International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 10 3286 – 3290

# P2P Computing Concepts and Their Discussions

Er. Anup Lal Yadav
M-Tech Student

Er. Sahil Verma
Asst. Prof. in C.S.E. Deptt.
EMGOI , Badhauli.
*sahilkv4010@yahoo.co.in*

Er. Kavita
Asst. Prof. in C.S.E. Deptt.
EMGOI , Badhauli.

***Abstract-***Peer-to-peer is a technology concept applied at different levels of the systems architecture. Its main characteristics are direct interaction and data exchange between peer systems. It is the basis for decentralized distributed computing. The concept is widely deployed in different contexts and no formal definition exists. This paper gives an overview of the different areas peer-to-peer technology is used and introduces the main characteristics of peer-to-peer systems. It also discusses the issues and problems encountered when deploying peer-to-peer technology.

***Keywords:*** *Distributed systems, peer-to-peer, algorithms, performance design, grid computing, peer-to peer.*

*****

## I. INTRODUCTION

During years and today the client-server paradigm is the battle horse of the most users' applications. In the last years there is a new paradigm that is emerging, peer-to-peer (P2P) mainly supporting applications providing file-sharing, content exchange like music, movies and programs, but have also successfully implemented distributing computing and Internet-based Telephony. A refined definition of the Peer-to-Peer is : "A Peer-to-Peer [P2P] system is a self organizing system of equal, autonomous entities (peers)which aims for the shared usage of distributed resources in networked environment avoiding central services". It is possible to say that peer-to peer is a system with completely decentralized self organization and resource usage.

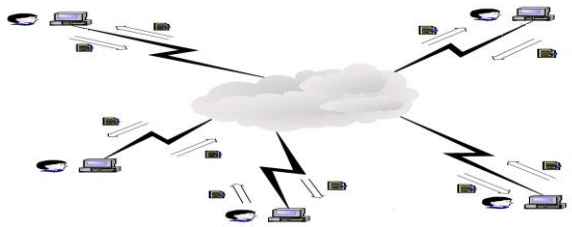

Figure 1 A picture of Peer to peer computing involving users and computers showing sharing or transfer of data.

Statistics establish that 50 per cent of Internet traffic obeys to peer-to-peer applications, in some cases up to 75 per cent. The growing of Internet, users and bandwidth, is requiring an increase of a diverse wealth of applications. The client-server paradigm requires a great effort and resources to meet these challenges. Internet-based applications identify three main characteristics:

- Scalability.
- Security and reliability.
- Flexibility and quality of services.

It is difficult for client-server based applications to meet the evolution of Internet. The client-server centralized approach is one of the main constraints (resource bottleneck), it is easily attacked and difficult to modify due its placement within the network infrastructure. All of the above expressed things indicate that there is a bias of paradigm, from client-server schemes to peer-to peer schemes.

## II. APPLICATION AREAS

The area where the concept of peer-to-peer is currently most heavily used is file and content sharing, collaboration support, distributed computing, communication and platforms.

### *A. Peer-to-Peer File and Content Sharing*

Among the systems that feature most prominent in the peer-to-peer domain are user applications running on top (or at the edge) of the Internet allowing a large group of users to interact and share resources. Most popular are file or content sharing applications such as Napster, Gnutella, Mojo Nation, eDonkey and Freenet. *Napster* was the first major system enabling the direct exchange and sharing of content. While the actual exchange of content in Napster is between peers, the discovery of the peers, however, is highly centralized (i.e. it is stored in a central directory) [2][4].*Gnutella* provides a purely distributed file sharing solution without a central node. In the strict sense Gnutella is not an application but a protocol used to search for and share files. To find content and other peers a user has to know the IP address of at least one other Gnutella node1. A node issues a query for a file by sending it to all other nodes known to it. If a node cannot serve a request it can forward it to other nodes. The query travels the Gnutella network until the file has been found or its time-to-live has been reached [1][4].

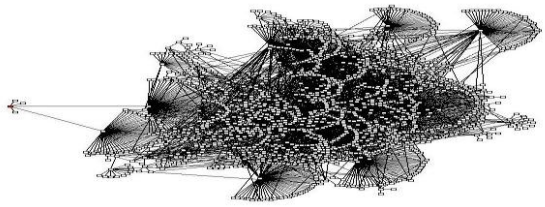

Figure 2 A Picture showing Gnutella network where the dots represents the nodes. *Mojo Nation* is a peer-to-peer content exchange application that introduces a virtual currency (so called Mojos) to counter free riders. This currency is also

used as incentive to contribute resources (such as storage space and content). The peers in Mojo Nation can have different roles, i.e. Block Server (provides storage), Content Tracker (content search services), Publishing Agent (content publishing service), and Relay Service (forwarding service). The content is split into blocks and distributed throughout the Mojo Network. Hence, a block server only hosts part of the content but not the entire file *Freenet* is also a file/ content sharing system. The primary goal of Freenet is to make its use completely anonymous. Neither the users requesting nor those placing files in Freenet can be identified. Further, an operator of a Freenet node is not able to determine what data is stored on its local disk. Freenet is completely decentralized and represents peer-to-peer in its purest form [1].

*B. Collaboration Support*

Peer-to-peer collaboration and communication support is provided by systems such as Centre span, Jabber, AIMster, Magi and Groove. These systems allow collaboration between users without the use of a messaging server. The simplest systems are only used for the exchange of messages. More sophisticated systems also allow joint authoring of documents, graphics, slides, etc. Games using peer-to-peer technology are also regarded as collaborative application since they also support interaction amongst users. *Groove* is a system that 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) [10]. In general, events and message exchange taking place in a peer-to-peer collaborative system are relayed instantly to all other members of the peer-to-peer group. Issues to be considered are fault tolerance and real-time constraints. The former is linked to reliable group communication whereas the latter refers to interaction constrains. Group and multipeer communications research have actually addressed these problems in the mid 1990s already [11]. Collaborative peer-to-peer systems share the problem of locating and addressing peers (respectively the peer-to-peer group) with all other peer-to-peer application areas. In contrast to the others, however, communication here is mostly synchronous amongst an identifiable group (although specific users might remain anonymous). Hence the addressing problem is related to addressing issues in group and multicast communication.

*C. Distributed Computing*

*A distributed system has been defined as a computer system in which several interconnected computers share the computing tasks assigned to the system as a single entity* [12]. Such systems are for instance considered clusters or the GRID. A prerequisite for distribute computing is that tasks can be split into sub-tasks, which can be processed independently from each other. Interaction between peer systems processing sub-tasks should be kept to a minimum. Suitable processes are Single-Process-Multiple-Data and multiprogramming problems. One of the most popular distributed computing applications is *SETI@home*, a project that aims to find prove of extraterrestrial intelligence [13]. Computers connected to the SETI@HOME system process data collected from a radio telescope. Their task is to find and identify any possible signals from intelligent populations outside the solar system. A database server controls the operations; the peers operate effectively as clients that get their jobs from this server. Hence peers in the context of this project refer to computing systems that make their resources available to others but that are still centrally controlled.

*D. Platforms*

There is an increasing trend away from native operating systems towards middleware platforms as application hosting environments (such as Java Virtual Machine or Web browsers). These platforms provide a largely operating system independent development environment and also offer high-level functions to system developers. Some of these platforms provide peer-to-peer support and use peer to- peer mechanisms. Peer-to-peer components used in this context are for instance naming, discovery, communication, security and resource aggregation. The two most comprehensive platforms supporting peer-to-peer at present are Sun's JXTA and Microsoft's .NET .

*E. Communication*

Peer-to-peer communication issues arise in various contexts. Communication is for instance an inherent aspect of every distributed application. The challenge here is to deal with the problems associated with the dynamic nature of the peering entities. Within communication itself peer-to-peer technology is a relatively old concept. Network topologies where the nodes are connected directly (i.e. meshed topologies and one-to-one connections) have always been part of communication infrastructures. *Flooding* or *network* broadcast are also technologies that support the communication in an anonymous peer group. Network multicast is another peer-to-peer communication mechanism where the user group is known. Group communication and multipeer communication models apply peer-to-peer principles at the transport layer [11]. In order to provide communication support to peer-to-peer applications one issue is addressing and routing. This should work for highly dynamic groups and also guarantee some degree of anonymity. Concepts such as document routing and custom naming have been proposed in this context but are limited to certain application areas.

## III. PEER-TO-PEER CHARACTERISTICS, CONCEPTS AND MECHANISMS

Peer-to-peer is not a well-defined and unambiguously used term. Though, definitions for specific areas have been proposed they usual do not cover all aspects conventionally subsumed under the notion peer-to-peer. For instance Shirkey defines peer-to-peer as a class of applications that takes advantage of resources (such as storage, processing cycles,

human presence) at the edges of the Internet. Peer-to-peer should be regarded as a set of concepts and mechanisms that enable distributed, decentralized computing. Systems that share characteristics associated with the peer-to-peer concept are therefore called peer-to-peer systems. As in the case with family resemblance, any two systems might not have anything in common. However, they still belong to a group that shares (as a group) enough features to consider them peer-to-peer systems.

*A. Structural Features*

One of the major concepts of peer-to-peer computing is ***decentralisation***. This includes distributed storage, processing, information sharing, etc. Even control information can be held in a distributed manner rather than centrally. The advantage of decentralisation is an increased extensibility, higher system availability and improved resilience. On an application level decentralisation can also imply a transferral of ownership and control (of data, information and computational resources) to the application users. However, there are also problems related to this property. In a completely decentralised and dynamic system it is difficult to get or maintain a global view of the system state. Also, the system also does not necessarily behave in a deterministic way. Depending on the number and kind of peers within the group certain request might be answered but this cannot be guaranteed. Another problem is how to join such a group and how to get to know the other peer group members. Further, ***interoperability*** between heterogeneous systems is a problem since they have different characteristics and might even have different interfaces or use different protocols. Peer-to-peer systems are said to be inherently scalable since bottlenecks caused by central instances or servers are largely avoided. The system scalability is usually restricted by the amount of centralized operations necessary. However, a larger number of control messages to co-ordinate the different peer instances might limit the scalability of a peer-to-peer system as well. Hence, it is proposed to use the term ***extensibility*** in this context since certain precautions have to be taken to scale up the number of peers (e.g. intelligent search and message distribution mechanisms have to be used to avoid flooding). However, even if central management and co-ordination is required this can be limited to a well-defined set of operations. Thus, systems can potentially grow very large since resources can be added almost indefinitely. Though, another issue here is that especially in heterogeneous systems no performance guarantees can be given since it is not known which instance is serving a request. Peer-to-peer systems are ***self-organizing*** in a sense that the different system components work together without any central management instance assigning roles and tasks. In this context the degree of organisation in a system increases without an increase of any external governance or system control. The system structure as well as the internal organisation of a peer-to-peer system is ideally built without external control or influence. In self-organised systems it is difficult to determine the system structure or predict the system behaviour as long as no system-wide, governing policies apply. Because of their structure and organisation peer-to-peer systems are inherently ***fault-tolerant***. Since there is no central point of failure they can easily compensate the loss of a peer or even a number of peers. With the loss of peers the system performance might decrease but as long as there is a sufficient number of peers in the system it should still be operational. However, in dynamic peer-to peer systems this also implies that there is a lack of consistency.

*B. Operational Features*

Generally peer-to-peer services and systems are relying on the functions and tasks provided by the peers. The availability of peers can vary considerably; some might be there for long periods whereas others are present for a short time only. Hence peer-to-peer systems are of ***ad-hoc*** nature. The ***performance*** of a peer-to-peer system is *unpredictable*. It depends on the resources (e.g. processing power and storage but also data) made available by the participating peers. Another resource type not (entirely) under the control of the peers is *network bandwidth*. The lack of bandwidth can decrease the overall performance of a peer-to-peer system considerably despite an abundance of all other resources required to compute a request. There are a number of measures that can improve the performance of peer-to-peer systems. Data replication can be used to increase the availability of data. Using caching techniques reduces path lengths and hence the number of messages exchanged between peers. The more copies there are (especially if they are strategically located within the system) the higher is the availability and system performance. Apart from replicating data, processes and services can also be replicated in order to increase availability and thus system performance. Finally, intelligent routing and network organization can help to decrease the number of messages and therefore increase (or more optimally utilize) the available bandwidth. This kind of optimisation also requires knowledge about application and user behavior.

*C. Functional Characteristics*

In distributed systems a main goal is to provide *transparency* to the application or user. Forms of transparency include communication, location, access and replication transparency for data and information. Further, the concurrent access of systems by multiple users, and user and client mobility should be kept transparent. In connection with resilience failure transparency is also an issue. Finally, the *scale* of the system (as long as this does not determine some of its functionality) should be kept transparent to the user. Most peer-to-peer systems provide these forms of transparency utilizing communication system, middleware, platform, programming language, and protocol features. Transparency shields the application and user from the details of the underlying system. However, some information about the system structure might be required at this level.

*D. Protection and Security*

Traditional *security mechanisms* to protect data and systems such as firewalls cannot protect peer-to peer systems since they are essentially globally distributed. On the contrary, such mechanisms can inhibit peer-to-peer communication. Therefore new security concepts are required that protect systems from intruders and attacks while still allowing interaction and distributed processing in peer to- peer systems. Data can be protected by the use of encryption schemes such as public-key-private key encryption. In order to protect IPR (Intellectual Property Rights) while still allowing the public use of content, signatures can be added to the data (e.g. techniques such as watermarking and Steganography). There are a number of methods being developed to deal with these issues. For instance sandboxing, safe languages, proof carrying code and compilers that certify code to protect the host environment; and information flow theory and model checking to protect the data. Other protection issues are related to *privacy* within a peer-to-peer system. Peer-to-peer systems are open by nature. The authorship, the publisher and consumer interactions (such as storage of material and query for certain topics) should be protected. Systems can be classified according to the privacy they provide. These privacy classes include Absolute Privacy, Beyond Suspicion, Probable Innocence and Probably Exposed . There are a number of ways to protect the identity of participating entities. Multicasting and flooding can be used to protect the identity of a receiver/ requestor. Spoofing can be used to protect both, sender and receivers. A covered path can also be used in this context. Further, untraceable aliases can protect participants. Forcing data onto hosting nodes can protect publishers and authors (so called non-voluntary placement) [1]. ***Ownership*** is also an issue in the context of peer-to-peer systems. Costs for system maintenance, communication and storage are usually shared between the peers. Even the costs for content and information can be shared between the operators of the participating peer systems. While cost sharing is an advantage a new problem is who owns the system, the content stored on such a system and the results produced by for instance shared processing. So far peer-to-peer systems and the results they produce have been regarded as public domain goods and assets..

IV. SUMMARY: THE PRESENT AND FUTURE

There was a lot of work did and there is a lot of work to do in the field. It is possible to classify and summarize all the activities in applications and research, present and future.

*A.. Present and Future Applications*

- Support for different communications forms
- Telephony.
- Streaming
- Scalable and flexible naming systems.
- Personal communications (e.g .e-mail).
- Inter organization resource sharing.
- Context/content aware routing.
- Video conference.
- Distribution of learning material.
- Location-based services in Mobile Ad Hoc Networks (MANET), distributed and centralized.
- Context aware service.
- Trustworthy computing.

*B. Present and Future Drawbacks*

Law suits against users.

Software patents.

Intellectual properties.

P2P requires *flat rates* access.Still low bandwidth end nodes. Digital right management. Best effort service insufficient for most applications.

- Lack of trust.
- Commercialization as the end of P2P.
- P2P integrated into other topics.

*C. Research Focus*

The present research work is done on the following things:

- Semantics integration of different information types in the specific peer-database.
- Quality of services criteria (consistency, availability, security, reliability).
- Legacy support in overlays.
- P2P and non-request reply interactions.
- Highly adaptive DHTs.
- Overlay optimization.
- P2P signaling efficiency.
- Data dissemination.
- Resource allocation (mechanism and protocols) and guaranteeing quality of services *P2P* system.
- Self determination of information source.
- Accounting incentive.
- Realistic P2P simulator.
- Decentralize reputation mechanism.
- Semantics queries.
- Efficient P2P content distribution.
- Content-based search queries, metadata.
- Reduction of signaling traffic.
- Data-centric P2P algorithm.
- Content management.
- Application/data integration.
- Security trust, authentication transmission.
- Incentive market mechanism.
- Reliable messaging.
- P2P in mobile cellular/ad-hoc.

In future we can accept to have systems with following features:

- Anonymous but still secure e-commerce.
- Interoperability and/vs. standards.
- ReaP2P for business information systems.

- Real time P2P data dissemination.
- P2P file systems.
- Concept of trust and dynamic security.
- Dynamic content update.
- Distributed search mechanism.
- P2P technologies in MANET.
- Mobile P2P.
- Intelligent search.
- Service differentiation.
- P2P-GRID integration.

Certainly there is a lot of work to do, nothing is over because all is just beginning. The fields of applications is huge. There are excellent readings that should be used for researching and teaching.

## V. CONCLUSION

The peer-to-peer concept has been around for some time. It became recently more popular by the advent of peer-to-peer applications for content and information sharing (e.g. Napster, Gnutella) and distributed processing and resource sharing (e.g. SETI@home). These applications exposed how many resources idle and unused computers currently waste. This has attracted considerable attention and it is thought that the potential of such systems is immense. However, it has to be distinguished between resource utilization through such systems and the actual technology and concepts that enable distributed computing in this context.Peer-to-peer is a very heterogeneous work and research area. Peer-to-peer concepts can be found at the application layer, in distributed systems and within the communication sub-system. There are no major research or standardization initiatives that look at all aspect related to peer-to-peer technology and computing. The major application areas for peer-to-peer methods and technologies are Internet and Web based applications GIRD computing and distributed systems, data sharing, and collaboration. Peer-to-peer concepts are used because they increase scalability (when the known restrictions are taken into account), can improve performance and are inherently fault tolerant. The term peer-to-peer is defined by its usage in different contexts and no formal definition exists. The application areas of peer-to-peer concepts are also too heterogeneous to clearly define a fixed set of attributes peer-to-peer systems have to adhere to. However, there are a number of characteristics any peer-to-peer systems share to realize the advantages of peer-to-peer computing.

## REFERENCES

[1] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne,B. Richard, S. Rollins, Z. Xu: “Peer to-Peer Computing”, hp Technical Report, 2002.

[2] D. Liben-Nowell, H. Balakrishnan: “Observations on the dynamic Evolution of Peer-to-Peer Networks”, Proceedings of the 1st International Workshop on Peer-to-Peer Systems, Cambridge, USA,2002.

[3] Open Napster Project, “Open Napster Messages”, http://openap.sourceforge.net, 2000.

[4] F. Kileng: “Peer-to-Peer File Sharing Technologies: Napster,Gnutella and Beyond”, Technical Report, 18/ 2001 Telenor,http://www.telenor.no/fou/publisering/Rapp01/R18_2001.PDF, 2002.

[5] MetaMachin,eDonkey2000,http://www.edonkey2000.com/, 2002.

[6] K. Kant, R. Iyer, V. Tewari, “On the Potential of Peer-to-Peer Computing: Classification and Evaluation”, http://kkant.ccwebhost.com/download.htm, 2002