Another example is Bit Torrent (BT), where a central server called a tracker helps coordinate communication among BT peers in order to complete a download.

P2P network can also be classified into the following types-

1. Centralized P2P network
2. Decentralized P2P network

The Decentralized P2P network can be further divided into-

1. Structured
2. Unstructured

1.3. Centralized P2P Network

Centralized P2P systems has the features of both centralized and decentralized architectures. Like a client-server system, there are one or more central servers, which help peers to locate their desired resources or act as task scheduler to coordinate actions among them. To locate resources, a peer sends messages to the central server to determine the addresses of peers that contain the desired resources or to fetch work units from the central server directly. However, like a decentralized system, once a peer has its information, it can communicate directly with other peers (without going through the server anymore). As in all centralized systems, this category of P2P systems are susceptible to malicious attacks and single point of failure. Moreover, the centralized server will become a bottleneck for a large number of peers, potentially degrading performance dramatically. Finally, this type of system lacks scalability and robustness. Some examples of this architecture include Napster and BOINC.[2]

1.4. Decentralized P2P Network

In a decentralized P2P system, peers have equal rights and responsibilities. Each peer has only a partial view of the P2P network and offers data that may be relevant to only some peers. As such, locating peers offering data quickly is a critical and challenging issue. The advantages of these systems are obvious:

(a) They are immune to single point of failure.

(b) Possibly enjoy high performance, scalability, robustness, and other desirable features.

1.5. Structured P2P Network

Structured P2P networks have the characteristic of using some type of algorithm for organization and optimization of the network. To ensure that any type of network can proficiently route a search to some peer that has a desired file, Structured P2P network use some type of global protocol. Example of Structured P2P network is Free net, Chord etc.[2]

1.6 Unstructured P2P Network

Unstructured P2P network do not follow any structured way for file placement and do not optimize the search algorithm. Due to their unstructured way, they flood the queries in network and increase the network congestion. Example of unstructured P2P network is Gnutella, Kazaa etc[2].

2. DHT

Distributed Hash Tables (DHT) are algorithms used in modern peer-to-peer applications, which provide a reliable, scalable, fault tolerant and efficient way to manage P2P networks in a true peer to peer manner.

Distributed Hash Tables promote several ideas that distinguish them from traditional Client-Server oriented services:

1. DHTs provide decentralized operation, without the need to maintain a centralized server to control the P2P network. This also results in a system composed of many parts. In this sense, P2P applications using a DHT are true P2P applications.
2. The system is scalable, meaning that the system functions properly even with very large node count and traffic.
3. Load is balanced fairly among the peer nodes in the network, in a way that does not over encumber any node, which could occur in client and server model.
4. The system is based on the assumption that the network is not static and changes occur fairly frequently with nodes joining and leaving the network. This is called "Churn".
5. Routing and data retrieval is fast, and can be accomplished in logarithmic time.[3]

In this report a detail, we will give a detail literary survey of different P2P networks and their advantages and disadvantages and will try to make comparisons among them

3. INTRODUCTION TO VARIOUS P2P NETWORKS

3.1 NAPSTER

Napster was developed by Shawn fanning in June, 1999 is a hybrid P2P file sharing application that made sharing of MP3 on the internet possible. Napster has a centralized directory (actually several) that describes how files are located in Napster and each host registers with this directory when they join the network. The centralized directories therefore have the IP addresses, the names of the files the hosts want to share and other data stored about each computer system connected to it.[39]

How does it work?

1. Each user must have Napster software in order to join in file transfers. The user runs the Napster program. Once executed, this program checks for an Internet connection.
2. If an Internet connection is discovered, another connection between the user's computer and one of Napster's Central Servers will be created. This connection is made possible by the Napster file-sharing software.
3. The Napster Central Server keeps a record of all client computers connected to it and stores information on them as described above.
4. If a user wants a certain file, they place a request to the Napster Centralized Server that it's connected to.
5. The Napster Server looks up its record to see if it has any matches for the user's request.
6. The Server then sends the user a list of all that matches (if any) it as found including the corresponding, IP address, user name, file size, ping number, bit rate etc.
7. The user chooses the file it wishes to download from the list of matches and tries to establish a direct connection with the computer upon which the desired file resides. It tries to make this connection by sending a message to the client computer indicating their own IP address and the file name they want to download from the client.
8. If a connection is made, the client computer where the desired file resides is now considered the host. The host now transfers the file to the user.
9. The host computer breaks the connection with the user computer when downloading is complete.

ADVANTAGES

1. Napster is easy to use.
2. Files can be found fast and effective.

DISADVANTAGES

1. Central structure eases censorship, hostile attacks and vulnerability against technical issues. Eg. Denial of service (DOS) attack.
2. Napster does not scale i.e growing number of participants implies a decline in performance.

3.2 GNUTELLA

Gnutella (nowadays word "Gnutella" used to refer to open network protocol for file distribution) was originally developed at Nullsoft (AOLs) which is a decentralized peer-to-peer (P2P) network that allows users to share files across the Internet without having to use a central server.[9]

How does Gnutella works?

To make use of the Gnutella protocol and get connected to a Gnutella network, a user first must download and install a Gnutella compatible program. According to Toadnode's FAQ, Gnutella compatible software works as follows:

The protocol defines the way of applications, such as Toadnode, communicate over the Internet. It is helpful to think of the P2P network as a conversation between computers. Some computers "tell" while others "listen". To coordinate this conversation packets are tagged with special descriptors so that each computer receiving the packet knows how to react. The current Gnutella protocol (version 0.4) defines five descriptors: Ping, Pong, Query, QueryHit and Push.

Step 1: Determining Who is on the Network

A "Ping" packet is used to announce your presence on the network. When another computer hears your Ping it will respond with a "Pong" packet. It will also forward your Ping packet to other computers to which it is connected and, in response, they too will send back Pong packets. Each Ping and Pong packet contains a Globally Unique Identifier (GUID). A Pong packet also contains an IP address, port number, and information about how much data is being shared by the computer that sent the Pong. Pong packets are not necessarily returned directly to the point of origin, instead they are sent from computer to computer via the same route as the initial Ping. After sending a Ping to one computer you will start receiving many Pong responses via that one computer. Now that the Pong packets have told you who your active peers are, you can start making searches.

Step 2: Searching

Gnutella is a protocol for distributed search. Gnutella "Query" packets allow you to search by asking other computers if they are sharing specific content (and have an acceptably fast network connection). A Query packet might ask, "Do you have any content that matches the string 'Homer"? This question is sent to all the computers that sent you Pong packets. Each of these computers does two things. First, each computer checks to see if it has any content that matches the search string. In this case it looks to see if there are any files in a specified directory marked "sharable to the outside world" that have the letters "Homer" in its complete file path. Second, each computer sends your Query packet on to all the computers to which it is connected. These computers check their directories and send your Query packet to all their connected computers. This process continues until you run out of computers to ask or until the Query packet gets too old and times out. This last detail is important because without a pre-defined Time To Live (TTL) the Query packet could get bounced around for a very long time, potentially forever. Most servents, including Toadnode, allow you to adjust the TTL. GUIDs in each packet are used to make sure that the same message does not get passed to the same computer again and again, creating a loop.

Step 3: Downloading

By the time you are ready to download, the question you asked in your Query packet has been distributed to a huge number of computers. Each computer has checked its shared information and determined if it is sharing anything that matches "Homer". Let us say that three computers that received your Query packet have a match for "Homer". The last two packet descriptors, called "Query Hit" and "Push" are responsible for content delivery. Each of the three computers will send you a QueryHit packet via the same delivery route, computer-to-computer,that the Query packet originally traveled. The QueryHit packet contains the IP address and GUID of the computer that has the data as well as information about the file that matched your query.

When you receive a QueryHit packet your servant software will display the name of the file for you and give you the option to download. File transfers use the HTTP protocol's GET method directly between your computer and the computer that has the file you want. Normally, your computer will initiate the HTTP connection to the computer that has the file. Occasionally, due to a firewall, you will be unable to initiate a connection directly to the computer that has the file you want. In these cases the "Push" packet is used. The Push packet allows a message to be delivered to the computer that has the file you would like to download via the route that the QueryHit packet originally traveled, except in reverse. The Push packet tells this computer that you would like to download a file but cannot manage to initiate an HTTP connection. This computer then becomes the initiator, attempting to connect directly to you, which often is possible because the firewall between the machines is only limiting connections initiated from outside the firewall.[10]

ADVANTAGES

1. Protocol allows users to transfer files between each other instead of having to search through a main server.
2. Distributed processing with low traffic.

3. Network can work all the time since no key piece of equipment that should it fail would cause the network to be disrupted.
4. Transfer of data is available under most of firewall systems.
5. Flexibility in Query processing.
6. Low nurture costs.
7. Easy access.

DISADVANTAGES

1 .Does not provide quality service due to overloading and downloading failures.
2. Unwillingness of file sharing by the peers.
3. Low downloading speed.
4. Lack of new features and service.
5. High Bandwidth usage.
6. Not guaranteed to search a file.

3.3 BitTorrent

BitTorrent is a technology/protocol which makes the distribution of files, especially large files, easier and less bandwidth consuming for the publisher. This is accomplished by utilizing the upload capacity of the peers that are downloading a file. A considerably increase in downloader's will only result in a modest increase in the load on the publisher hosting the file. BitTorrent is by far the most popular peer-to-peer programs ever. Analysis shows that it accounts for about 35% of all Internet traffic.[12]

How BitTorrent works?

BitTorrent is a **protocol** (a set of rules that different computer systems agree to use) based on P2P that can be used to share large files very efficiently. Suppose the Sharks decide they want to use BitTorrent. They take their music track and make it available on their computer as a file called a **torrent**. The computer that hosts the original file, in its entirety, is called a **seed** and it splits the file up into lots of pieces.

Anyone who wants the file uses a program called a **BitTorrent Client** to request it from a seed. The client is sent one of the pieces and gets all the remaining pieces, over a period of time, from other people's computers through P2P communication. At any given moment, each computer is downloading some parts of the file from some of these peers and uploading other parts of the file to other peers. All the computers cooperating in this way at any time are called a swarm. The more popular a file is, the more computers there are in the swarm and the quicker the process is all round.

Share and share alike is the ethos behind BitTorrent so, when people have finished downloading a file, they are encouraged to stay online for a while so they can continue uploading the file to others in the swarm—an activity known as seeding. Quitting from a swarm the minute your download is complete, without seeding, is a selfish activity that's earned itself the nickname leeching. If everyone leeched, BitTorrent wouldn't work at all.

Although BitTorrent is a decentralized P2P process very different from old, client-server-type downloading, there has to be some sort of order and control. Someone has to keep track of which computers have which bits of the file. This works in different ways with different BitTorrent clients. Some rely on centralized computers called trackers which, as their name suggests, keep track of where all the pieces of the file can be located at any moment. There is also a more decentralized version of BitTorrent where the clients manage the tracking process among themselves (sometimes called tracker less torrents or distributed torrents).[19]

With a P2P BitTorrent

1. The originating server (the seed) makes available one copy of the file, which is then split up into chunks. Different chunks are sent out to the various computers (BitTorrent clients) trying to get hold of a copy of the file.
2. Each client uploads their part of the file to other clients while simultaneously downloading bits of the file they don't have from other clients. All the clients work together as a swarm to share the file. The file-sharing process doesn't happen in the systematic, sequential way we show here (purely for simplicity): clients upload and download simultaneously and the file actually builds up in a more random way. There are often hundreds of clients involved in each swarm.
3. Eventually, every client receives a complete copy of the file. However, in this example, as in real life, one client (lower left) finishes downloading before the others. If the owner of that machine switches off as soon as they're done, the other clients will never receive complete copies of the file.[29]

ADVANTAGES

1. Highly efficient.
2. Makes sharing easier.
3. Prevents tampered or broken files from being shared.
4. Ability to upload and download large files at short amount of time.
5. Free uploading and downloading.

DISADVANTAGES

1. If the seeder leaves the swarm too early, no one is able to use the file.
2. Seeder can only seed one or two files at a time.
3. The computer's program drops drastically while using the program.
4. Old and unpopular files are hard to find.
5. Torrent files are too flaky due to excessive popular

3.4 eDonkey

The eDonkey network is a decentralized hybrid peer-to-peer file-sharing network with client applications running on the end-system that are connected to a distributed network of dedicated servers. Contrary to the original Gnutella protocol it is not completely decentral as it uses servers; contrary to the original Napster protocol it does not use a single server (farm) which is a single point of failure, instead it uses servers that are run by power users and offers mechanisms for inter-server communication. Unlike super-peer protocols like KaZaa, or the modern Gnutella protocol the eDonkey network has a dedicated client/server based structure. The servers are slightly similar to the KaZaa super-nodes, but they are a separate application and do not share any files, only manage the information distribution and work as several central dictionaries which hold the information about the shared files and their respective client locations.[32]

How it works?

In the eDonkey network the clients are the only nodes sharing data. Their files are indexed by the servers. If a client wants to download a file or a part of a file, it first has to connect via TCP to a server or send a short search request via UDP to one or more servers to get the necessary information about other clients offering that file. The eDonkey network is using 16 byte MD4 hashes to (with very high probability) uniquely identify a file independent of its filename. The implication for searching is that two steps are necessary before a file can be downloaded in the eDonkey network. First, a full text search is made at a server for the filename, it is answered with those file hashes that have a filename associated which matches the full text search. In a second step, the client requests the sources from the server for a certain file-hash. Finally, the client connects to some of the sources to download the file.[35]

ADVANTAGES

1. eDonkey servers act as communication hubs for the clients, allowing users to locate files within the network.
2. It is best suited to share large files.

DISADVANTAGES

1. It is subjected to heavy traffic.
2. It is more vulnerable to malicious attacks.

VII. CONCLUSION

While P2P networks open a new way for efficient downloading and sharing of files and data, yet there are lots of legal and security problems that need to be solved to make the various P2P networks and applications more work friendly to the users.P2P networks and applications still has a long way to go to become a real success story. A lot of problems still need to be solved and a lot of developments still need to be made to make peer to peer networks a real force to beckon.

4. .REFERENCES

[1]https://www.computerworld.com/article/2588287/networking/peer-to-peer-network.html.

[2] https://www.infosec.gov.hk/english/technical/files/peer.pdf.

[3]http://www.cs.princeton.edu/courses/archive/spr11/cos461/docs/lec22-dhts.pdf.

[4] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems," Proc. Middleware, 2001.

[5] S. Ratnasamy et al., "A Scalable Content Addressable Network," Proc. ACM SIGCOMM, 2001, pp. 161–72.

[6] I. Stoica, R. Morris et al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," IEEE/ACM Trans. Net., vol. 11, no. 1, 2003, pp. 17–32.

[7] B. Y. Zhao et al., "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," IEEE JSAC, vol. 22, no. 1, Jan. 2004, pp. 41-53.

[8] Napster, available at http://www.napster.com/

[9] https://www.techopedia.com/definition/445/gnutella

[10] https://computer.howstuffworks.com/file-sharing2.html

[11] (2002) Gnutella ultrapeers, available at http://rfcgnutella.sourceforge.net/developer/testing/Ultrapeers_1.0.html

[12] web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf

[13] B. Karp et al., "Spurring Adoption of DHTs with OpenHash, a Public DHT Service," Proc. 3rd Int'l. Wksp. Peer-to-Peer Systems (IPTPS 2004), Berkeley, California, USA, Feb. 26-27, 2004.

[14] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," Proc. IPTPS, Cambridge, MA, USA, Feb. 2002, pp. 53–65.

[15] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A Scalable and Dynamic Emulation of the Butterfly," Proc. ACM PODC 2002, Monterey, CA, USA, July 2002, pp. 183–92.

[16] P. Francis, "Yoid: Extending the Internet Multicast Architecture," unpublished, Apr. 2000. http://www.aciri.org/yoid/docs/

index.html [17] J. Kubiatowicz et al., “Oceanstore: An Architecture for Global-Scale Persistent Storage,” Proc. ACM ASPLOS, Nov. 2002.

[18] W. J. Bolosky et al., “Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs,” Proc. 2000 ACM SIGMETRICS Int’l. Conf. Measurement and Modeling of Comp. Sys., June 2000, pp. 34-43.

[19] www.explainthatstuff.com/howbittorrentworks.html

[20] D. Karger et al., “Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web,” Proc. 29th Annual ACM Symp. Theory of Comp., May 1997, pp. 654–63.

[21] National Institute of Standards and Technology (NIST), “Secure hash standard,” U.S. Department of Commerce, National Technical Information Service FIPS 180-1, Apr. 1995.

[22] F. Dabek et al., “Wide-area Cooperative Storage with CFs,” Proc. 18th ACM Symp. Operating Systems Principles, 2001, pp. 202–15.

[23] R. Cox, A. Muthitacharoen, and R. Morris, “Serving DNS using Chord,” Proc. First Int’l. Wksp. Peer-to-Peer Systems, Mar. 2002.

[24] Y. Rekhter and T. Li, An Architecture for IP Address Allocation with CIDR, IETF Internet draft RFC1518, available at http://rfc.net/rfc1518.html, 1993.

[25] S. Rhea et al., “Maintenance-free Global Storage in Oceanstore,” IEEE Internet Comp., 2001.

[26] L. Peterson et al., “A Blueprint for Introducing Disruptive Technology into the Internet,” SIGCOMM Comp. Commun. Rev., vol. 33, no. 1, 2003, pp. 59–64.

[27] S. Q. Zhuang et al., “Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination,” Proc. 11th Int’l. Wksp. Network and Op. Sys. Support for Digital Audio and Video, 2001, pp. 11-20.

[28] F. Zhou et al., “Approximate Object Location and Spam Filtering on Peer-to-Peer Systems,” Proc. Middleware, June 2003.

[29] http://www.explainthatstuff.com/howbittorrentworks.html

[30] M. Castro et al., “SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Infrastructure,” IEEE JSAC (special issue on Network Support for Multicast Communications), October 2002.

[31] M. Castro et al., “Splitstream: High-Bandwidth Multicast in Cooperative Environments,” Proc. 19th ACM Symp. Operating Systems Principles, Oct. 19–20, 2003, pp. 298–313.

[32] citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.8596&by O Heckmann

[33] P. Druschel and A. Rowstron, “PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility,” Proc. 8th Wksp. Hot Topics in Op. Sys. (HotOS-VIII). Schloss Elmau, Germany: IEEECompSoc, May 2001.

[34] A. Rowstron and P. Druschel, “Storage Management and Caching in Past, a Large-Scale, Persistent Peer-to-Peer Storage Utility,” Proc. 18th ACM Symp. Operating Systems Principles, Oct. 2001, pp. 188-201.

[35] ftp://130.83.198.178/papers/HBMSt04_707.pdf

[36] A. Muthitacharoen, B. Chen, and D. Mazières, “A Low-Bandwidth Network File System,” Proc. 18th ACM Symp. Op. Sys. Principles, 2001, pp. 174–87.

[37] U. Manber, “Finding Similar Files in a Large File System,” Proc. USENIX Winter 1994 Conf., Jan. 1994, pp. 1–10.

[38] H. J. Siegel, “Interconnection Networks for SIMD Machines,” Computer, vol. 12, no. 6, 1979, pp. 57–65.