Peer-to-peer Communication

Jari Sukanen
Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
Jari.Sukanen@hut.fi

Abstract

Internet has changed a lot since its early days. At the beginning, goal of the Internet was to share computing resources between universities and scientists. In those days communication happened almost completely directly between computers or people. This is something we now call peer-to-peer communication.

After the commercialization of Internet, millions and millions of home users started to use different services like web and e-mail. This led to a situation where new technologies for addressing and communication had to be developed, client/server model was born. For some years users were happy to play just the consumer role and there were relatively small amount of servers that these users connected to. Client/server model was the dominant model of communication but some other minor models co-existed.

It's quite clear that client/server model is not the best model for communication between people since it was designed for completely different purpose. This creates a need for other models that let people to share knowledge and resourses over the Net. Peer-to-peer communication model is just what is needed for this kind of two-way communication and it also lets people to act both consumers and producers of data.

This report focuses mainly on Microsoft solutions for peer-to-peer communication analyzing current products and technologies. Analysis is done with respect to home environments but also other aspects are briefly considered. In home environments especially wireless communication is studied.

1 Introduction

1.1 Background of the Study

Since 1994, not only scientists but the general public as well have been joining to world wide community of Internet users, placing many restricions and high pressure on used technologies. Client/server was the model of the transitional period and the Net has survived very well scaling to much larger amount of users than originally planned. [14]

In the last few years something has changed. Users of the Internet are striving to find new ways to take advantage of the enormous capabilities of the Net. Communication in the Internet has kind of reverted towards the model that was in use before the great expansion. It was probably the music-sharing application Napster that really started the change. There were signs of it earlier but Napster was the “killer application” that led to intensive use of P2P model. [9, 14, 16]

Nowadays people want to do much more than just browse the web or send e-mails. They want to communicate directly with other people, form discussion groups and share resources like computing power and files. P2P communication is something that enables this without intervention of public servers. It also makes more efficient use of bandwidth as there aren’t so many bottleneck servers anymore. [1]

As the role of the service infrastructure diminishes in the P2P model, it seems to fit quite well for home environments also. In home environments there are typically many devices that share different resources like printers and disk storage. However, it is not feasible to maintain one server that serves all devices. More elegant approach is to use P2P and let devices communicate directly with each other.

1.2 Scope of the Study

This study concentrates on presenting the basics of P2P technologies and analyzing Microsoft technologies and solutions for P2P communication. Some wireless technologies for home computing are also studied and their uses are discussed briefly.

Finally, an open solution called Home Audio Video Interoperability (HAVi) is studied from the viewpoint of home computing and P2P communication.

1.3 Definition of Terms

.NET	XML Web services platform that "lets applications to communicate over the Internet, regardless of operating system, device, or programming language".
.NET Alerts	Microsoft's communication medium that lets service providers to send short instant messages to people who opt to receive them.
.NET My Servces	Family of XML Web services that builds on Passport and Alerts. Formerly known as Hailstorm.
.NET Passport	Authentication system for .NET My Services. .NET Passport is alsod used by many commercial site as their authorization system.
Gnutella	File sharing application
Hailstorm	See .NET My Services
HAVi	Home Audio Video Interoperability
ICQ	Another IM application, shorthand for "I Seek You".
IE	Internet Explorer, Microsoft's Web browser
IEEE	Institute of Electrical and Electronic Engineers
IM	Instant Messaging
LAN	Local Area Network
MSN	Microsoft Network
Napster	MP3 sharing application
P2P	Peer-to-peer
PAN	Personal Area Network
PUID	Passport Unique ID
URL	Universal Resource Locator
WLAN	Wireless LAN (see LAN)

2 Peer-to-Peer Communication Model

2.1 Overview

P2P model is a communication model in which each party may act as a client and a server. Both parties have the same capabilities and both can initiate connection. It is the opposite of client/server model in the sense that there is no central entity that the other parties contact but every single entity is able to initiate connection directly with all other entities. [14]

Pure P2P communication doesn't need any kind of central server or database to make a connection. In addition to pure P2P there are a lot of solutions that are neither pure P2P nor client/server. They combine charasteristics from both models. What makes them more P2P than client/server solutions is that the communication channel that is established will be directly between communicating parties rather than through some central server. [13]

Napster, for example, uses a hybrid communication model in which there is one central server (or a bunch of servers) that manages mappings from names to addresses. Napster servers are just brokers that relay requests to the place where the actual data is. [14]

The great idea behind P2P communication is the "Content at the Edges" data model. This is completely different than traditional Web's data model that could be called "Content at the Center". What makes content-at-the-edges data model superior is the discovery that most of the content in the Internet is created at the edges, not at the center. Using P2P model content can be kept at the edges and there is no need to upload it to the center. As content is kept at the edges, P2P also harnesses the enormous power of the millions and millions of PCs that are connected to the Internet. [13, 14]

P2P also makes more efficient use of the limited bandwidth. It releases servers from acting as mediators in communication and there is no need for data to pass through the server. This lets communicating parties use the fastest line between them.

Besides Napster, there are ever increasing number of products that make use of benefits that P2P offers. These are Gnutella, SETI@home and Freenet just to mention few. Microsoft has also been working on P2P technology intensively and they have announced .NET product family that provides a rich platform for building P2P applications [13].

2.2 Detailed Description

There are many differencies between different kind of P2P communication methods. The main differencies are in the following three key features: discovering of other peers, querying peers for content and sharing content with other peers. For discovering other peers traditionally the most used way is to use a discovery server that registers peers and lets other peers query for list of available peers. However, there are also many other methods available. [13]

In the following we will examine four different approaches for P2P communication. Of course, there are many other architectures as well and some applications may use multiple of these or some kind of combination of all four.

2.2.1 Pure Peer-to-peer

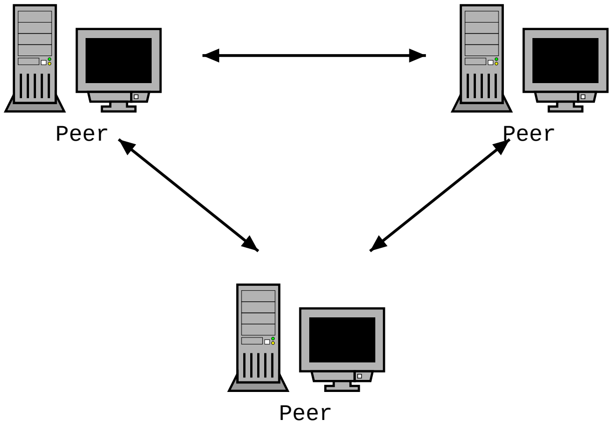

Figure 1: Pure P2P [13]

Figure 1 illustrates the structure of pure P2P network. As can be seen from the picture there are no central servers but peers connect directly to each other.

Peer that wants to connect to some other peer dynamically discovers other peers. Discovering process can be handled by using for example broadcasting or multicasting. It is also possible to use for example a well-known node approach to find other peers. In well-known node method all peers know at least one other peer that is known to exist and this way peers create a large set of nodes that are aware of each other via some other node. In the Internet when it's not usually possible to use multicasting or broadcasting some other method has to be used. [13]

Pure P2P applications benefit from the lack of central server since they don't have to register to any machine before other peers can connect. This also poses a problem for pure P2P approach since there has to be an alternative method for finding peers than central server. [14]

2.2.2 Peer-to-peer with a Simple Discovery Server

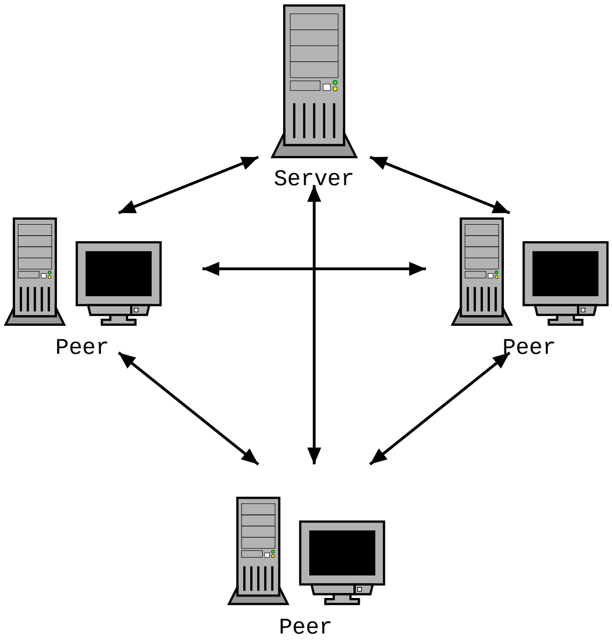

Figure 2: P2P with a simple discovery server [13]

Figure 2 represents the configuration of a P2P network with simple discovery server. In this configuration there is one central server that is used to discover locations of other peers. Usually this is implemented so that when application starts it registers to discovery server that keeps track of machines that are available. When a peer wants to contact another it first queries discovery server for a list of locations and then one at the time makes queries directly to peers in the list. [13]

This approach can be made more easily much more scalable than pure P2P architecture since discovery of other peers is quite trivial. The major drawback is that this method relies heavily on the central server and can't operate at all without it. Another disadvantage is that making queries to each peer individually takes a huge amount of time if working in an environment like Internet where there could be millions of peers operating at the same time. [13]

2.2.3 Peer-to-peer with a Discovery and Lookup Server

P2P with a discovery and lookup server architecture is very similar to the one in figure 2. The only difference is that the server also functions as a lookup server. This means that when a P2P application starts it not only registers to server but also uploads the list of content that it is providing to the server. When contacting another peer application first makes a query to the lookup server that responds with an address of the peer that provides content that was queried. [13]

This approach relies even more on the server than previous one but it also scales better since content queries are much faster and they also use less bandwidth. [13]

2.2.4 Peer-to-peer with a Discovery, Lookup, and Content Server

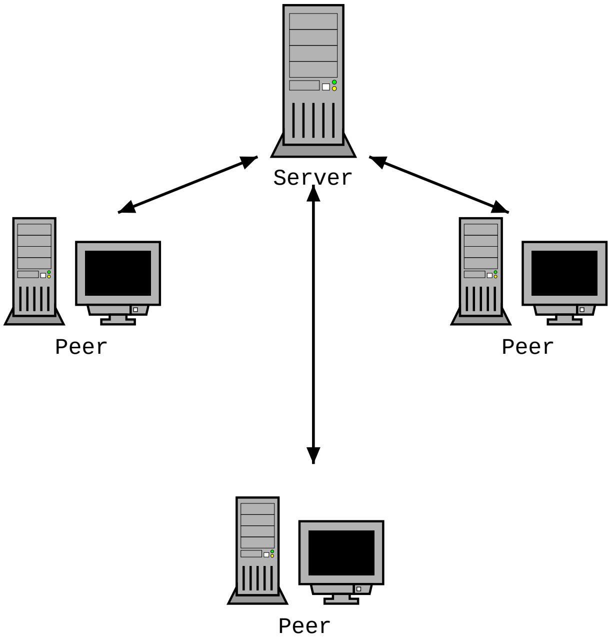

Figure 3: P2P with a Discovery, Lookup, and Content Server [13]

Figure 3 shows structure of a network where there are three peers and a discovery, lookup, and content server. This last approach is here to only show that P2P model can be reduced back to client/server model if we rely more and more on the server. [13]

In this model peers (clients) upload all content that they have to provide to central server so that any other peer can access the content even when the provider is not online.

The drawbacks of this model are those of client/server model, ie. server easily becomes too burdened when there are many clients and if there are many providers the server may run out of disk space. It is also quite expensive to maintain this kind of server. [13]

3 Microsoft Solutions

3.1 Overview

Microsoft has announced their new technology that they say is about to set the guidelines for the next decade or so. They call this new platform Microsoft .NET. .NET is an XML Web services platform providing services that “can be called across platforms and operating systems, regardless of programming language”. [11]

Microsoft has realized that focus is moving more and more towards networked services and farther away from traditional desktop applications. .NET is their response to this evolution. It introduces computing model that is essentially P2P expanding client/server model to distributed computing. There is no clear distinction between client and server but computing takes place where it makes most sense. .NET as well as other P2P platforms and applications changes radically the way we think about networking and therefore also application development. [13]

Microsoft has already released Visual Studio .NET that lets developers build applications using P2P paradigm. Visual Studio .NET is built XML Web services in mind so that it enables applications to share data over the Internet using XML regardless of operating system and programming language.

There are also a couple of applications that Microsoft has developed using P2P paradigm. In the following some of these are studied and analyzed focusing on their use in home environment but also briefly considering a broader view.

3.2 MSN Messenger

Microsoft MSN Messenger is an instant messaging application that is built using P2P technology. It is Microsoft’s solution to people’s ever increasing need to be reachable all the time. Just like any other instant messaging software, Messenger lets users send messages to each other in real time. Instant messaging lets people interact much faster than using e-mail and also enables many other useful ways to communicate. With most of the instant messaging software it is possible to see if someone is online or not and therefore direct urgent messages to people who are reachable at the time. This is very useful for example when someone working at home wants to see if some of his/her colleagues is ready to have a short discussion. [8, 10]

In addition to normal instant messaging, Messenger also offers many other useful features. File sharing has become almost a standard in instant messaging applications so it is provided of course. In addition to these basic aspects Messenger also lets users make voice and video conversations. If both users have speakers and microphone it is possible to make direct voice conversations between two computers. Using .NET Voice Services it is even possible to make phone calls to telephone or mobile device using prepaid account with a voice service provider. [10]

During recent years instant messaging has become a very important tool for many individuals but it is an important device for business as well. In many workplaces managers are concerned about security so it is an essential aspect to consider when discussing about instant messaging. There have been several reports about bugs that let other users to hijack Messenger connection and steal contact lists. Microsoft has already released an update but it is certain that vulnerable versions are still used. Since Messenger works closely with IE, some of bugs in IE are security holes for Messenger also. [8, 14]

Messenger handles privacy using Microsoft .NET Passport that is discussed in 3.4. Because Passport is used, user has to create a Passport account before Messenger can be used. This is normally done creating a Hotmail account that seems like another Microsoft's intrigue to gather more Hotmail users since Passport could easily have been a completely different service. [8, 10]

Because there are no widely accepted standards established in the instant messaging world, there are many different applications that can't talk to each other. This divides people to different camps that are all using different software and cannot communicate with each other. Actually this is not quite true since Microsoft has made Messenger an open standard and submitted it to open standards body IETF [5]. In the future we shall see if there will be co-operating messengers or will there be a messenger war just like there is/was browser war.

There is no direct information about what kind of P2P solution Messenger uses but most likely it uses an approach similar to the one described in 2.2.2.

Messenger is quite similar to many other instant messaging software and there are many older competitors in the area, for example ICQ. However, Messenger integrates very well with other Microsoft products like Netmeeting and Outlook and many people say that Microsoft has catched up the gap that there was between Messenger and ICQ [5]. Since ICQ has been in the business longer, it has many devoted users but it still cannot compete with Messenger because Microsoft bundles Messenger with its Windows XP and it is likely to spread very fast. It is also important that it's very easy to start using Messenger since it is already installed in many new computers.

3.3 .NET My Services

.NET My Services is an XML Web services infrastructure. It lets developers create user-centric webservices that offer high level of personalization. Using Passport service (see 3.4) My Services integrates all the bits of information that the user holds to be accessed anywhere. As Microsoft uses open standards, all this is independent of the used device, network and operating system. People can reach their information anywhere using PDAs, PCs at work or at home, WebTV or any other device that is connected. Passport service also enables another important feature: user can decide what information he wishes to publish, to whom and for how long time. My Services also uses .NET Alerts service that lets users have notifications for e-mails but also to create their own alerts for example to monitor web sites. [2, 4]

In P2P world .NET My Services works as a glue to bind different P2P services that nowadays work as different applications. This means that in the future P2P applications may work seamlessly with each other providing value-adding services directly to each other. It is not necessary to have separate file sharing functionality for each application but every service can use the one file sharing service that My Services brings to them. [4]

There is one big issue when integrating all the information under the same service and it is the trust. Using Passport and My Services people trust very much on Microsoft. When centralizing all information under one authority user reveals very much information to that authority. Goal of the integration is very good and would work nicely if Microsoft was indisputably trustworthy. Even though we say Microsoft here, it could be any other company and the problem would be the same. [4]

But when considering the Microsoft's point of view, the question is not if Microsoft is trustworthy but if the end-users will start to use the system. Because of the position of the Microsoft, end-users probably will want to use system like this. It is very tempting to address all the information everywhere and be able to address all the services using just one username and password. Most of the people don't even realize that there is some kind of problem here, they just take this as a good free service that is "about to change their lives" as Microsoft puts it.

In pressrelease Microsoft says about XML Web services: "no Microsoft runtime or tool will be required to use them". This is indeed something new for Microsoft, a company that has kept its technology secret quite carefully. If this really means a truly open system that lets developers create application for example in Linux environment using Passport services, this is a huge step for Microsoft.

3.4 .NET Passport

Microsoft .NET Passport is a centralized user authentication service designed to allow easy and secure authorization of users. It lets people to use participating sites with only one sign-in username and password. Participating sites rely on Passport service in authentication and they don't have to build their own authentication system. There is a huge number of accounts in the Passport service because every MSN Hotmail account is also a Passport account. [12]

.NET Passport also offers another service called "Passport wallet". This lets people to store their credit cards and addresses among the other information. This enables fast online shopping as participating sites can fetch needed information straight from Passport servers. [12]

If the user is already authenticated by .NET Passport, user can sign-in on a participating site using just one click. After user has clicked "Log in" button, he is silently redirected to Passport server with URL of the participating site. Login server checks the validity using site ID and ticket data which consist of Passport Unique ID and time stamp. After that participating site recieves encrypted ticket data and user profile and can let the user in. This way no password checking is done and user don't have to log in to every site he uses. [12]

The concept of single sign-in is attractive but it also poses the same problem as using .NET My Services (see 3.3), a lot is trusted in the hands of Microsoft. It also creates some security issues that have to be considered. Is it really safe to use single sign-in? Marc Slemko has showed in his article that it is possible to steal session from Passport server and with that session comes credit card information and everything. Microsoft has already fixed this particular bug but in a system like this there is likely to be more of them. [12, 17]

Microsoft .NET Passport is P2P in a sense that it centralizes authentication. However, it is clearly not pure P2P and that becomes a problem when considering denial of service attacks. If someone would like to take out as many commercial sites as possible, he would most definitely try to take out the Passport service. This would paralyze all the sites that are using Passport authentication. [17]

In addition to real technical problems it is always a human who has to remember the password. There are far too many people in the world who use their son's name as a password or store their secret words under their keyboard. Microsoft also tries to address this problem by educating users. They say: ".NET Passport aims not only to provide the best technology but also to educate users on good practices". [12]

4 Peer-to-Peer Wireless Home Communication

4.1 Overview

Wireless communication is becoming more and more important technology especially in home networks. It has major advantages over traditional local area networks including absence of cables and flexible moving inside apartment and also outside.

P2P brings many useful features to home network. It lets files reside on any machine and other computers can access those files seemlessly using peer-to-peer communication protocols.

Probably the most important thing that P2P computing adds to the wireless computing environment is that every node in P2P network can act as a router. That way there is no need for central access point that relays queries. Also, every device doesn't have to be near the central access point but it is sufficient to be near a peer that is connected to access point (probably through many other peers). This so called Multi-Hopping capability lets nodes orient themselves to a robust network that automatically routes around line-of-sight obstacles and congestion points. [15]

4.2 Technologies

There are several technologies to choose from when designing wireless home computing environment. Here we will concentrate only to the two most important ones: Bluetooth and WLAN (IEEE standard 802.11b).

Bluetooth

Bluetooth is a wireless low-power, low-cost technology that is designed for short range wireless communication. Specification is developed by Bluetooth Special Interest Group.

It is designed to smooth wireless communication between PDAs, laptops and other similar devices. Bluetooth works in the frequency band of 2.4 GHz. [3]

Even though it has been a long time since Bluetooth was published, devices that support Bluetooth have been appearing quite slowly. However, now there are many devices, and chips have become relatively cheap as originally intended.

Bluetooth is ideal in indoor communications since its range is very short, about 10 meters. It has many advantages in home environment in addition to being wireless. There has been news flashes about high technology homes in which television can communicate with refrigerator. With Bluetooth this becomes quite easy. Every Bluetooth chip can communicate with each other and every chip has its own identity. It is possible for example to use Bluetooth chip as a key that opens door when person approaches his home. When entering, the chip might connect to the lighting system and turn on the lights while refrigerator would upload shopping list to person's PDA. Possibilities are endless. [3]

Bluetooth's close range throughput is 1 Mbps so it doesn't compete with WLAN with its speed [3]. However, Bluetooth can be an optimal solution for devices like printers and MP3 players that don't need large data flow. Actually, Bluetooth can be used to create a so called Personal Area Network (PAN) that transfers data with mobile devices that can be carried all the time.

Wireless LAN

Wireless LAN (name WLAN is used here to refer to the IEEE standard 802.11b) is a communications system that can be used to replace or extend wired LAN. WLAN is actually an IEEE standard which means that different products work (or at least should work) well together.

WLAN operates in the radio frequencies just like Bluetooth but its main advantage over Bluetooth is its fastness. WLANs throughput can be as high as 11 Mbps. It typically needs an access point that connects it to the wired network but it can also be used in "P2P mode" that lets two computers communicate whenever they are in the range of each other. [7, 15]

WLAN that works in "P2P mode" have exactly the same advantages than Bluetooth network but it is much more expensive. However, when large throughput or longer ranges are needed WLAN is superior to Bluetooth. As WLAN cards are quite high-priced, building the same kind of network with them as with Bluetooth chips is way too expensive for most people. It makes no sense to add WLAN cards to refrigerators and devices like that so it is worth thinking a little bit before choosing between WLAN and Bluetooth. Actually, there is no need to choose, because the two technologies are complementary and can be used together to fill anyone's needs at least in home environment. [15]

5 Open Solutions

In addition to proprietary solutions there are some open solutions available. Here we concentrate only in Home Audio Video Interoperability (HAVi) which is the most interesting open solution currently available.

5.1 Home Audio Video Interoperability

Home Audio Video Interoperability is a common, openly-licensable specification for networking digital home devices. The purpose of HAVi is to create a specification that lets any digital device communicate with any other digital device at home. Goal is that a person could operate any device using whichever appliance is nearest. [6]

Because the most frequently used electronic devices in home are entertainment products, HAVi tries to make controlling them very simple. HAVi defines a middleware that manages multi-directional audio-video streams. It also defines an application programmer's interface for developers to create new software and make it easier for third-party companies to create applications for existing devices. [6]

HAVi uses IEEE 1394 (FireWire) as its interconnecion medium. The most important reason for this choice is that IEEE 1394 has enough capacity for several high quality digital audio and video streams. [6]

Using HAVi is very simple. When new devices are installed they register themselves automatically and install needed software. Other devices become conscious of the device at the same moment it is installed. There is no need for many hours of administrative work since much of the intelligence of the system is built into products. [6]

After appliances are installed, they work smoothly together providing services to each other and most importantly they provide services for users through other devices which is something that HAVi is really all about. For example a person can use his VCR or TV through voice recognition system. Some even more science fiction like schemes become reality with HAVi: It is possible to set video cameras outside the house to send their video signal directly to TV whenever motion sensors detect that someone is at the door. [6]

It really looks like HAVi is something technology oriented people have long been waiting for. But does it fulfill all its promises? There hasn't been much conversation of any HAVi compliant devices, but the technology really sounds interesting. Actually, members of HAVi say that "leading suppliers of consumer electronics are already committed to producing HAVi compliant products" so it is possible that products will start appearing quite soon.

6 Summary and Conclusions

It seems that P2P technology is still evolving and all of its capabilities haven't even been found yet. However, there is no doubt that P2P is here to stay and quite likely it will replace client/server model in the near future. As distributed computing environments become more and more common, P2P seems superior to client/server model since in P2P model there is no need for central server and networks can be made self orienting.

The transition to P2P Internet won't be very fast because most of the information providers are still large companies and communities that want to control their information strictly.

Regardless of that, it seems that users are willing to take the step towards the “Content at the Edges” model and P2P will inevitably conquer traditional communication model sooner or later.

P2P communication model seems to fit perfectly to home environments where there are many different devices and it is crucial that they are able to communicate with each other. Using P2P model home networks can orient themselves to the most effective form and P2P networks become much cheaper than centrally managed networks. Interoperability of different digital devices definitely is something that has been waited for and it opens huge amount of possibilities when designing home of the future.

Microsoft has clearly moved towards open solutions by using XML as their main data representation model and by making MSN Messenger an open standard. This can be seen as a very positive trend since most of the people are using Microsoft products and now there is a chance for developers to produce Microsoft compliant applications. The idea behind .NET product family is to use many open standards to make it easier for developers to create communicating applications even with any other operating system than Windows.

The whole .NET product family appears to be quite interesting and it integrates tightly to Windows. There is no doubt that .NET applications will work very nicely in Windows but it is quite likely that there won’t be usable development environments for other operating systems in the nearest future. As Microsoft sets standards like this, it is very hard for others to follow except by designing their own standards. This again leads to a situation where there are many standards and the noble object of interoperating systems without operating system boundaries is not reached.

HAVi looks like the most interesting idea studied in this report. However, for now it seems only like a good idea since there aren’t many HAVi-enabled devices in the market. But as soon as suppliers of electronic devices start to support HAVi (and no doubt they will), there will be quite interesting times in the field of home environments.

The purpose of the study was to present a view of P2P technology and analyze some products that are already in the market. Also future trends of P2P computing especially in home environments were viewed. The basics of the technologies were presented and more information can be found by following the references.

References

[1] ANON. *Modern Peer-to-Peer File-Sharing over the Internet*. 2001. http://www.limewire.com/index.jsp/p2p referenced February 5, 2002.

[2] ANON. *Microsoft Announces HailStorm, a New Set pf XML Web Services Designed to Give Users Greater Control*. 2002. Microsoft press release, http://www.microsoft.com/presspass/features/2001/mar01/03-19hailstorm.asp referenced March 22, 2002.

[3] BLUETOOTH. *Specification of the Bluetooth System*. February 22, 2001. http://www.bluetooth.org/docs/Bluetooth_V11_Core_22Feb01.pdf referenced March 25, 2002.

[4] DORNFEST, R. *Brewing a Hailstorm*. March 23, 2001. http://www.oreillynet.com/pub/a/network/2001/03/23/brewing_a_hailstorm.html referenced March 22, 2002.

[5] FAWCETTE. *Microsoft MSN Messenger: The Ultimate NetMeeting Add-on*. 2000. http://www.devx.com/netmeeting/messenger.asp referenced February 5, 2002.

[6] HAVI. *The HAVi Specification*. 2001. http://www.havi.org/techinfo/docs/release-May15-HAVi1.1(clean).pdf referenced March 26, 2002.

[7] IEEE. *IEEE Std 802.11b*. September 19, 1999. http://a957.g.akamai.net/7/957/3680/v0001/standards.ieee.org/reading/ieee/std/lanman/802.11d-2001.pdf referenced March 25, 2002.

[8] ISENBERG, D. S. *Windows Messenger: New Waves of Innovation*. 2001. http://www.microsoft.com/WindowsXP/pro/techinfo/planning/networking/windowsmessenger.asp referenced February 5, 2002.

[9] MCDOUGALL, P. *The Power Of Peer-To-Peer*. 2000. http://www.informationweek.com/801/peer.htm referenced February 5, 2002.

[10] MICROSOFT. *Messenger - Most Frequently Asked Questions*. 2002. http://messenger.msn.com/support/helphome.asp?client=1 referenced February 5, 2002.

[11] MICROSOFT. *What are XML Web Services?* 2002. http://www.microsoft.com/net/defined/xmlservices.asp referenced March 26, 2002.

[12] MICROSOFT. *Microsoft .NET Passport Technical Overview*. September 2001. http://www.microsoft.com/myservices/passport/technical.doc referenced March 22, 2002.

[13] OLSON, L. *.NET P2P: Writing Peer-to-Peer Networked Apps with the Microsoft .NET Framework*. 2001. http://msdn.microsoft.com/msdnmag/nettop.asp?page=/msdnmag/issues/01/02/netpeers/netpeers.asp&ad=ads.ddj.com/msdnmag/premium.htm referenced February 20, 2002.

[14] ORAM, A. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. 2001.

[15] PAULO, G. *The Connected Couch Potato: Living it Up in the Wireless Home*. 2001. http://www.meshnetworks.com/pdf/ar_cahners_excerpt.pdf?Segment=Residential%20Connectivity referenced March 25, 2002.

[16] RUTHERFORD, E. *The P2P Report*. 2000. http://www.cio.com/research/knowledge/edit/p2p_content.html referenced February 5, 2002.

[17] SLEMKO, M. *Microsoft Passport to Trouble*. October 5, 2001. http://alive.znep.com/m̃arcs/passport referenced March 22, 2002.