Peer – To – Peer Computing: Architectures, Applications and Challenges

D Giridhar Kumar, Ch D V SubbaRao

ABSTRACT--- In Peer – to – Peer (P2P) computing technology, several individual systems are grouped together and are characterized by direct access between peer systems for exchanging services without any centralized server. Unlike traditional client-server model, in Peer – to – Peer each peer can act as client as well as server based on its requirements and each peer has its own capacity & responsibility (as client it can raise requests for services and as server it can serve requests coming from other peers). With the availability of internet, distinct systems at distinct locations are being connected easily which enhanced the usage of P2P applications. In this paper, we present the overview of various P2P architectures, characteristics of P2P in addition to its applications and challenges.

Index Terms— Peer to Peer Computing, P2P Architectures, Structured P2P & Unstructured P2P systems, virtual overlay

I. INTRODUCTION

Peer – to – Peer Computing [1] model is based on how we (human) communicate in real world. If we need something then we communicate directly to other corresponding peers (may be friends) who may in turn refer us to their corresponding peers for working towards completion of the request. Thus there is a direct access between the peers without any third party intervention. According to C.Shirky, “P2P is a class of applications that takes advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet.”

Peer-to-peer (P2P) systems are getting more popular over the internet based applications. Such kind of networks allows individual peers to collaborate and perform different types of services in a distributed manner over the network. P2Pnetworks are characterized in such a way that there are no central monitoring servers for communication among peers.

A. Evolution of P2P from Client-Server Model

Many of the internet applications are using Client-Server model (Fig. 1) Example: WWW, email etc. In such model there will be a centralized server shared by many clients. The clients query the request to server and get services [2]. It will be facilitated if the server is available and capable of serving all the requests from distinct clients at a particular moment.

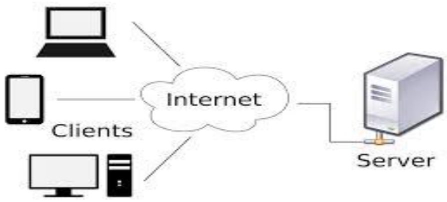

Fig. 1 Client-Server Model

There may be a chance of getting performance issues because of the unavailability of resources (e.g.: memory, bandwidth, processing speed) at the server system when too many requests arise. In order to overcome the drawback of previous model, Grid has came in to existence. Grid Computing refers to the implementation of client server architecture for distributed computing. The aim of grid computing is to provide high performance computing among the individual peers of the network. The peers involved in the network are of having high resources, so it’s a costlier model.

Unlike Client-Server model, P2P networks are decentralized distributed systems where participating peers can share and integrate their computing resources. Although P2P and Grid look similar they address to different domains. In Grid model higher end resources are involved where as in P2P idle resources are involved at the edge level. In addition to that as a contrast to Grid, P2P doesn’t rely on centralized server for services.P2P network is a logical overlay network over the physical underlying infrastructure as shown in Fig.2.

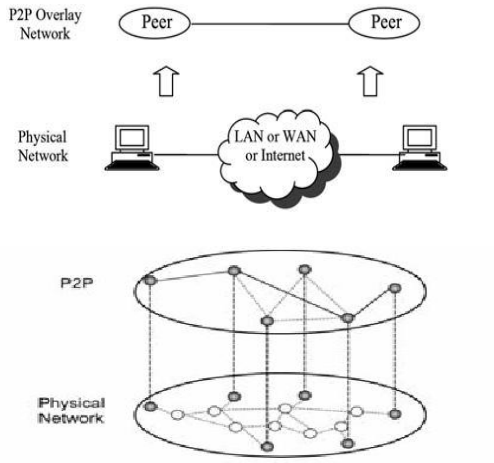

Fig. 2 Peer – to – Peer Models

Revised Manuscript Received on June 10, 2019.

D Giridhar Kumar, Research Scholar, Dept. of CSE, SVUCE, Tirupati, A.P, India.. (E-mail: giridhar518@gmail.com)

Dr. Ch D V SubbaRao, Professor, Dept. of CSE, SVUCE, Tirupati.A.P, India.(E-mail: subbarao_chdv@hotmail.com)

B. Advantages of P2P Systems:

- As P2P does not rely on central servers, the end users can easily share & retrieve resources from other connected peers directly.
- P2P Systems are more resilient to single-point-of-failure than client-server model as there is no central server. Failure of a single peer doesn't affect the network.
- The responsibilities of the central server are distributed among each individual peer of the network. Every user is a administrator of his machine and have control on his shared resources.
- P2P network is a logical overlay for the underlying physical systems, so the communication in overlay network is through virtual channels which avoid the obstacles from firewalls and Network Address Translation (NAT).

The rest of the paper is organized as follows: Section-2 gives the basic characteristics of P2P networks. Section-3 covers the different types of P2P architectures and also differentiates Structured P2P and Unstructured P2P. Section-4 includes the basic application areas where P2P technologies are deployed. Section-5 deals with the challenges and issues that arise with the usage of P2P. Finally we draw a conclusion in section-6.

II. CHARACTERISTICS

P2P network is one form of distributed computing networks where independent machines are grouped to appear as a single coherent system. So P2P shares common characteristics with distributed systems, in addition some special characteristics that make P2P distinct from other networks are as follows:

Nature of node: The nodes are symmetric in nature that means every node in the network can act as client (raising requests) as well as server (serving queries).

Scalability: There is no restriction on number of participating peers where asin traditional networks the number of participating nodes depends on the capacity of server [3].

Heterogeneity: The participating machines are not necessarily homogenous. A P2P network may have a very slow machine and a high end super computer working together.

Attacks: The heterogeneous peers make virus and worms harder in the network. Thus P2P is resilient to attacks.

Dynamism: In P2P applications we find dynamic change of topology due to joiningof new nodesor leaving of existing nodes from the network.

Self Organization: The nodes of the network reconfigure according to the dynamic changes in the topology due to a node joining or leaving [3].

Fairness: Each participating machine should contribute resources to the network based on its capacity [22].

Huge Resources: In P2P, we will have large collection of resources due to voluntary participation of millions of simultaneous users from all over the world.

Flexibility: As there is no central controlling system, each participating peer is completely flexible making the overall system unreliable.

Performance: To avoid single-point-of-failure, data and object references are replicated at distinct peers. This also balances access load and enhances search & retrieval of data.

III. TAXONOMY OF P2P SYSTEMS (ARCHITECTURES)

Based on the degree of network centralization, the P2P networks have been categorized in to three broad categories: Centralized P2P, Decentralized P2P, and Hybrid P2P. This type of classification is mainly visible in the case of file sharing P2P applications [1][3].

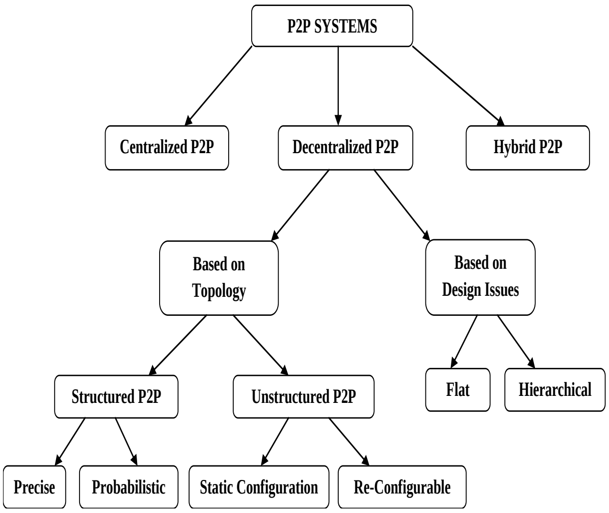

Fig. 3 Taxonomy of P2P Systems

A. Centralized P2P

This is the first generation of P2P, where there will be one or more central servers. Unlike traditional client-server model, here the servers contain only the Meta information about the shared resources (e.g.: only node ID or address where the shared content is available) instead of maintaining/storing the actual resources. The peer in the network has to initially raise query request for resources to the central server, the server which is having the meta information replies with the list of peer ID's who can provide services to the request.

A direct access link is established between the requester peer and servicing peers as shown in Fig. 4. In the figure, initially Peer A requests server for resources (e.g.: data). The central server finds Peer B and Peer C can individually serve the request of Peer A and sends the same node ID's to Peer A. Now Peer A establishes direct communication link with Peer B, Peer C and gets its request serviced.

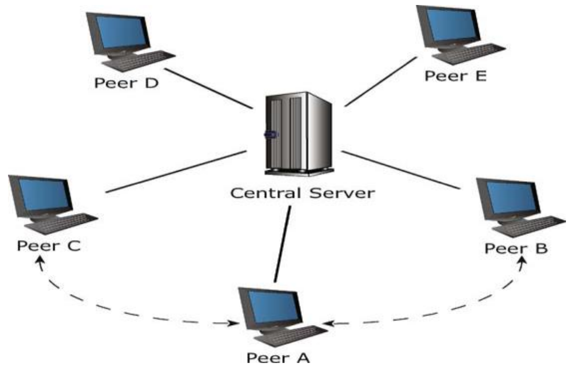

Fig. 4 Centralised P2P System

Centralised P2P architecture speeds up the resource identification and guarantees the availability of nodes with desired resources. But as of traditional Client server, scalability and single point of failure are two main challenges as it rely on central servers. Examples for Centralised P2P systems in real world usage are Napster (file sharing application), SETI@home etc.

B. Decentralised P2P

Decentralised P2P means there will be no central server, and all the peers in the network are offered with equal freedom and responsibilities. As there is no central server to guide, finding the service providing peer for any query is a challenging task.

Based on the design issues (Network Structure) decentralised P2P are classified into two types: Flat & Hierarchical. Flat Structure consists of Single-tier with uniform distribution of load and functionality among all the peers in the network. Coming to hierarchical, it's a multi-tier structure with multiple layers of routing and thus providing the benefits of fault isolation, security, effective caching, bandwidth utilization, hierarchical storage etc. Some of the recent P2P networks are using the advantages of both flat and hierarchical structures by converting flat to hierarchical model, Example: Canon network.

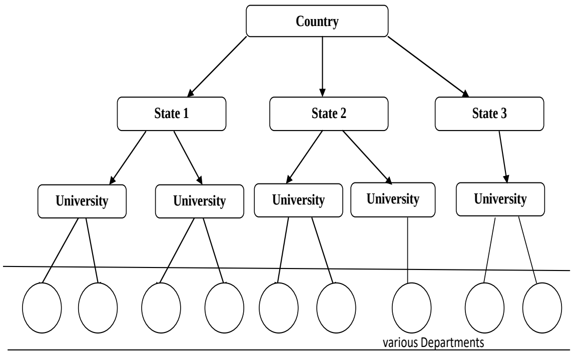

Fig. 5 A Sample Hierarchical Structure

Fig 5 shows a sample hierarchical structure with multiple layers. Nation as the top level, followed by distinct states which include different universities and various departments in the next subsequent hierarchies.

Based on the Network Topology & query routing, decentralized P2P are classified as Structured P2P and Unstructured P2P. The Virtual Overlay network is considered as the topology but not the underlying physical network. A virtual overlay network [7] is a form of network virtualization which provides path between application software and underlying physical machines. Formulation of Virtual Overlay provides many advantages like Traffic Isolation, Scalability, and Network Independence.

B.1.Unstructured P2P

Differentiating factor between unstructured and structured P2P is query routing (the way in which queries are being forwarded). In Unstructured P2P, the peers should have the knowledge about its neighboring peers to which query should be forwarded. The neighboring nodes/peers in the network can be Static or Re-Configurable. In the former, the nodes are predetermined and are fixed in the network. In the later, the neighboring nodes are reconfigurable based on the peer interests. The peer may issue similar kind of requests during a period of time, so the nodes which answered the previous requests are likely to be neighboring nodes to reduce the query time. If the peer interests changes then the query requests will change which in turn need the change of neighbors to reduce query latency.

The basic routing strategies that are followed in unstructured P2P for searching, data sharing are unicast and broadcast. In unicast based searching the query request is forwarded to one of the neighboring nodes and if we get query hit, the search process get terminates else the query is forwarded to other node and the process continues until all nodes are visited or if we get query hit whichever is earlier. This approach consumes less bandwidth but requires more time. Example: Freenet [4] follows unicast based routing strategy for its search and data sharing operations.

In broadcasting approach, the search query request is forwarded to all the neighbors who forward the same to its neighbors, thus broadcasting the request in the network. This approach creates flooding of the request in the network which increases traffic. This approach is efficient in terms of response time but uses high bandwidth and exponential number of messages. Example: Gnutella is an Unstructured P2P network using flooding based routing strategy. In Gnutella network new node joins by forwarding ping message to the existing nodes which responds through pong message. The joining node establishes some among the replied nodes as its neighbors. Fig 6 shows a sample Gnutella network how searching takes place through flooding. Suppose Peer A requires data which is available with Peer D, therefore Peer A starts broadcasting its request to its neighbors and gradually to others peers in the whole network.

In Unstructured P2P [5], it is difficult to predict which peer is having data (in case of search requests) and there will be no guarantee for completeness of answers (as peers having the data may or may not be available in the network at the particular moment). The response time may also vary for the same request at distinct times.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

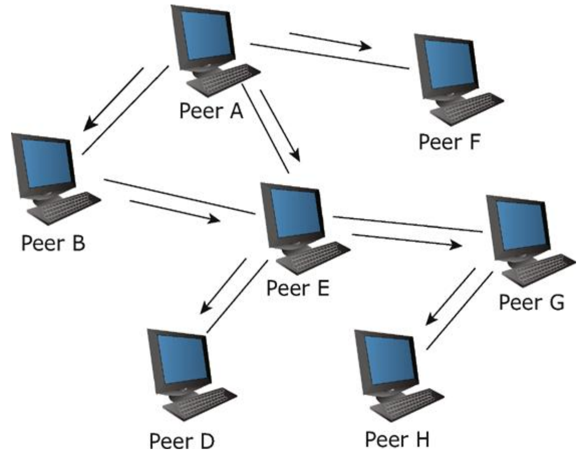

Fig. 6 searching in Gnutella

B.2.Structured P2P

In Structured P2P system, there will be certain mechanism to precisely determine the file location in the network. This can be facilitated by applying Distributed Hash function (e.g., SHA-1) on both files and peers. The files are placed in the Distributed Hash Tables (DHT) whose hash value is closer to the peer Id, thus providing a link between files and peers. Thus the query routing is directly to the peer having the file through hash values, so unnecessary routing to other peers is avoided. Example: PAST, Chord, CAN are few examples of structured P2P networks. Among those PAST is a structured P2P file sharing system which uses prefix based routing strategy. In routing, the next node in the path is precisely identified based on the maximum prefix match of its neighboring PeerID to the destination PeerID.

Unlike unstructured P2P, desired objects can be identified precisely and efficiently but it requires extra storage space for storing routing tables that contain routing information. The difficulty here is to maintain the routing tables when a node joins or leaves the network at higher rates.

Regardless of Structured P2P or Unstructured P2P, in common a fully decentralised P2P system avoid single point of failure as there is no central server and has the advantages like high performance rates, scalability, robustness, availability, self organization of nodes fault resilience etc.

C. Hybrid P2P

Hybrid P2P System [6] is a combination of both Centralised P2P and Decentralised P2P and thus provides reliable resource locating as of Centralised P2P and has wider scalability as of Decentralised P2P. This is facilitated with the help of Super Peers. Super Peers are similar to central servers in Centralised P2P but not exactly same. It is a node which is having more capabilities than others and is made as incharge of subset of peers. These super peers form the higher layer of the network and provide the services as of central servers in locating resources within its subset of peers.

Example: BestPeer (A self configurable P2P system), PeerDB, CQBuddy are examples for hybrid P2P system. In BestPeer system, just like super peers it maintains Location independent global names lookup servers (LIGLOs) in the upper layer.

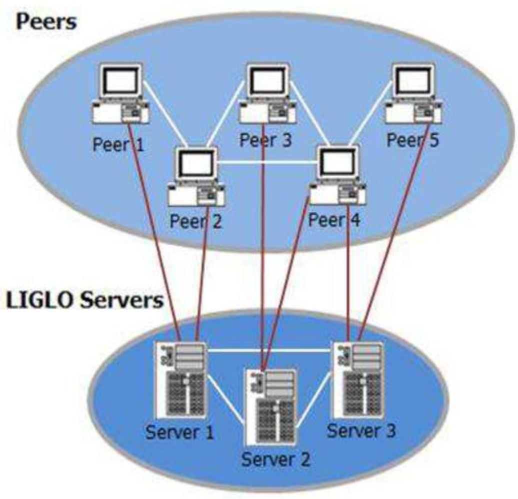

Fig. 7 BestPeer Network

The underlying peers communicate with the corresponding LIGLOs for the queries. A new node can join to any number of LIGLOs and chose its neighbors. Hybrid P2P provides the advantages of optimized network topology, improved response time and also avoids single point of failure.

IV. APPLICATIONS OF P2P SYSTEMS

There are several applications that have employed P2P technologies, now we will have a look at few areas where P2P technologies are deployed.

A. Data Sharing

By using Client-Server model, the data sharing is asymmetric. This means, the data is available only at the server systems and the clients have to request server for the data which then shares the data. This method creates single point of failure, limited servers for huge numbers of clients and for vast data repositories.

Applying P2P technology for file/data sharing applications [6][7] has overcome the above limitations. Here no separation of servers and clients, all the participating nodes are symmetric (can act as client or server) and the data is shared directly among the fellow nodes. The P2P platform for searching can be in fully decentralised manner (Example: Gnutella, Refer Fig 6) or by using centralized mechanism (Example: Napster, Refer Fig 4).

B. Storage

P2P provides backup facility for the data [8], but instead of maintaining single server this technology uses unused storage spaces at distinct users. Designing and maintaining such a P2P backup system is challenging task as P2P is dynamic in nature making peers to join and leave at any time. If we simply store without backup then there may be chances of storage attacks which includes i) A node may not serve the data request which it is responsible for ii) A node may masquerade as a different peer iii) A node may delete data which creates permanent loss of data.

Duplication of data is stored at multiple peers is followed to avoid few problems and to make the data available nearer (within less time). Another solution to the above is PIPE (Peer to Peer Information Preservation and Exchange network) which considers malicious nodes and node failure rates for storage and retrieval of data. PAST and OceanStore [9] are examples of P2P storage systems (they just store data without backup semantics). The P2P systems that store data along with backup facility are pStore [10] and Pastiche.

C. Collaboration and Communication

In today's work environment, as example if we see software companies, resource persons working for same project are at distinct locations but they have to collaborate to work together [11]. P2P provide cooperative collaboration environments where a virtual space is created and given for team members to enable interactive working. Groove and JBuilder are examples for P2P collaborative tools.

With respect to communication, P2P is providing instant messaging facility and serverless communications. The widely used Skype is a P2P Application of communication which enables users to communicate, exchange files and also provides information about availability of peers.

D. Search Engines

P2P technologies are applied to develop distributed search engines [3], where there is no central server. In contrast to traditional centralized search engines, here the works such as crawling, data mining, indexing, and query processing are distributed among several peers in a decentralized manner.

Examples: Opencola, YaCy [12], FAROO [24], and QueenBee (A decentralized search engine project launched in August 2018 aiming to build fully decentralized search service on top of decentralized web).

E. Medical and Scientific

In the field of medical sciences, patients and doctors collaborate to improve medicine. Data collected on individual patients can eventually combine to construct an extensive database of facts that would act as an ongoing clinical trial and could bring important new insights to medicine as a whole. Example: Sciencenet, it's a P2P based search engine used for searching large data space (space used for biological experimental results) distributed geographically.

Super computers are required for scientific research computations. But instead of using super computers (which are costlier) P2P technology enables usage of large number of idle computers to complete the same task. Example: SETI@home (Search for Extraterrestrial Intelligence at home). In this entire task is split into manageable work units and are assigned to idle computers distributed at distinct locations.

F. Distributed DBMS

P2P supports high level complex query processing instead of just keyword based search mechanisms. The input can be high dimensional data or multiple tables. Those data has to be managed which are distributed in the form of local databases. Sometimes those local databases itself have to be shared (enhancement to data sharing). It is difficult and costlier to maintain large databases by a single entity so they are distributed to different organisations for its storage and maintenance but gets shared when needed.

Example: Patient information is maintained in distinct databases by the hospital and similarly by the clinical laboratory. Through P2PDB sharing, the doctor is shared with databases from both the hospital and the laboratory which makes him to give appropriate treatment by correlating the data.

V. DESIGN ISSUES AND CHALLENGES OF P2P NETWORKS

Even though P2P technologies provide wider range of benefits through its applications (mentioned in section 4 of this paper), it faces certain challenges and issues which have to be properly monitored. Some of them are as followed

A. Resource Discovery

Locating the resources as per the user requirements is becoming more and more important with the emergence of huge amounts of information and growth size of P2P networks. Especially in Unstructured P2P networks, due to the feature of dynamic and loosely coupling relations among peer nodes, the data resources resided in peers are usually independent of logical topology and thus making the resource discovery strategies hard to work.

There are many methods for finding the resources in unstructured P2P, some of them are flooding, Random Walk algorithm, Gossip based Search mechanism [13], and Query Routing Trees (QRT) [14]etc...

B. Availability of Data/Resource

A node can join or leave the P2P network at any instant of time based on its interests. As the nodes are autonomous it is difficult to predict whether the required data/ resources are available at a particular moment. At some situations the data may available but not complete. This is because in P2P the data is divided and shared among distinct peers whose availability is not guaranteed. One of the basic solutions for such problem is Replication [7] (Duplication of data at multiple peers).

C. Data Consistency

Continuously relying on Replication for availability of data and for reducing query processing time not only increases storage cost but also creates data consistency problem [15][16]. If any change is done in the data, then it has to be reflected in all its replicas (replicas should get updated), without which the outdated copies of data may be served to the user requests. So it is the basic task of the network to maintain consistency among all its replicas. It can be done by removing outdated copies or by applying few methods to update replicas immediately when change occurs at original/owner node (it's a costlier process).

As an alternative approach, the nodes having replicas should check with the node having original data at regular time intervals for any updations. If any updations are there, then it will be indicated with version numbers and thus consistency has to be maintained.

D. Load Balancing

Load Balancing means distribution of workload on multiple peers in the network to achieve optimal resource utilization for maximizing throughput and minimizing overall response time. If in the network nodes are homogeneous then the workload is equally distributed, but P2P allows heterogeneous nodes so in that case the distribution of load should be based on the capacity of the particular node participating in the network. As nodes join and leave the network at high rate the load has to be balanced dynamically.

Thus load balancing in P2P forms a major challenging issue such that no node should be overloaded or underloaded with work. Under Static methods, the load has to be transferred from heavily loaded node to lightly loaded node and vice versa at the times of node joining and node leaving respectively.

E. Congestion

Congestion of traffic is mainly seen in P2P file sharing applications because of which packet (data is transferred as packets) loss may occur or may create a delay in process. Some of the factors that create congestion (Overflow of traffic) are attacks and through relying on same route for several requests. In order to control congestion in P2P, we should monitor on attackers who increases traffic by flooding and at the same time we should find optimal routes for the transfer of packets [17][18][19].

F. Security

Security is more complex in P2P systems than in client-server model, because of openness and autonomous nature of P2P. Unlike traditional client-server model, in P2P internal data is exposed to its fellow nodes so easy for the attackers to hack the data. A malicious node can easily enter the network and can create Denial of Service (DoS) [20][21]. The Commonly seen security issues [23] on Peer – to - Peer networks are Routing attacks, Retrieval attacks, Distributed DoS (DDoS) attacks, privacy & identity, poisoning the network etc.

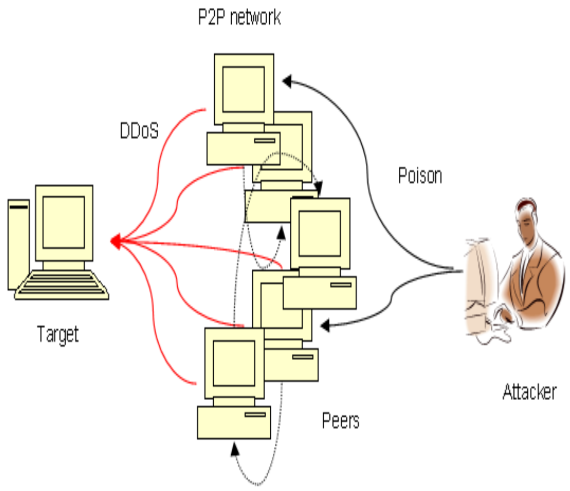

Fig 8 DDoS in P2P

Routing Attacks are mainly seen in Structured P2P where routing is through routing tables. Attackers will wrongly updates the tables or forward the query not as in routing table (incorrectly). DoS attacks can be seen at network layer by flooding traffic and at application layer by creating huge application requests.

VI. CONCLUSION

With the rapid increase of P2P enabled applications over the internet (mainly file sharing, communication) there is an increased focus on P2P Networks by the researchers. In this paper, we have drawn a overview about the various P2P system architectures and the challenging issues that have to be taken care while working with P2P systems. Further we can integrate P2P technologies over IoT Applications or we can incorporate Artificial Intelligence over the Cognitive P2P networks (Self Learning Peers).

REFERENCES

1. QuangHieu Vu ,MihaiLupu , Beng Chin Ooi, “Peer-to-Peer Computing Principles and Applications” , Springer Publication, ISBN 978-3-642-03513-5 e-ISBN 978-3-642-03514-2 DOI 10.1007/978-3-642-03514-2
2. HAND BOOK OF PEER-TO-PEER NETWORKING, Springer Publications ISBN 978-0-387-09750-3 , e-ISBN 978-0-387-09751-0 , DOI 10.1007/978-0-387-09751-0
3. B Lalitha, Ch D V SubbaRao, “Peer-to-Peer Systems: Taxonomy and Characteristics”, International Journal of Computer Science and Technology, ISSN: 0976-8491 (Online) | ISSN: 2229-4333 (Print), Vol. 3, issue 2, pp. 886-896, june-2012.
4. Freenet, http://freenet.sourceforge.net
5. XianfuMeng&Dongxu Liu, “A traffic-efficient message forwarding approach in unstructured P2P networks”, Springer publication, DOI 10.1007/s12083-015-0326-2.
6. ZhenhuaDuan, Cong Tian, Mengchu Zhou, XiaobingWang, Nan Zhang, Hongwei Du, · Lei Wang, “Two-layer hybrid peer-to-peer networks”, Springer Publication, DOI 10.1007/s12083-016-0460-5.
7. HaiyingShen,” An Efficient and Adaptive Decentralized File Replication Algorithm in P2P File Sharing Systems”,IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 6, JUNE 2010, doi: 10.1109/TPDS.2009.127.
8. Zhi Yang, Jing Tian, Ben Y. Zhao, Wei Chen, and Yafei Dai, “Protector: A Probabilistic Failure Detector for Cost-Effective Peer-to-Peer Storage”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 9, SEPTEMBER 2011, doi: 10.1109/TPDS.2010.205.
9. OcenStore, http://oceanstore.cs.berkeley.edu/
10. C.Batten, K.Barr, A.Saraf, S.Treptin, “pStore: asecure peer-to-peer backup system”, technical memo MIT-LCS-TM-632. December, 2001.
11. Michele Amoretti, · Alessandro Grazioli, · Francesco Zanichelli, “An autonomic approach for P2P/cloud collaborative Environments”, Springer publication, DOI 10.1007/s12083-015-0367-6.
12. https://yacy.net/en/index.html
13. HE Ming, ZHANG Yujie and MENG Xiangwu, “Gossip-Based Resource Location Strategy in Interest Community for P2P Networks”, Chineese Journal of Electronics, Vol.24, No.2, April-2015, doi:10.1049/cje.2015.04.008.

14. Gong Weihua, Jin rong, YANG Lianghuhai and HUANG Decai, ‘ Enhanced Entropy-BasedResource Searching in Unstructure P2P Networks”, Chineese Journal of Electronics, Vol.24, No.2, April 2015, doi: 10.1049/cje.2015.04.002.
15. YI HU, Laxmi N. Bhuyan, Min Feng, “Maintaining Data Consistency in Structured P2P Systems”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 11, NOVEMBER 2012, doi: 10.1109/TPDS.2012.81.
16. Haiying (Helen) Shen, “IRM: Integrated File Replication and Consistency Maintenance in P2P Systems”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 1, JANUARY 2010, doi: 10.1109/TPDS.2009.43.
17. Mritunjay Kumar Rai, Max Bhatia, “Identifying P2P traffic: A Survey”, Springer publication, published online: 18 june 2016, doi: 10.1007/s12083-016-0417-2.
18. Tao Qin, lei Wang, Dan Zhao, Min Zhu, “CUFTI:Methods for core users finding and traffic identification in P2P systems”, Springer Publication, published online: 11 april 2015, doi: 10.1007/s12803-015-0350-2.
19. Xiang-Jun Shen, Quing Chang, Lu Liu, John Pannerselvam, Zheng-Jun Zha, “CCLBR: Congestion Control-Based Load Balanced Routing in unstructured P2P systems”, IEEE Systems journal, 2016.
20. Hyun Mi Jung1 & Il-Sun Hwang1 &Jeong-Kyung Moon2 & Hark- Soo Park1, “A security monitoring method for malicious P2P event detection”, Springer Publication, DOI 10.1007/s12083-015-0369-4.
21. N. Daswani, H. Garcia-Molina, Query-flood DoS attacks in Gnutella, in Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), pp. 181–192, Washington, DC, 2002.
22. AlirezaNaghizadeh, BehroozRazeghi, EhsanMeamari, MajidHatamian, Reza EbrahimiAtani, “ C-trust: A trust management system to improve fairness on circular P2P networks”, Springer publication, DOI 10.1007/s12083-015-0402-7.
23. Security issues: https://www.cse.wustl.edu/~jain/cse571-07/ftp/p2p/
24. Faroohttps://mashable.com/2007/09/17/faroo/#MANxu_SaCuq4

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication