IJRSET AUGUST 2014 Volume 2, Issue 2 Pages: 74-83

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

# A SURVEY ON PEER TO PEER (P2P) NETWORK IN A DISTRIBUTED ENVIRONMENT

[1] P. Arockia Mary, [2] M. Radhakrishnan
[1] Department of Information Technology, [2] Director/IT,
[1, 2] Sudharsan Engineering College, Sathiyamangalam,

**ABSTRACT:** The term "peer-to-peer" (P2P) alludes to a class of frameworks and applications that utilize disseminated resources to Play out a basic function in a decentralized way. With the unavoidable pervasive deployment of PCs, P2P is progressively accepting consideration in research, item advancement, and venture circles. Peer-to-Peer frameworks depend on the idea of assets confinement and mutualisation in element setting. In particular environment, for example, portable systems, described by high changeability and dynamicity of system conditions and exhibitions, where hubs can join and leave the system progressively, assets unwavering quality and accessibility constitute a basic issue. The asset disclosure issue emerges with regards to peer to peer (P2P) systems, where at any point of time a peer might be put at or expelled from any area over a universally useful system. Finding an asset or administration effectively is a standout amongst the most vital issues identified with peer-to peer systems. This paper exhibits a review on P2P systems of grouping, applications, and it stages.

**KEYWORDS: [P2P, Routing, Complexity, Design, performance in peer to peer]**

## 1. INTRODUCTION

Peer-to-peer (P2P) network is prevalent and its core Idea is that each peer in the network acts as both client and server. They can exchange data directly with other peers. The openness, anonymity, dynamic of P2Pnetworks, doomed that they are unauthentic networks. The main challenge in P2Pcomputing is to design and implement a robust distributed system composed of distributed and heterogeneous peer nodes, located in unrelated administrative domains. In atypical P2P system, the participants can be “domestic” or “enterprise” terminals connected to the Internet. “P2P allows file sharing or computer resources and services by direct exchange between systems” or allows the use of devices on the Internet periphery in a non client capacity. Also, it could be defined through three key requirements: **a)** they have an operational computer of server quality, **b)** they have a DNS independent addressing system” and **c)** they are able to scope with variable connectivity. Also, as defined in: P2P is a class of applications that takes advantage of resources-storage, cycle, content, human presence-availability at the edges of Internet. Because accessing to these decentralized

resources means operating in environment with unstable [1] Connectivity and unpredictable IP addresses. P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers.

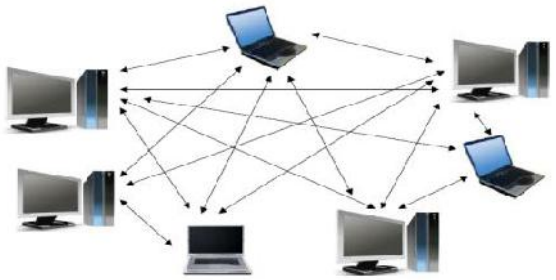

**Figure1- Peer To Peer Network**

In P2P networks all peers cooperate with each other to perform a critical function in a decentralized manner. All peers are both users and providers of resources and can access each other directly without intermediary agents. Compared with a centralized system, a P2P system provides an easy way to aggregate large amounts of resource residing on the edge of Internet or in ad-hoc networks with a low cost of system maintenance. P2P systems attract increasing attention from researchers recently. Such architecture and systems are characterized by direct access between peer systems, rather then through a centralized server [2]. More simply, a P2P network links the resources of all the nodes on a network and allows the resources to be shared in a manner that eliminates the need for a central host. In P2P systems, nodes or peers of equal roles and responsibilities, often with various capabilities, exchange information or share resources directly with each other. P2P systems can function without any central administration and coordination instance. A P2P network differs from conventional client/server or multi tiered server's networks.

## 2. HISTORY OF P2P NETWORK

Peer-to-peer networking essentially started with the ARPANET, which began as a venture, through DARPA, to network together computers that were under contract with them in 1966. Larry Roberts successfully connected computers at UCLA, SRI, UCSB, and UoU, in 1969 utilizing a new method of data transfer called “packet switching [13].” The network could utilize email, file transfers, and remote logons. The ARPANET started out as an experiment to connect computers together but it was continually funded because of military interests [14].

The ARPANET allowed computers to send and receive information from other nodes on the network by the use of packet switching. Packet switching allowed information to be sent by breaking up the information into smaller parts called packets. Along with the packet a destination message was sent, telling the intermediate nodes on the network where the packets where supposed to end up. This method is analogous to P2P networking today because information can hop between nodes in order to end up at the correct destination and the network for the most part is decentralized. Version two ARPANET added the TCP/IP protocols which were used in the virtual network setup to connect multiple nodes together, regardless of their hardware configurations. The ARPANET eventually lead to the NSF net which successfully connected 2000 university computers together. These are the antecedents to the internet of today.

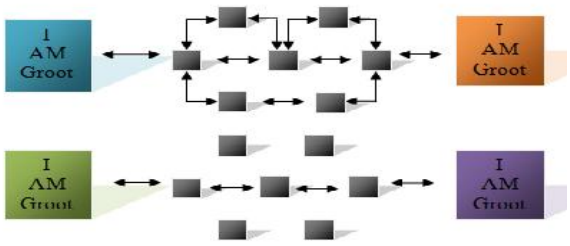


**Figure2- Packet Switching vs. Circuit Switching**

## NAPSTER

The first large P2P network scheme and application that the public used was Napster. It was an application that allowed users to share MP3 files with each other that was officially operational from 1999 to 2001. Napster utilized a hybrid centralized P2P model where a centralized server facilitated the pairing of two users based off of files desired and files owned closely resembling Figure 2 [15].

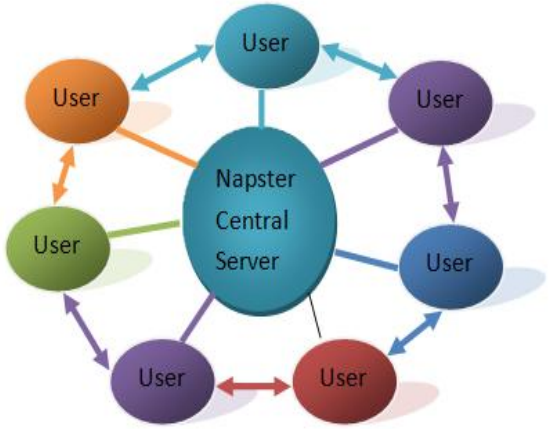


**Figure 3- Napster's centralized P2P architecture**

Napster led to many clone applications that utilized P2P networking, usually with the intention of sharing files and not just MP3's. Not only were clones developed after the initial success of Napster, other P2P architecture networks were developed and adapted, hoping to create a more efficient and secure network.

## GNUTELLA

Gnutella began in early 2000 at a subsidiary of America Online. The development was scrapped shortly after, but not before a few users downloaded it, leading to the reverse engineering of the application and its widespread use.

The Gnutella protocol does not rely on a central server to handle user queries; instead it utilizes a "flat ad-hoc topology [16]." Every user, or node, in the system acts as a servent, both a server and a client, able to respond to queries from neighboring nodes and issue queries to neighboring nodes as can be seen in Figure 3. The network can also be classified as a decentralized P2P network where every node is connected to many other nodes. This architecture insures the networks survivability, if one node goes offline, the whole network does not suffer.

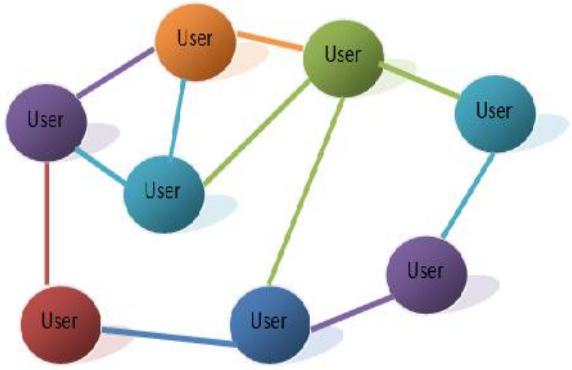


**Figure 4- Gnutella's decentralized P2P architecture**

## BIT TORRENT

The Bit Torrent protocol was developed in 2001 by Bram Cohen, a University of Buffalo student at the time. Bit Torrent's network architecture is much like Gnutella's, it is a decentralized P2P network where none of the file transfers occur within the protocol itself. In order for users to find other users, they must use a "tracker" which is a list of IP addresses of users sharing a certain file [17]. In order for users to download a specific file they have to find a tracker of it, which can usually be found on tracker websites like The Pirate Bay. The files you download from tracker websites include a tracker file and a torrent file. The torrent file includes the number of pieces and blocks a file contains, the IP address and port number of the tracker, and also the SHA1 hash tables of the pieces for the file. The SHA1 hash tables allow users to "verify the integrity" of each piece downloaded [18]. Files shared through a Bit Torrent network have to be broken up into "pieces" and "blocks." Typically a piece is 512 kBytes and a block is 16 kBytes. When downloading files from

neighboring users, the protocol allows for downloading several pieces in parallel, utilizing many neighbors at once. A user can either be a "leecher" or a "seeder" within the Bit Torrent protocol. A leecher is a user who has not completely downloaded the file associated with the tracker he is a part of. A seeder is a user is has a complete version of the file and is sharing it with leechers.

In order to address the inefficiency problems that Gnutella suffered due to users who did not participate in sharing files, BitTorrent's protocol favors users who upload files. Seeders periodically check the upload rates of its neighbors and only share with those who also upload. This "chokes" users who do not upload, forcing them to have slower download speeds or even restricting them from downloading from that tracker [18].

## 3. P2P NETWORK ARCHITECTURE

P2P network architecture can be classified by their "degree of centralization", i.e. to what extent they rely on one or more servers to facilitate the interaction between peers. Three categories are identified as:

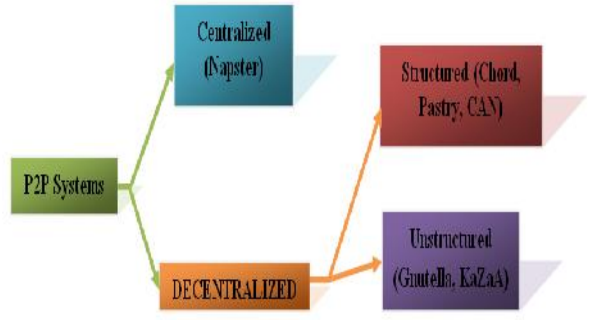


**Figure 5- Classification of Peer-to-Peer Architecture**

**Purely decentralized systems** - All nodes in the network perform exactly the same tasks, acting both as servers and clients, and there is no central coordination of their activities. The nodes of such networks are often termed "SERVENTS" (SERVers+ clients). Example networks are original Gnutella architecture and Free net. This kind of architecture requires all peers to act as both a client and a server. All nodes are therefore "assigned" the same tasks and there is no central point of failure [3].

The nature of these networks requires that the index cannot be stored on a central server. This means that it can be either distributed or local. Networks which adopt the local index logic require each node to hold an index for its own content. As an early P2P implementation for a distributed index network was Freenet. The early versions of Gnutella (v0.4) used a locally stored index which is however a terribly inefficient technique [4] [5]. In order for a peer to find available content in the network the peer has to flood the whole network by broadcasting its request and wait for a response from the nodes that have the required content. This of course renders the network almost unable to scale to larger sizes. However, this also means that the network is very fault tolerant and if a node fails or just disconnects this would not affect the network. When a node wants to join the network the only requirement is for it to connect to any existing and active peer. Some techniques to improve the scalability issues will be discussed later in the thesis. It is important to note that under this classification, besides the Gnutella-type networks, there are other networks also that are formed deterministically. Such networks form connections between their peers that are somehow organized, structured. However, "structured" in this case does not refer to the network topology but merely to the fact that peers do not join the network at random locations, as with Gnutella for example, but deterministically take a position in the decentralized network[6][7].

**Partially centralized systems** -The basis is same as the one with purely decentralized systems. However, some of the nodes are assumed to play a more "important" role than the rest of the nodes, acting as local central indexes for files shared by local peers. These nodes are called "Super nodes", and the way in which they are selected for these special

tasks vary from system to system shown in figure FF. It is important to note that these Super nodes do not constitute single points of failure for a p2p network, since they are dynamically assigned and in case they are subject to failure or malicious attack the network will take action to replace them with others. Example networks are Kazaa, Morpheus and more recent Gnutella.

Generally peers are automatically elected to become super nodes if they have sufficient bandwidth and processing power. Super nodes index the files shared by peers connected to them, and proxy search requests on behalf of these peers. All queries are therefore initially directed to super nodes. Two major advantages of partially centralized systems are that:

• Discovery time is reduced in comparison with purely decentralized systems, while there still is no unique point of failure. If one or more super nodes go down, the nodes connected to them can open new connections with other super nodes, and the network will continue to operate.

• The advantage of inherent heterogeneity of peer-to-peer networks is exploited. In a purely decentralized network, all of the nodes will be equally (and usually heavily) loaded, regardless of their CPU power, bandwidth, or storage capabilities. In partially centralized systems, however, the supernodes will undertake a large portion of the entire network load, while most of the other (so called "normal") nodes will be very lightly loaded, in comparison[8].

Kazaa is a typical, widely used instance of a partially centralized system implementation (as it is a proprietary system, there is no detailed documentation on its structure and operation). Edutella is another partially centralized architecture. The research [Yang and Garcia-Molina] addresses the design and searching techniques for partially centralized peer-to-peer networks. The concept of super nodes has also been proposed in a more recent version of the Gnutella protocol. A mechanism for dynamically selecting super nodes organizes the Gnutella network into an interconnection of super peers and client nodes[9]. When a node with enough CPU power joins the network, it immediately becomes a super peer and establishes connections with other super peers, forming a flat unstructured network of super peers. If it establishes a minimum

required number of connections to client nodes within a specified time, it remains a super peer. Otherwise, it turns into a regular client node.

**Hybrid decentralized systems** - There is a central server facilitating the interaction between peers by maintaining directories of the shared files stored on the respective PCs of registered users to the network, in the form of meta-data. The end-to-end interaction is between two peer clients; however these central servers facilitate this interaction by performing the lookups and identifying the nodes of the network (i.e. the computers) where the files are located. The terms "peer-through-peer" or "broker mediated" are sometimes used for such systems. Obviously in these architectures there is a single point of failure (the central server)[10]. This makes them vulnerable to censorship, technical failure or malicious attack, which in itself is enough to defeats the purpose of p2p as we view it. Each client computer stores contents (files) shared with the rest of the network. All clients connect to a central directory server that maintains:

• A table of registered user connection information (IP address, connection bandwidth etc.)

• A table listing the files that each user holds and shares in the network, along with metadata descriptions of the files (e.g. filename, time of creation, etc.)

A computer that wishes to join the network contacts the central server and reports the files it maintains. Client computers send requests for files to the server. The server searches for matches in its index, returning a list of users that hold the matching file[11]. The user then opens direct connections with one or more of

the peers that hold the requested file, and downloads it (see figure FF). The advantage of hybrid decentralized systems is that they are simple to implement, and they locate files quickly and efficiently. Their main disadvantage is that they are vulnerable to censorship, legal, action, surveillance, malicious attack, and technical failure. The reason that the shared content or at least descriptions of it, and the ability to access content is controlled by the single institution, company, or user maintaining the central server. Furthermore, these systems are considered inherently un-scalable, as there are bound to be limitations to the size of the server database and its capacity to respond to queries. Large Web search engines have, however, repeatedly provided counterexamples to this notion. Examples of hybrid decentralized content distribution systems include the notorious Napster and Public us systems that rely on a static, system-wide list of servers. Their architecture does not provide any smooth, decentralized support for adding a new server, or removing dead or malicious servers. It should be noted that systems that do not fall under the hybrid decentralized category may still use some central administration server to a limited extent, for example, for initial system bootstrapping or for allowing new users to join the network by providing them with access to a list of current users (e.g. gnutellahosts.com for the gnutella network).

## 4. P2P APPLICATIONS

The Peer to Peer applications can be classified into four types-

**1. File sharing-**Content storage and exchange of the areas where Peer to Peerequipment has been most successful. File sharing applications [9], [11], [11] focus on storing information on and retrieving information from various peers in the networks. The popular example of Peer to Peer system is Napster, it became famous as a music exchange system. Other instances are Gnutella, Freenet, Kazaa, Chord, etc.[13].

**2. Collaboration-**Collaborative Peer to Peer applications aim to allow application level collaboration between users. These applications range from immediate messaging and chat, to on line games, to shared applications that can be used in business, educational, and home environments. Such as Groove, Jabber. [12] is a traditional of streaming XML(Extensible Mark-up Language) protocol and technology that enable to the entities of Internet to the exchange messages, presences, and other structure information in to the close real time. Groove [14] provides a variety of applications for communication, content sharing (files, images and contact data), and collaboration (i.e. group calendaring, collaborative editing and drawing, and collaborative Web browsing).[13]

**3. Distributed computing-**These applications use resources from the number of networked computers. The general knowledge behind these applications is that idle cycles from any computer connect to the network can be used for solving the problems of the other computers that require extra computation. SETI in home is one example of the such systems. SETI (Search for Extraterrestrial Intelligence) [16] is a scientific search project aimed at building a huge virtual computer based on the aggregation of the computer power offered from internet connected computers during their idle periods. The project uses two major components: the database server and the client. Clients can help with search for excess terrestrial life by running the search program for a specified portion of the universe. This project strongly relies on its server to distribute jobs to each participating peer and to collect results after processing is done.[14]

**4. Platforms-**P2P platforms provide infrastructure to support distributed applications using p2p mechanisms. P2P components used in this context are for instance naming, communication, discovery, resource aggregation andsecurity. JXTA [20] is p2p platform that provides a general-

purpose of the network is being programming and distributed computing infrastructure. It creates a Peer to Peer system by identifying a small set of basic functions necessary to support p2p applications and providing them as building blocks for higher-level functions. it includes three layer: core, services and applications. JXTA core provides core support for peer-to-peer services and applications [15]. At the core, capabilities must exist to create and delete peer groups, to advertise them to potential members, to enable others to find them, and tojoin and leave them. At the next layer, the core capabilities can be used to create a established of peer services, including indexing, searching, and file sharing. In the third layer peer applications can be built using these facilities [16] [17].

## 5. PEER TO PEER (P2P) CHALLENGES

P2P system is an offer the many number of advantages over conventional client-server systems such as fault Tolerance, scalability, performance. However, there are some challenges are:-

**Security:**-Distributed implementations create additional challenges for security compared to client-server architecture. Since in P2P systems the set of active peers is dynamic and also peers don't trust each other, achievement a high level of security in peer-to-peer systems is more difficult than non-peer-to-peer systems [18]. Traditional security mechanisms to protect data and systems from intruders and attacks such as firewall can't protect peer-to-peer systems since they are essentially globally distributed and also these mechanisms can inhibit peer-to-peer communication. Therefore new security concepts are required that allows interaction and distributed processing in peer-to- peer systems. [19] [20].

**Reliability** A reliable system is a system that can be recovered when a failure occurs. The factors should be occupied into account for reliability are data replication, node failure detection and recovery, existence of multiple guarantees for location information to avoid a single POF(point of failure) and the availability of multiple paths to data. Data replication increases reliability by increasing redundancy and locality. These are two strategies for replication, owner replication and path replication. In owner replication, when the search is successfully of the data stored on the client node only. In path replication, when the search succeeds, data is stored in all nodes beside the route from requester node to provider node [21]. P2P communities can also replace and replicate the data to achieve adequate performance [22]. In structured P2P overlay networks the messages is routed in minimum number of nodes. The overlays should modified routing states are automatically when nodes are join and leave It should route messages are correctly even a huge segment of nodes the network partitions or crash. To achieve reliability in such systems, nodes essential consume networks bandwidth to maintain routing state, so to reduce this cost the techniques should be employed that adapt to operating condition [23]. For increasing fault-tolerance and reliability in unstructured Peer to Peer systems, dynamically adding terminated links to the systems have been addressed [24] [25].

**Flexibility:**-Flexibility is the important aspects in Peer to Peer system are the autonomy of peers so that they can join or leave at their will. Recent P2P(Peer to Peer) systems can be distinguished by their decentralized control, extreme and large dynamism in the network. To deal with the scale and dynamism the properties of adaptation and self-organization are required to be considered in building p2p systems. More recent unstructured Peer to Peer systems, like KaZaA and GIA [26] address the dynamic environment. Queries In Kazaa are send only to super nodes, which maintain a list with the file names of their connected peers, avoiding overloading all peers of the system. GIA is a Gnutella like system which aims to respond to highly aggregated query rates. In GIA each peer calculates the maximum

number of queries it can handle per second and based on the metric to number of neighbors to which the peer can connect or forward a request is computed [27].In standard structured Peer to Peer systems, Peers are assigned static identifiers and distributed data structures are constructed based on these identifiers, so the overlay network structures are determined through the choice of these identifiers and in turn any self-organization of the systems are prevented. Structured systems based on DHTs should perform lookups quickly and consistently while nodes arrive and depart from the system [28], [29]. For instance Chord adapts as nodes join or leave system, and respond answer of queries although the system is changing continuously. Self-stabilization protocol run by every node periodically is used to discover joined nodes . Complex Adaptive Systems which are used to describe social systems and certain biological behavior can be used as a model to build adaptive P2P networks.

**Load Balance** Data distribution to be warehoused or computations to be carried out by the nodes are critical issues for the efficient operation of P2P networks. A particular method for such distribution in P2P systems are the DHT (distributed hash table), in which each data item that is stored is mapped to unique identifier ID. The identifier space is divided among nodes and nodes have the responsibility of storing the data mapped to identifiers in its portion of the space. In such approaches load balancing should be considered in both address-space balancing key address-space distribution among nodes and item balancing in the case that distribution of data in address-space can't be randomized. In this method, nodes are free to migrate anywhere and it has no restriction to be in a certain number of virtual node locations (it means the items can migrate among the nodes) [30], [31], [32]. Load balancing among the computing nodes in Peer to Peer systems can be implemented by agent-based self organization models. Messor is a Anthill load balancing algorithm. In Messor, their behavior is adopted by ants on the load conditions, wandering about randomly when the loads are uniformly balanced, moving rapidly to regions of network with high unbalanced loads. There are high tendency of failures if jobs are assigned to crashed nodes are simply reinserted in network by the nest that generated them and they are self-organized as new nests or nodes may join to a system and the computing power is rapidly exploited to carry on the computation, as soon as ants discover the nest and start to assign it jobs transferred from other nests.

## CONCLUSION

The restrictions of client/server frameworks get to be distinctly obvious in huge scale disseminated situations. P2P systems can be utilized for enhancing correspondence handle, streamlining assets revelation/ confinement, encouraging appropriated data trade. Peer-to-Peer applications need to find and find productively the hub that gives the asked for and focused on administration. Numerous P2P designs have been proposed in the writing, these models don't work together, and after that P2P stages have been showed up. It introduces a summed up and finish study on P2P exercises. Essential elements that ought to be tended to on P2P system are execution, versatility, upkeep, dependability, ease of use, naming, organizing, directing and finding, asset over seeing, topology upgrading.

## REFERENCES

[1]. S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer file sharing technologies, 2002.

[2]. Beverly Yang Hector Garcia-Molina, Improving Search in Peer-to-Peer Networks, Computer Science Department, Stanford University.

[3]. BitTorrent, 2005, http://www.bittorrent.com/.

[4]. Chang B J, Kuo S L. Markov chain trust model for trust value analysis and key

management in distributed multicast MANETs. IEEE Transactions on Vehicular Technology, 2009, 58(4) pp. 1846-1863.
[5]. I.Clarke, O. Sandberg, B.Wiley and T.W. Hong, Freenet: Adistributed anonymous information storage and retrieval system,Proc. of ICSI Workshop on Design Issues in Anonymity andUnobservability, 2000.
[6]. Dan S.Wallach. A survey of peer-to-peer security issues. In ISSS, pages 42–57, 2002.
[7]. Imad Jawhar and Jie Wu, "A Two-Level Random Walk Search Protocol for Peer-to-Peer Networks", Department of Computer Science and Engineering, Florida Atlantic University.
[8]. Jingyu Feng. The research of trust management techniques for open P2P network environment. Xidian University, 2011.
[9]. Jelasity, Márk. "PeerSim: A peer-to-peer simulator." http://peersim.sourceforge.net. 2009.
[10]. John Risson and Tim Moors, Survey of research towardsrobust peer-to-peer networks: Search methods, journal ofComputer Networks, Elsevier, vol. 50, pp. 3485-3521, 2006.
[11]. Jing Wang, Shoubao Yang, Ying Gao and Leitao Guo, FCAN: A Structured P2P System Based on Content Query, in Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), 2006.
[12].http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html.
[13]. JXTA v2.0 Protocols Specification.
[14]. KaZaA, 2001, www.kazaa.com. 2001.
[15]. Liu, Kai, et al., 2014, "A Participation-Based Trust Model for Mobile P2P Networks." Journal of Networks 9.7: 1738-1746.
[16]. Lo, V., Zhou, D., Liu, Y., Gauthier Dickey, C., Li, J.: Scalable Supernode Selection in Peerto- Peer Overlay Networks. Second International Workshop on Hot Topics in Peer-to-Peer Systems, La Jolla, California (2005)
[17]. Leonardo Mariani. Fault-tolerant routing for p2p systems with unstructured topolog. Inproceedings of the 2005 International Symposium on Applications and the Internet (SAINT 2005), IEEE Computer Society, February 2005.
[18]. Michael J. Freedman and Robert Morris, Tarzan: A Peer-to-Peer Anonymizing Network Layer, CCS02, Washington, DC,USA, November 2002.
[19]. Mourad Amad and Ahmed Meddahi, P4L: A four layer P2Pmodel for optimizing resources discovery and localization,APNOMS 2006, LNCS 4238, pp. 342-351.
[20]. IH. Witten, A.Moffat, and TC. Bell, Managing Gigabytes:Compressing and Indexing Documents and Images, MorganKauffman, second edition, 1999.
[21]. Napster, http://www.napster.com
[22]. B. Pourebrahimi K. Bertels S. Vassiliadis Computer Engineering Laboratory, ITS, TU Delft, The Netherland sfbehnaz, koen, stamatisg@ce.et.tudelft.nl
Project JXTA, http://www.jxta.org.
[23]. CG. Plaxton, R.Rajaraman and AH. Richa, Accessingnearby copies of replicated objects in a distributed environment,In Proceedings of ACM SPAA, ACM, June 1997.
[24]. Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In ICS '02: Proceedings of the 16th international conference on Supercomputing, pages 84–95. ACM Press, 2002.
[25]. Qureshi B, Min G, Kouvatsos D. M-Trust: A trust management scheme for mobile P2P networks. IEEE/IFIP 8th International Conference on Embedded and Ubiquitous, Hong Kong, China. 2010 pp. 476-483.
[26]. RatulMahajan, Miguel Castro, and Antony Rowstron. Controlling the cost of reliability in peer-to-peer overlays.InIPTPS'03, February 2003.
[27]. Ra Ti On. Project jxta: An open, innovative collaboration.
[28]. Q.Lv, S.Ratnasamy, and S.Shenker. Can heterogeneity make gnutella scalable?, In Proceedings of the 1st InternationalWorkshop

on Peer-to-Peer Systems (IPTPS '02), MIT FacultyClub, Cambridge, MA, USA, March 2002.
[29]. Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble.A measurement study of peer-to-peer file sharing systems. In Proceedings of Multimedia Computing and Networking 2002(MMCN '02), San Jose, CA, USA, January 2002
[30]. S. Seuken and D. C. Parkes. Sybil-proof accounting mechanisms with transitive trust. In Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Paris, France, 2014.
[31]. SETI@home, 2001, setiathome.ssl.berkeley.edu.
[32]. H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A survey of multi-agent trust management systems. IEEE Access, pages 35–50, 2013.
[33]. J. Tang, S. Seuken, and D. C. Parkes. Hybrid transitive trust mechanisms. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, pages 233–240, 2010.
[34]. Tsoumakos and N. Roussopoulos. "Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks". Technical Report CS-TR-4451, Un. of Maryland, 2003.
[35]. Ulrike Lechner. Peer-to-peer beyond file sharing. In IICS, pages 229–249, 2002.
[36]. V. Vishnumurthy and P. Francis." A comparison of structured and unstructured P2P approaches to heterogeneous random peer selection". In Proc. Usenix Annual Technical Conference, 2007.
[37]. Vladimir Vishnevsky, Alexander Safonov and MikhailYakimov, Scalable Blind Search and Broadcasting in Peer-to-Peer Networks, In Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06), 2006.