How New Peer to Peer Developments May Effect Collaborative Systems

Dr. Daniel C. Hyde
Department of Computer Science
Bucknell University
Lewisburg, Pennsylvania USA 17837
hyde@bucknell.edu Phone: (570) 577-1281 Fax: (570) 577-1822

Abstract

This paper describes the latest peer-to-peer developments in designing distributed systems. It discusses key characteristics and issues surrounding the use of the latest peer-to-peer approaches and possible effects on designing collaborative systems. It also presents a classification scheme for peer-to-peer systems to aid in the design of collaborative systems. It provides pointers to some current peer-to-peer collaborative projects and attempts to predict the direction of future developments.

Keyword Phrases

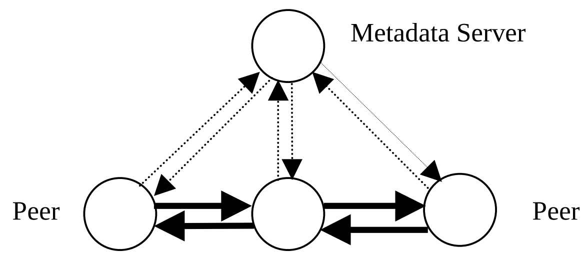
peer-to-peer, P2P, collaborative systems design, collaborative enabling technologies

INTRODUCTION

This paper describes the latest peer-to-peer (P2P) developments in the area of computer science called distributed systems [Coulouris]. It explains where these new developments may influence the design of collaborative systems. Even though collaborative systems designers have used P2P ideas in the past, they may benefit by studying the latest P2P trends.

In the last year, with the high profile case of Napster, Internet watchers have proclaimed "peer-to-peer is the next great thing for the Internet." High Tech journalists have scrambled to try to understand and write about P2P. Many companies have jumped on the bandwagon and claimed P2P products. As with many over-hyped concepts, P2P is variously and loosely defined and poorly understood by many.

Investigating the technical literature reveals that P2P's common theme is a participant-oriented view of the world [Fox1; Oram]. A participant is a machine of an end user, i. e., at the edge of a network. A participant host may have both a server and a client. The participants (peers) communicate with other participants (peers). However, P2Ps do not need to be totally decentralized. A P2P system may have a centralized metadata server, say to maintain a name service. However, any central server is subservient to the participants. The participants do most of the work. This



Typical Peer-to-Peer Architecture

is in contrast to the traditional client- server model where powerful servers perform extensive processing to handle thin clients such as web browsers. The metadata server maintains the data needed for participants to register and coordinate the participants.

This paper explores the characteristics of P2P systems, their issues and their possible impact on collaborative systems.

BACKGROUND

The technical aspects of a computer-based collaborative system are a type of distributed system. A collaborative system is a collection of collaborators and mediators that interact for a common purpose. In general, a collaborator could be a human, a software program that interacts with a human or an autonomous software agent. For this paper, we assume a collaborator is a software program interacting with a human. In collaborative systems terms, the participants of P2P would be collaborators and the metadata server would be a mediator.

Napster

Napster (www.napster.com) is the most well know and popular P2P system. Developed by Shawn Fanning, while a freshman at Northeastern University, Napster lets any participant advertise the MP3 music files stored on its local disk and allows other participants to copy the files. It is P2P as the copying of the MP3 files is directly participant to participant. The participants do the bulk of the work. The system is not purely decentralized as a centralized list of participants and information on the music available at each participant is maintained. By the end of 2000, the system had become staggeringly popular with over 75 million users. Like most good ideas, Napster was designed to solve a real need – to enable Fanning, a musician, to share his music with his friends on campus. However, the Napster file sharing mechanism easily allows individuals to "publish" a music piece without being the artist or "author." With massive copying of copyrighted music -- Estimated at 10,000 music works a second! -- the music industry felt it had to take legal action.

Napster usage became so popular especially on college campuses that it disrupted the Internet service of others. Many university network administrators became alarmed when Napster traffic became over 60% of their Internet connection's bandwidth and as a result blocked Napster access on their campuses. With Napster's disruptive behavior, popularity and high profile court case, many people took notice and proclaimed P2P the next great thing for the Internet.

P2P Is Not New

However, P2P is not new. P2P was used for years before the Internet existed in mobile police radios. To increase the range between the police car and the station as well as bypass obstacles like mountains, the communication signal was automatically relayed through other police cars in between. Therefore, a good use of P2P is to increase the coverage of a wireless system. A similar approach is used in community wireless networks such as www.Guerrilla.net in the Cambridge and Boston area which is run by rugged individualists who desire an underground alternative to the wired Internet and independence from possible government regulation and business interests.

The original Internet of 15 years ago was based on the symmetry of hosts as equals. The original designers worried about the vulnerability of centralized control. They realized that such an approach would not scale and a failure of the central control could bring down the whole network. Therefore, IP routers operate as P2P. Routers talk to routers. There is no centralized control for Internet routing. This foresight has allowed the Internet to scale to millions of hosts far exceeding the designers' wildest dreams. Also, the scheme used by IP routers is tolerant of failures. If a router fails or a connection to a router fails, another route for the IP packets is found and taken. Another important P2P-based system of the Internet is the Domain Name Service (DNS) which resolves human-friendly names like "www.eg.bucknell.edu" to 134.82.131.10, an IP address. DNS servers talk to other DNS server to resolve domain names and if one DNS server does not know the address it relays the request on to others until the name is resolved. A third widely used Internet service that is P2P based is Usenet that distributes individual's postings on newsgroups around the world. We recommend that any collaborative system designer considering a P2P-based approach carefully study the designs of IP routing, DNS and Usenet and the design issues encountered by these successful P2P systems.

If successful P2P systems have been fundamental to the Internet from the start, why all the current hype and fuss about P2P being the next best thing for the Internet since the Web?

Explosive Growth of Internet Due to Web

Starting in 1994, the Internet experienced explosive growth mainly due to the introduction of the Web. Most of the growth was very positive but there were several developments that had detrimental effects on the design of P2P systems. First, the web allowed Internet access to millions of common folk from grandmothers to first graders. The typical Internet user was no longer a computer geek. A tiny fraction of these new members ignored Internet etiquette and, for personal gain, abused their new found capability, e. g., by using SPAM. The use of SPAM, the sending of massive amounts of unsolicited junk messages, just about destroyed the P2P-based Usenet that assumed individuals will cooperate and be nice. Many users were forced to abandon the newsgroups of Usenet and form their own moderated list services. Any designer of new P2P systems must worry about the possible actions of this small group of unscrupulous individuals.

Second, to keep unwanted individuals from accessing their machines, many organizations installed firewalls. These firewalls were effective in enhancing security, however, they made designing P2P systems harder if some of the participants were behind the firewalls.

A third consequence of the explosive growth of the Internet was the shortage of usable IP addresses. To solve this problem, some organizations installed Network Address Translation (NAT) boxes between the Internet and their internal networks with the result that no longer did each host have a unique IP address. Many P2P-based systems can't handle NAT boxes. With wide acceptance of the latest Internet Protocol (IPv6) with its much larger IP addresses (128 bits versus 32 bits in IPv4), this problem would go away. However, adoption of IPv6 has been slow.

With millions of new users, Internet Service Providers (ISPs) had many technical problems to solve. One was how to assign IP addresses to dialup users. Their solution was to use dynamic IP addresses, i. e., to reuse an IP address after the end of a dialup session.

A second technical problem faced by the ISPs was how to allocate bandwidth. Many broadband ISPs use broadband technologies, e. g., cable and ADSL, that provide a much higher bandwidth (as much as a factor of 10) down to the user machine then the up link to the Internet. This approach assumes the typical user will be relatively passive and perform only actions such as read email and surf the web. Since a participating machine of a P2P-based system is really both a server and a client, bandwidth is needed in both directions. Customers of broadband ISPs that were participants in Napster-style uploading of large files have been disconnected not because of illegal copyright concerns but for utilizing too much bandwidth. This was justified by claims that such use was in violation of their service contract. Any designer of a P2P-based system must consider the asymmetry in the bandwidth of many possible participants.

Problem with Dynamic IP Addresses

The original Internet was designed with the assumption that each host would have a unique and static IP address. The assumption of static IP addresses made the design of routers much easier as the routing tables would need to be updated only infrequently. Wide acceptance of IPv6 will help the situation with non-unique IP addresses with the elimination of NAT boxes. But it will not eliminate the need for dynamic IP addresses. Even if we eliminated all dialup users, dynamic IP addresses are still important for mobile computing. As a laptop user roams from building to building, a new IP address needs to be assigned dynamically to her laptop (usually assigned with the Dynamic Host Configuration Protocol (DHCP)). Therefore, the designers of new P2P systems must assume dynamic IP addresses.

In a P2P-based system, a participating machine is both a server and a client. However, with dynamic IP addresses, a P2P participant can not always contact the server portion of another participant. The IP address of the server portion of the participant is unknown because it may not be assigned yet or was just recently assigned.

A Solution to Dynamic IP Problem

To solve the problem of dynamic addresses resulting from intermittently connected PCs, ICQ (http://www.icq.org), the first PC-based chat system (launched in 1996), created their own directory of protocol specific addresses (e. g., DNS and dynamic IP addresses) for each user. This centralized metadata service updates new IP addresses in real time. Groove (http://www.groove.net), Napster and commercial instant message systems such as AOL Instant Messenger (AIM) (http://www.aol.com/aim/) follow the same approach. Updating in real time is critical for implementing the popular "presence feature" that alerts a user when a buddy comes on-line or goes off-line.

An interesting side benefit of maintaining this large directory of IP addresses is that some P2P systems such as AOL Instant Messenger and ICQ create names that refer to human beings and not machines. As people move around the planet from machine to machine, the system updates their latest machine address. Designers need to realize a user is not tied to one machine. Maintaining this directory for a world-wide P2P system such as Napster requires significant resources. Napster had created more than 23 million non-DNS addresses in 16 months [Oram, page 24]. Unfortunately, each new P2P system has to recreate a similar directory structure. Each commercial P2P has it own closed directory scheme that is incompatible with the others. There is currently no universal service one can tap. See the Jabber Project [Oram, Chapter 6] (http://www.jabber.org) for one attempt at solving this. As part of Microsoft's new .Net initiative, Hailstorm [Shirky] is Microsoft's solution to a world-wide registry of net addresses for users. However, many developers mistrust Microsoft creating such an important infrastructure. A vendor-neutral solution is needed.

A CLASSIFICATION OF P2P SYSTEMS

Pure P2P

A pure P2P system has no centralized server. All communication is completely decentralized and only between participants. Building a pure P2P system with only two participants is easy. It is when the number of participants becomes large that a pure P2P is hard to implement. Each participant needs a way to acquire the information, e. g., host id and port number, on how to connect to the other participants. In a pure P2P system this information is obtained outside of the system, e. g., by telephone or lookup on a known web site.

Gnutella was an early attempt at a pure P2P. In the early days of Gnutella, the way you found your way onto the network was by word of mouth.

Registry P2P

The second class of P2P is a registry P2P. Many P2P systems use a metadata server primarily or solely to register new participants. The Gnutella network uses a series of servers called host caches. Host caches provide a jumping off spot for Gnutella users. It is a host that's always running that gives a place for a participant's Gnutella software to connect to and find the rest of the Gnutella network. Once connected, the participant no longer needs the host cache.

Coordination P2P

The third class of P2P system is a coordination P2P where the metadata server not only registers new participants but coordinates activity between participants (collaborators). The participants still do most of the work. However the metadata server (mediator) coordinates activities that must be centralized such as voting or acquiring exclusive access rights to a resource.

Hierarchical P2P

The fourth class of P2P system is hierarchical P2P where the metadata servers are connected in some hierarchy. In large-scale multi-user Internet games over 1000 users may be playing simultaneously. Since one metadata server can't possible handle the workload, a tree of communicating metadata servers are used.

DESIGNING P2P-BASED COLLABORATIVE SYSTEMS

Designing a new collaborative system as a P2P may have many advantages. First, the basic idea behind P2Ps is to push more work out from servers to the participants. This allows the system to have a smaller, less expensive server compared with the traditional client-server model used so extensively in the last decade. Today's typical PC has plenty of CPU cycles and disk storage. These PCs are much more powerful than the typical user thinks. Their machines should not be relegated to passive "terminals" for presenting content as millions are used today. A goal of P2P systems is to try to effectively utilize the rich resources of the machines at the edge of a network.

Second, a P2P-based system may help with bandwidth. Since participants communicate with other participants directly, the communication bandwidth requirements of any central server is less. For example, in a Napster-style system if all the MP3 files had to be downloaded from a central server for each request, one server could not handle the load (estimated to be over 10,000 music works per second). Very high *total* bandwidth is possible with a P2P system.

Collaborative systems are designed for sharing. P2P-based systems with participants communicating with other participants can facilitate this sharing. A P2P system could share basic computer resources such as storage for a very large store, CPU cycles as in the SETI@home project (http://www.setiathome.ssl.berkeley.edu/) as well as share content, e. g., music files or design documents. P2P are naturals for sharing human conversation, human presence (Who is available?), decision making, editing, voting and other human activities.

Fourth, P2P systems can be more democratic and return content, choice and control to ordinary users. Unfortunately, because of this ability the popular press has presented P2P in a negative light. Many early P2P projects have an overtly political mission: routing around censorship. P2P techniques provide a level of obscurity from control by government or other agents who might exert control over any centralized approach. For example, Napster-like file sharing systems use P2P techniques to avert legal action from possible copyright infringements. Collaborative system designers need to sort through the political rhetoric and bad press and discover the technical merits of P2P systems.

Fifth, P2P systems should be easier to design in the near future. Many key industrial players are investing lots of money and energy in infrastructure and tools to create P2P-based systems. On April 25, 2001, Bill Joy, chief scientist at Sun Microsystems, presented an online Webcast on JXTA, Sun's P2P initiative. The JXTA initiative will provide infrastructure services for P2P applications [JXTA]. JXTA, pronounced "juxta" represents an abbreviation for the word "juxtapose" which according to Joy means "putting things next to each other, which is what peer-to-peer is about." JXTA will work in conjunction with Sun's Java and Jini in what Sun claims will provide a complete approach to distributed computing.

While Sun Microsystems has been direct in stating that JXTA is for P2P infrastructure, Microsoft has not been so helpful. Microsoft's new mega-initiative .NET appears to have many P2P attributes [.NET]. It is hard to read between the lines in Microsoft's rhetoric, but their press releases speak of pushing more work out to the end user PCs with services that will allow the PCs to work together to deliver broader, richer solutions. PCs will be able to collaborate directly with each other. All of an individual's files will sit on the Internet (like Napster music files) and be accessible from anywhere (with guarantees of privacy and security, of course). Using Napster as an example, Bill Gates and his team announced that .NET will facilitate how the Internet is moving toward many-to-many interactivity, where every computer can be both server and client. Though the message is murky, .NET sounds like an ambitious P2P infrastructure initiative.

However, not all collaborative systems should be designed as P2P. If your collaborative system has a large central database, the traditional three-tiered approach with client-server and database may be more appropriate. Any accountability or correctness of shared information should be centralized. For example, in an online auction, the algorithm and data for determining the winner should be centralized. Designing a correct distributed algorithm (Remember no global clock!) for such situations is tricky.

A collaborative system must be designed for social expectations, e. g., collective cooperation. The designer must worry about ill-behaved individuals (Remember SPAM!). Giving up too much control to participants in a P2P may be undesirable. Some multi–user Internet games were won by unscrupulous individuals rewriting their local game client to give themselves huge unfair advantages. The game designers had to redesign the software with accountability checks. After SETI found several individuals altering the results in order to claim they had found extra-terrestrial life, SETI had to add special encrypted checksums to their work units.

Some P2P-based synchronous collaborative systems, e. g., ICQ and AOL Instant Messenger, seem to be geared for "teenager chat." One has to wonder if the same P2P techniques are appropriate for industrial strength collaborative systems such as required in the military, health care, engineering design and distance learning. Users have legitimate concerns with privacy and security. System administrators want tools for routine tasks such as periodic backup of local data to avoid lost data and automatic dissemination of new releases. Providing these features and tools may make the P2P route less attractive.

One issue is the lost of local data. Napster does not worry about lost or unavailable music files, i. e., it tolerates unreliability. By using redundancy, – the same music files in many places – the users can find the music they want. SETI also tolerates unreliability and uses redundancy. If a work unit is not returned after a specified time, SETI sends it again to some other participant.

ADDITIONAL P2P RESOURCES

For those interested in learning more about P2P, Geoffrey Fox has written a good introductory article with many references to P2P projects and organizations [Fox1]. At Florida State University, Fox is researching and developing his own collaborative system called the Garnet Collaboration System. His Web site has many further references [Fox2].

Tim O'Reilly, founder and CEO of O'Reilly & Associates, Inc., organized on September 18, 2000, a so-called "peer-to-peer summit" of many key developers in P2P. One outcome was the valuable book on P2P edited by Andy Oram of O'Reilly & Associates, Inc. [Oram]. This book is a *must* read for any person interested in P2P design. Written by over thirty P2P researchers, it consists of nineteen chapters on the context and overview of P2P systems, specific projects, and technical issues such as metadata management, performance, trust and security.

Tim O'Reilly has organized a third P2P conference that occurred in Washington, D.C. November 2001 [O'Reilly2]. The reader can find general discussions on P2P technologies at the Web sites for the O'Reilly P2P Working Group [O'Reilly1] and the P2P Industrial Working Group originally initiated by Intel [Intel].

CONCLUSIONS

The main characteristic of P2P systems is that the focus is on the participants (collaborators). The participants communicate between themselves and do most of the work. We have classified P2P systems into four categories: pure, registry, coordination and hierarchical. These reflect the role of the metadata server (mediator)

If participants need to communicate directly, a collaborative system may benefit from a P2P approach. The P2P approach needs less central server support, has potential for very high total communication bandwidth, more work out of the participant machines and more effective sharing. Key industrial players, including Sun Microsystems with JXTA and Microsoft with .NET are developing P2P infrastructure which will make designing P2P systems much easier.

Many technical issues need resolving. Missing is a universal user naming scheme for dynamically updating IP addresses in real time. Other infrastructure that is missing is a Web server that ordinary PC users can easily install and use on their PC. Users want privacy and security. System administration tools to facilitate such tasks such as periodic backup of local data and automatic dissemination of new releases will need to be developed for P2P systems.

REFERENCES

[Coulouris] Coulouris, George, Jean Dollimore and Tim Kindberg, *Distributed Systems: Concepts and Design*, third edition, Addison-Wesley, 2001.

[DISCIPLE] DISCIPLE (DIstributed System for Collaborative Information Processing and LEarning) Project at Rutgers. Web site at `http://www.caip.rutgers.edu/multimedia/groupware/`

[Fahraeus] Fahraeus, Eva, Noel Bridgeman, Joze Rugelj, "Teaching with Electronic Collaborative Learning Groups," *SIGCSE Bulletin*, Vol. 31, No. 4, Dec 1999.

[Fox1] Fox, Geoffrey, "Peer-To-Peer Networks," *Computing in Science & Engineering*, May/June 2001, pages 75-77.

[Fox2] Fox, Geoffrey, maintains the Garnet Collaboration System Web site at `http://aspen.csit.fsu.edu/collabtools/`

[Intel] Industry P2P working group originally initiated by Intel. Web site at `http://www.peer-to-peerwg.org/`

[JXTA] Sun Microsystems JXTA web site at `http://www.jxta.org/`

[.NET] Microsoft's .NET Web site at `http://www.microsoft.com/net/`

[Oram] Oram, Andy, *Peer to Peer: Harnessing the Power of Disruptive Technologies*, editor, O'Reilly & Associates, Inc., 2001.

[O'Reilly1] O'Reilly P2P Group Web site at `http://www.openp2p.com/`

[O'Reilly2] O'Reilly's P2P & Web Services Conference in Washington D. C., Nov. 5-8, 2001 at `http://www.openp2p.com/pub/a/p2p/conference/dc_con.html`

[Shirky] Shirky, Clay, "Hailstorm: Open Web Services Controlled by Microsoft", *O'Reilly Network*, May 30, 2001, at `http://www.openp2p.com/pub/a/p2p/2001/05/30/hailstorm.html`

BIBLIOGRAPHY

Dr. Daniel C. Hyde is an Associate Professor in the Department of Computer Science at Bucknell University, Lewisburg, Pennsylvania, USA. His research interests include parallel and distributed computing. Dan received the BSEE degree from Northeastern University and the PhD degree in computer science at the University of Illinois, Urbana-Champaign. He is currently the newsletter editor for the IEEE Computer Society Task Force on Cluster Computing.