HELSINKI UNIVERSITY OF TECHNOLOGY 2.12.2004
Telecommunications Software and Multimedia Laboratory
T-111.590 Research Seminar on Digital Media
Fall 2004: Web Service Technologies

Mobile P2P
–
Creating a mobile file-sharing environment

Johnny Biström 21548C

Ville Partanen 48664F

Mobile P2P – Creating a mobile file-sharing environment

Johnny Biström Ville Partanen
HUT, Telecommunications Software and Multimedia Laboratory

johnny.bistrom@hut.fi
ville.partanen@hut.fi

Abstract

In the recent year's peep to peer (P2P) have evolved rapidly in fixed networks. Especially file-sharing between users is extremely popular. File-sharing applications have not yet beed adopted in mobile networks though the development of hardware would allow it. In this paper the current mobile file-sharing possibilities in a 2G and 2.5G environment are presented. Solutions with and without the help of a fixed network peer are discussed. The paper then looks at the future and predicts what the P2P architecture will be in 3G environment. The open source JXTA solution is used as an example. Finally different threats for mobile P2P are discussed

1 INTRODUCTION

Peer-to-peer (P2P) is a way of communicating between computers without the need of centralized computer or server. Peers, who are computers, are connected to each other and thus users have a possibility to share and search information or chat with each other. In this paper the mobile peer is seen as a mobile phone that has data transfer capability.

In the current 2/2.5G environment the mobile phone cannot have a true end-to-end connection to another mobile terminal, since the network does not support it. In the 3G networks this feature however is supported.

In recent years the media has seen P2P applications just a way of distributing copyrighted material. P2P Software such as Kazaa and Napster are mostly seen as a way of doing something illegal, all though the potential behind P2P is much more extensive and gives great benefits to the user. In cellular networks the billing is a problem for the P2P-applications as well. The network operators can charge as they desire for events in the network. It is very common that the user has to pay for every down- or uploaded MB. This leads to an intresting paradox. If your mobile phone is a peer and you share information on it someone wants to download, you would have to pay for the other parts download. Are users willing to do that?

The case which will be studied here is mobile file-sharing because it is the most wanted and most used feature in fixed networks. It has all the complexity needed to be representative for the P2P environment. If file-sharing can be realized, most the other features of P2P are possible to realize.

In this paper the current possibilities for file-sharing in a mobile 2/2.5G environment will be studied (chapter 2). A look at the future is also done with the help of the JXTA architecture and protocols (chapter 3) and then current and future obstacles in P2P networking will be presented (chapter 4). Finally some conclusions concerning this topic are drawn (chapter 5).

In this paper our research questions are:

1. How can file-sharing be realized in mobile networks today?

2. How will the upcoming 3G-networks support file-sharing?

3. What are the threats for file-sharing and how can they be overcomed?

2 MOBILE FILE SHARING IN THE CURRENT 2/2.5G ENVIRONMENT

The most demanding obstacle in 2/2.5G environment is the network limitations: currently operators do not allow mobile terminals to act as independent terminals with an IP address. Thus when transferring data from Internet to mobile phone, from the Internet side the tunnel seems to end at the GGSN, Gateway GPRS (Mishra, 2001).

After GGSN the operator will route data to the mobile phone according to it's own specifications. For example here in Finland the two major operators (Sonera and Elisa) informed that applications are allowed one by one to transfer data to and from the mobile phone. This means that operator control all of the TCP/IP ports and because of this it is only possible to send data through ports, which are always open. (Based on a phone interview with Elisas and Soneras networks configuration managers and various end user experiences around the world).

Addition to this there is also the problem that the GGSN might change during the data transfer because of mobile terminal movement. This causes the change of mobile terminals IP address. Therefore even if we could find out what is the mobile phones IP address at one place we would not be able to sustain active connection during movement since the IP address would change. The only possible way to sustain active connection would be to implement this feature to the network for example with Session Initiation Protocol (Handley, 1999), which creates, modifies and terminates connections between users. With this protocol the operator would always know what is the current location (IP address) of user and therefore active connection could be maintained.

In 2/2.5G environment we also have the restriction that data and speech can not be send simultaneously unless mobile phone belongs to class A or operator adds new features to the network. The first option, class A mobile phone, would demand lots of

CPU and because of this there are no class A mobile phones. The second option, network assistance, would demand investments in the network and currently it seems that there are no operators that are willing to invest to this technology (Pecen, 2001).

Due to these limitation we have to find away of transferring data in 2/2.5G environment without IP addresses and GPRS data. In the next chapter we present a solution for file sharing by using MMS.

2.1 Using MMS for file-sharing

In GSM networks operators and content providers have been waiting for rapid growth of MMS messaging for several years. In MMS messaging content is distributed in a same way as in SMS messaging: a user will sent a message that contains various information for example pictures, voice and text and the recipient is recognized by using ISISDN number or an email address. MMS messages size is not restricted to any size, but due to the limitations of mobile phone memory and capacity one message maximum size could be for example 100 kbytes. Thus this depends on the mobile phones manufacturer. However, using software that could slice large file into smaller ones could revolve this limitation.

To use MMS for file-sharing in the mobile network is one way of realizing P2P-functionality. The size is not a limitation if you would like to send audio as a ring-tone or pictures as logos or icons to your friend. Often however, the sending of a ring-tone, logo or icon is limited by the phone. It is not because of the technology but it is done in order to prevent the operators' economical interests in selling ring-tones, logos and icons. This makes it difficult or even impossible to share files of interest in the mobile network.

Antother problem is the search functionality. In 2/2.5G networks it is impossible to send out a search request as in Internet based P2P-networks. This limits the possibility of mobile file-sharing. There is a solution for this problem if an Internet based server would be used as a contact srver. The contact server would allow users that are interested in sharing files to register and store their phone number, phone type and a list of files available for sharing on their mobile phone on the server. The contact server could store this information in a database, which could be used for searches by other registered file-sharers. The interface to the database could be realized as a WAP- or a XHTML-based search page. If a search in the sharing database would get a hit, the phone number to the sharer would be given and the file could be requested by sending a MMS-request for the file to the sharing phone. If the sharer is interested in sharing the file, he would send it to the requester. This can even be made fully automatic, so the sharer does not have to interact at all by installing a small program that sends the requested file on the mobile peer. Figure 1.

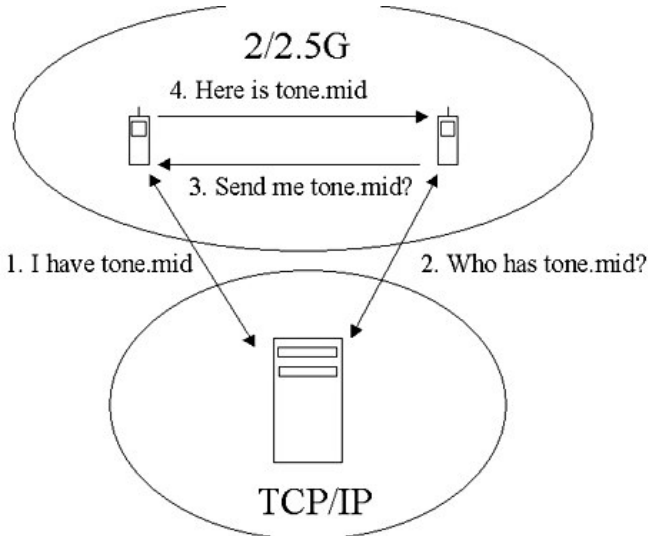

Figure 1: MMS file search and share

2.2 Mobile phone connects to a peer-computer in the fixed network

Another way of solving the sharing problem in the mobile network would be that every mobile terminal that is interested in file-sharing would have a private peer-computer connected to a fixed environment? In this kind of architecture the mobile terminal would act more as an end peer that is connected to a peer in the fixed network. With this kind of architecture we could overcome most of the limitations of mobile phones:

- Two-way communication – We can share the information we like from the computer and if we would like to download or search files we could do that with the help of the computer
- Power – We do not have to waist power for insufficient search because the computer can do this. When the computer has search information we can transfer the only the search results to the mobile phone.
- Bandwidth – In a fixed network bandwidth is no problem. Everyone could download files from the. Addition to this files would be downloadable all the time compared to 2/2.5G environment where files can be transferred only when there is no circuit data.
- Efficiency – Computer has enough power to handle all kind of messaging, transferring data and searching information.

The software needed is available today as the Internet peer would run any P2P-software available that already has the search and file-sharing functionality. The mobile

phone connects to the peer with the communication software that comes with the phone (e.g. PC Suite or corresponding) or directly via the infrared interface. Many users already import their ring-tones from common midi-archieves today instead of buying them from the operator. By sharing files made for mobile phones in your normal P2P-environment the advantages brought by P2P are accessible also in your mobile phone.

This solution gives you access to any P2P-network and all the files available there but the transfers to your phone are not automatic and you have to be in range of your computer. The searches also have to be made from your P2P-software. Figure 2.

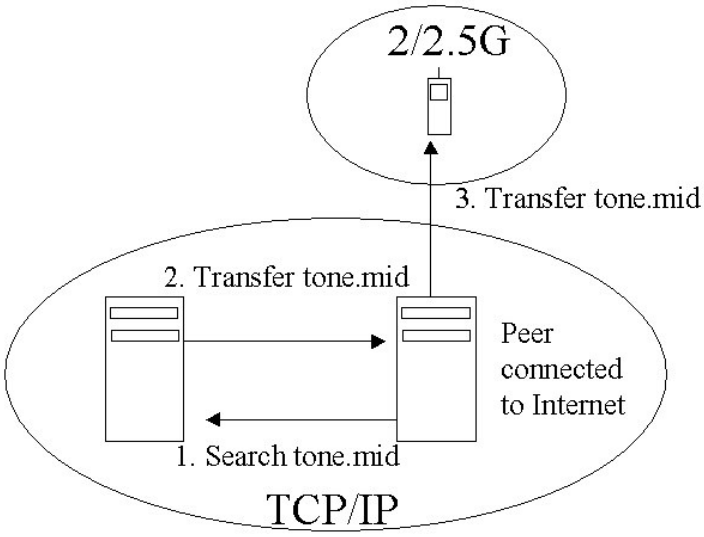

Figure 2: Computer aided P2P

2.3 EMule: mobile phone as a remote control for peer

The eMule development, which started as a SourceForge-project (SourceForge.net. Project: eMule, 2004), has developed yet another solution to mobile P2P today. eMule is a well known open source program for P2P networking (eMule Project, eMule Home Page, 2004). The Mobile eMule is a subproject of eMule and its goal is to provide a way of controlling the eMule peer on the computer with a mobile phone (eMule Project, Mobile eMule Home Page, 2004).

Mobile eMule is thus not a P2P-client for your mobile phone but a remote control for your eMule-client on the computer (Figure 3). Basically all java enabled mobile phones which have 40 KB free fixed memory and about 150-200 kB heap are supported. A color display is recommended. Mobile eMule connects, through the telecommunication network, as a client to your eMule server in the computer through port 80. With the mobile eMule client you can control the eMule server on your PC. You can do searches, downloads, check download status and check file properties.

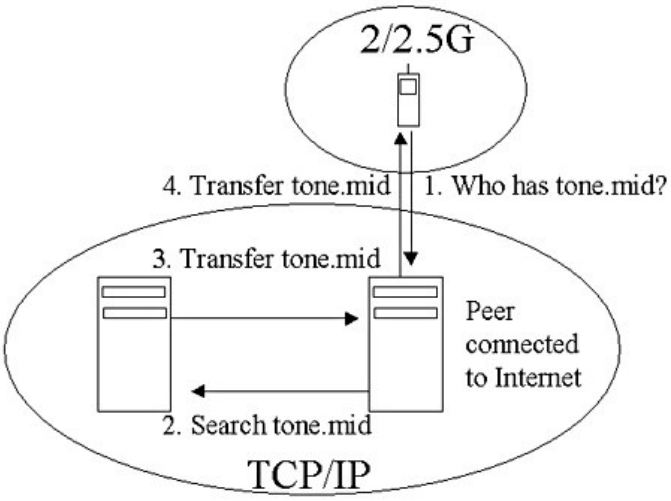

Figure 3: Computer aided P2P with GRPS connection between computer and mobile phone

To use Mobile eMule you have to download a small Java program to your phone. Once downloaded you start the session from your Mobile eMule client by logging in to your eMule server as shown in figure 4

.

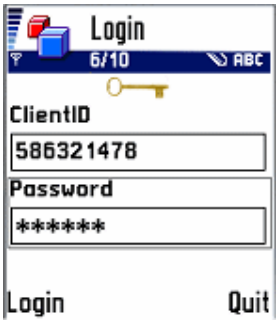

Figure 4. Logging in to Mobile eMule.

Then you can start a file search and get the results as shown i figure 5. Type the keyword you want to search for into the field (in the first screen), select a file type and then click start. MobileMule will show you up the 15 results which have the most sources available. The number on the right will you you how many sources this file has. However it has max. 2 digits, meaning if there are more than 99 sources it will show 99.

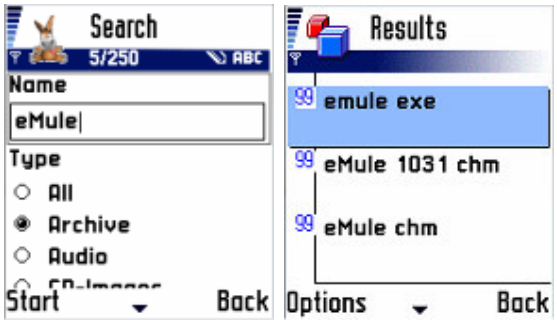

Figure 5. Searching for files with Mobile eMule.

When you are downloading you can check the status of your downloads as shown in figure 6. On this screen you will see all current downloads and their status. Two red rectangles means this file is paused, a green circle indicates that this file has the waiting state while a filled green circle is a file which is currently receiving data.

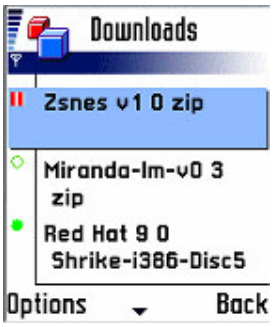

Figure 6. Download status shown by Mobile eMule

The eMule solution has the advantage that you have full P2P functionality available in the mobile phone though you might be far away from your computer. You cannot download the files directly to your computer today but as Mobile eMule works through a web interface to eMule future versions will allow downloading of files to the mobile. This will be an excellent solution for mobile P2P in EGPRS-networks.

3 JXTA – THE SOLUTION FOR P2P NETWORKS OF TOMORROW

JXTA is aimed to be the base platform of the P2P networks of tomorrow. It is an architecture description and a set of protocols that enables the programming of advanced P2P applications. It is open-source and it is built to be device and platform independent. JXTA has been tested on fixed networks but the work on mobile networks is still not finished.

3.1 Background

The JXTA project started as a research project incubated at Sun Microsystems in 2001 to address the P2P space. JXTA is pronounced "juxta" and it is an abbreviation of the word "juxtapose", which means to put things next to each other. The JXTA technology is a set of open generalized P2P protocols that allow any connected device, from mobile phone to server, to communicate and collaborate. JXTA peers create a virtual, ad-hoc network on top of existing networks. JXTA uses proven web technologies as HTTP, TCP/IP and XML. Sun introduced the JXTA technology with an open-source, royalty-free license model and today 16000 registered members of the community contribute to the development process (Sun Microsystems 2001. Project JXTA: An Open, Innovative Collaboration).

3.2 Software Architecture

Any P2P software, and so also JXTA, can generally be broken down to three layers; the core layer, the services layer and the application layer as seen in figure 7. The core layer deals with peer establishment, communication management, such as routing and other low level tasks. The services layer deals with higher-level concepts such as indexing, searching and file sharing. At the top is the applications layer which includes applications for emailing, auctioning and storage systems.

JXTA was designed to provide a general platform on top of which services and applications could be built. This platform was intended to be thin and small, yet providing interesting and powerful primitives for use by the services and applications (Gong. 2002).

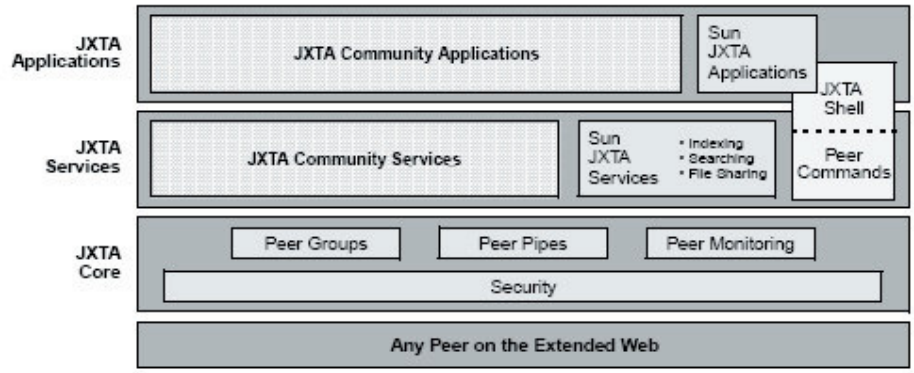

Figure 7. JXTA Software Architecure.

3.3 Technology

The addressing model of JXTA is based on a uniform and location independent logical addressing model. Every network resource is assigned a unique peer ID independently of its physical address. JXTA uses a universal resource binding mechanism called the resolver to perform all resolution operations found in a traditional distributed system, such as resolving a peer name to an IP address. All resolution operations are unified under the simple discovery of one or more advertisements. The default resolver policy is based on Rendezvous super-peers. Rendezvous are peers that have agreed to cache advertisement indices which are pointers to edge peers that cache the corresponding advertisement. Rendezvous super-peers organize into loosely-coupled networks where each rendezvous maintains its own rendezvous peer view which is an ordered list of other known rendezvous in the peer group as seen in figure 8 (Traversat et al., 2003).

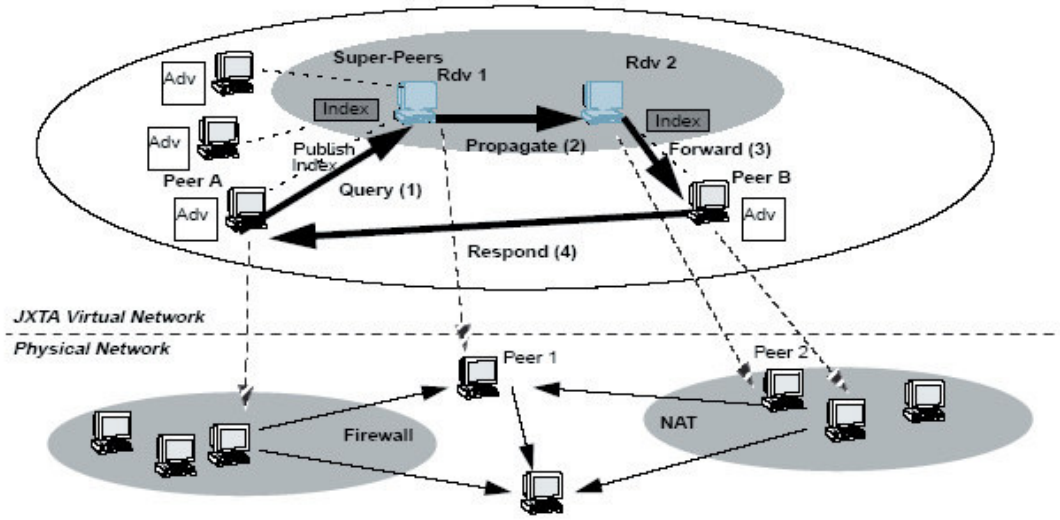

Figure 8. The Architecture of the JXTA network.

3.4 Protocols

The six JXTA protocols can be divided into *Core Specification Protocols* and *Standard Service Protocols.* The Core Specification Protocols are:

• The *Endpoint Routing Protocol* (ERP) can discover a route through an intermediary peer if a direct route can not be found between two peers.

• The *Peer Resolver Protocol* (PRP) is the protocol for sending generic resolver queries to another peer and receiving responces.

The Standard Service Protocols are:

• The *Rendezvous Protocol* (RVP) makes it possible to subscribe for a propagation service. RVP is used by PRP in order to propagate messages.

• The *Peer Discovery Protocol* (PDP) is the protocol by which a peer publishes its advertisements and discovers other advertisements. PDP uses the PRP for sending and receiving advertisements requests.

• The *Peer Information Protocol* (PIP) makes it possible to obtain information about other peers. PIP uses PRP for sending and receiving information requests.

• The Pipe Binding Protocol (PBP) makes it possible to establish a pipe between one or more peers. PBP uses PRP for sending and receiving pipe requests.

More information on the protocols can be found from the specification (Sun Microsystems, 2004).

3.5 Realized JXTA Example Applications

The **JXTA Shell** application permits interactive access to the JXTA platform via a simple command line interface. The Shell enables access to the core building blocks: *peers, peer groups, pipes* and *codats*. Codats are units of contents that can hold both code and data. Codats are the smallest units of contents manipulated by JXTA. Using the Shell, a user can interact with the JXTA platform to publish, search, and execute codats, discover new peers or peergroups, create pipes to connect two peers, and send and receive messages.

Every JXTA Shell command has the following syntax

```
command [< pipe] [ >pipe] options arguments ;
```

Some example commands are peers, groups, search, ls, env and cat which all lists different elements. To create elements mkadv, mkpgrp, mkpipe and mkmsg are used. To transfer information commands as send, recv, put and get are used (Sun Microsystems 2001. Project JXTA: Technical Shell Overview)

MyJXTA is an open source exemplary application that has the following features: group chat, secure one to one chat, open and credentialed groups, sharing of files, drag-and-drop content publishing, pipe search and messaging interfaces (Sun Microsystems. MyJXTA2 Project Home Page, 2004). A screenshot is shown in figure 9.

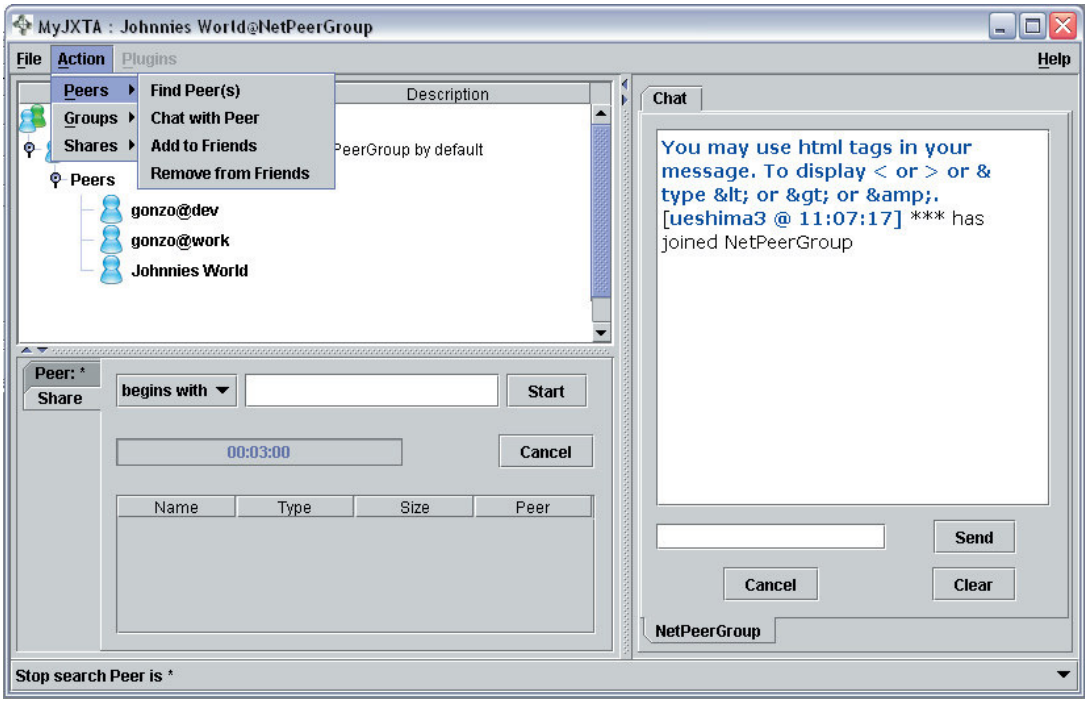

Figure 9. Screenshot of MyJXTA2.

3.6 4.6 JXME – The mobile version of JXTA

JXME is a development project that aims to adopt JXTA also for J2ME. The purpose of project is to provide JXTA compatible functionality's on contained devices using the Connected Limited Device Configuration (CLDC) and the Mobile Information Device Profile 2.0 (MIDP). The range of devices includes the smart phones to PDAs. Using JXTA for J2ME, any MIDPdevice is able to participate in P2P activities with other MIDP devices. At the same time, a MIDP device is able to participate, with some restrictions, in P2P activities with JXTA peers running on desktops/workstations/servers (Sun Microsystems. JXTA for J2ME Project Home Page, 2004).

Some time ago it was stated that because of platform resource constraints, and the corresponding limitations in MIDP-1.0, JXTA for J2ME peers can only act as edge peers. That is, they cannot assume the role of more sophisticated peers that offer services to other members in the peer group. From a purely practical point of view, wireless peers must principally act as windows into the network for their users. As a result, the heavier lifting (searching for resources, performing computationally-intensive tasks, etc.) must be done by other members of the peer group. Indeed, even some of the basic tasks associated with peer group membership must be assumed by other peers called JXTA relays which is described in figure 10 (Arora et al., 2002).

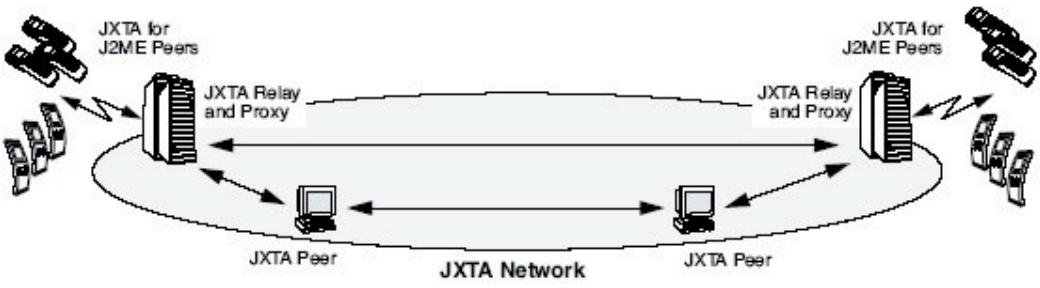

Figure 10. The wireless proxy based JXTA Architecture.

Today a proxyless version is considered possible in the near future and the features supported in the first release of proxyless JXME will be:

1. **The discovery of**

• **Pipes**. An application is able to search for a named pipes created by other Peers.
• **Groups**. An application is able to discover a JXTA group and join it.
• **Contents**. Applications are able to discover application specific contents.

2. **The creation of**

• **Pipes**. An application is able to create pipes - both P2P and propagate pipes.
• **Groups**. An application is able to create peer groups to limit the scope of discovery.
• **Contents**. Application specific contents.

3. **The joining of groups**. An application is able to join a given group.

4. **Possibility to communicate**. An application is able to communicate with other JXTA users thru JXTA pipes.

Listed below are a few of the critical constraints imposed by cell phones and like devices. JXTA for J2ME is designed to provide P2P functionality within these limitations. Next generation devices may have a lesser constraints than the one currently-available.

1. 50k MIDlet size Current cell phones have a total limit of about 123K for storing all MIDlet suites. In addition, Motorola phones currently limit each MIDlet to be no more than 50K and NTT DoCoMo phones limit MIDlets to 30K.
2. Persistent storage on cell phones can be as little as 8K which is shared by all the MIDlets.
3. Runtime heap is of the order of 32K - 64K
4. Bandwidth is very limited and latency is high
5. CPU power is very limited - around 20MHz.

6. Battery life is very critical.
7. Limited libraries MIDP-2.0 has
   - No XML parser
   - Optional support for Sockets and ServerSockets.

The proxyless JXME design concept is described in the figure 11 below.

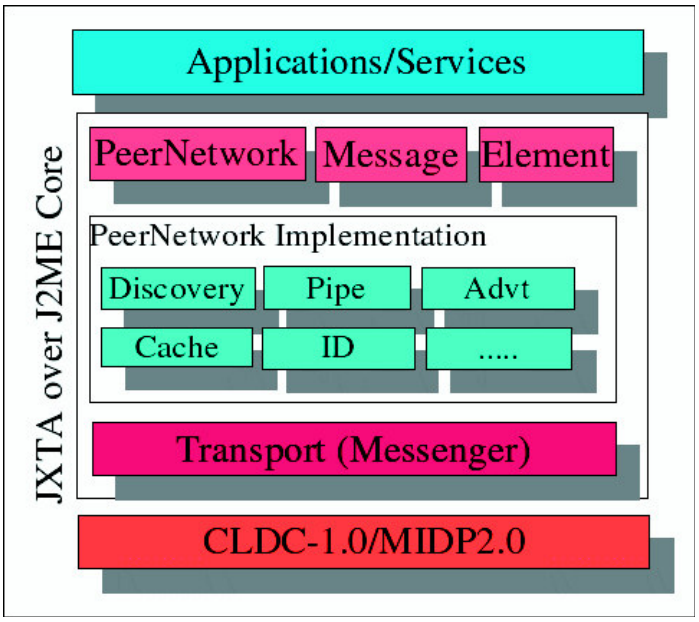

Figure 11. The proxyless JXME design concept.

3.7 Current Status of Research Work on JXME

At the moment the goals and functionality desired can not be achieved. A number of technical problems remain. There is a plan to refine the JXME 2.0 towards the proxyless approach but so far the work has not really begun. JXME must therefore be considered as a possibility of the future and proxyless mobile JXTA P2P networks cannot yet be built.

4 THREATS TO MOBILE P2P

The 3G mobile network enables true IP-based communication for mobile phones on the telecommunication network which means that a phone will have an IP-address of its own. If the architecture and protocols of JXTA are used, file-sharing applications and full P2P functionality will be available on mobile phones. This would lead to a reduction in operators revenues on SMS- and MMS-services as mobile phone users instead could chat and exchanges multimedia files through P2P. Also the revenues on ring-tones, logos and icons would be reduced as people would exchange these through file-sharing. It is not very likely that the operators will allow this if they can prevent it. It can be expected that some kind of restrictions on the network that prevent P2P functionality will be introduced.

On the other hand the amount of transferred data would increase and bring new revenues to the operators. If the lost revenues will be compensated by higher transfer rates the price of sharing might be as expensive as buying the files from the operator.

Another threat for P2P on mobile networks is the reduced security and the possibilities to transfer viruses, spy- and adware to the mobile phone. In order to prevent this, the phone manufacturers or operators my introduce limitations in the possibility of downloading and installing software on the mobile phone. Some kind of virus protection and firewalls will be necessary on the mobile phone in the future.

Will the interest to download files from the network remain as big as it is today and will the mobile phone ever offer the same functionality as a computer connected to the Internet. Is it not easier to download to the computer and then transfer the files to the mobile phone?

Digital Rights Management Systems are growing and in the future DRM will certainly include music and videos on the Internet. How far will the rights management go? Will pictures, icons, logos and ring-tones also be included. If so, this will greatly reduce the interest in file-sharing in the future. On the other hand we can not underestimate the end users productivity: Currently there are Internet pages where you can download themes, ringtones and images made by end users to your mobile phone. It is very likely that these kind of files will be always distributed for free and thus the need for P2P networking will be immense.

File-sharers have been prosecuted recently and the legislation considering P2P is being looked over in many countries. Will file-sharing be considered a crime for which you can be punished so heavily in the future that the popularity will be reduced or will file-sharing prevail and live on?

5 CONCLUSIONS

P2P networking offers new functionality to the mobile phone and especially interesting is the file-sharing possibility but currently there are difficulties that prohibit true mobile P2P. In 2/2.5 environment mobile P2P has obstacles that are nearly impossible to overcome. Speech and data can not be transferred at the same time; when using MMS aided mobile P2P file sharing we had several restrictions such as 100 kb file size; cost issues since the person who is uploaded has to pay and finally we have phone limitations.Also in 2/2.5G mobile terminal can not be identified with IP address when transferring data from the Internet because of the operator limitations.

Because of the these prerequisites we can come up to the conclusions that in 2/2.5G environment, without the help of computer, mobile P2P can not be implemented in such a way that it would attract users. However an architecture where mobile terminal is connected to a computer, which has a P2P software seems a promising solution. With this kind of architecture user has the advantages of fixed network (efficient search, bandwidth, high CPU and low cost) and the mobility of mobile device. Connection between these two can be done for example over http protocol and therefore the operator can not restrict the connection. Thus we believe that in current networks mobile P2P

applications will be dependent of fixed network computers. As an example of this service the NewBay software will begin their mobile P2P service in the beginning of 2005.

In 3G environment there are not technical obstacles in mobile P2P.We believe that operators, who can control data traffic, will mainly dictate the future in this area. At this paper we presented one possible architecture, JXTA, that would be suitable for G3 environment. To which extent P2P-networks in the future will exist on the mobile platform is more a commercial, security related, DRM and legislation question.

REFERENCES

Arora, A., Haywood, C. and Pabla, K.S. 2002. JXTA for J2ME – Extending the Reach of Wireless with JXTA Technology. Sun Microsystems, Inc. Palo Alto, CA, USA, March 2002. Available: http://www.jxta.org/project/www/docs/JXTA4J2ME.pdf

eMule Project. eMule Home Page. Developers Page (Online) 2004. [Referenced 29.11.2004]. Available: http://www.emule-project.net/

eMule Project. Mobile eMule Home Page. Developers Page (Online) 2004. [Referenced 29.11.2004]. Available: http://mobil.emule-project.net/

Gong, L. 2002. Project JXTA: A Technology overview. Sun Microsystems, Inc. Palo Alto, CA, USA. October 29, 2002. Available: http://www.jxta.org/project/www/docs/jxtaview_01nov02.pdf

Handley M; Schulzrinne H; Schooler E; Rosenberg J. 1999. SIP: Session Initiation Protocol. RFC 2534. Internet Engineering Task Force.

Mishra A. 2001. Performance and architecture of SGSN and GGSN of general packet radio service (GPRS). Global Telecommunications Conference, GLOBECOM '01. IEEE, **6**, 3494–3498.

Newbay.com. Newbay Home Page. Developers Page (Online) 2004. [Referenced 1.12.2004]. Available: http://www.newbay.com/index.php.

Pecen M.; Howell A. 2001. Simultaneous voice and data operation for GPRS/EDGE: class A dual transfer mode. Personal Communications, IEEE, **8**, 14–29.

PerPhone 2004. The PerPhone Project. Developers Page (Online). [Referenced 1.12.2004]. Available: http://www.nokia.com/nokia/0,,5169,00.html.

SourceForge.net. Project: eMule. Developers Page (Online) 2004. [Referenced 29.11.2004]. Available: http://sourceforge.net/projects/emule/

Sun Microsystems 2001. Project JXTA: An Open, Innovative Collaboration. Draft 1.0. Sun Microsystems, Inc. Palo Alto, CA, USA. April 25, 2001. Available: http://www.jxta.org/project/www/docs/OpenInnovative.pdf

Sun Microsystems 2001. Project JXTA: Technical Shell Overview. Draft 1.0. Sun Microsystems, Inc. Palo Alto, CA, USA. April 25, 2001. Available: http://download.jxta.org/files/documents/27/21/JXTA-Shell-Overview.PDF

Sun Microsystems 2004. JXTA v2.0 Protocols Specification. Sun Microsystems, Inc. Palo Alto, CA, USA. Available: http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.pdf

Sun Microsystems. MyJXTA2 Project Home Page. Developers Page (Online) 2004. [Referenced 29.11.2004]. Available: http://myjxta2.jxta.org/

Sun Microsystems. JXTA for J2ME Project Home Page, Developers Page (Online) 2004. [Referenced 29.11.2004]. Available: http://jxme.jxta.org/

Traversat, B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J-C., Pouyoul, E. and Yeager, B. 2003. Project JXTA 2.0 Super-Peer Virtual Network. Project JXTA. Sun Microsystems, Inc. 25 May 2003. Available: http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf