Mobile P2P: Turning Heterogeneity to an Advantage

Andreas Berl, Ivan Dedinski, Emanuel Georgiew and Hermann de Meer
University of Passau
Faculty of Computer Science and Mathematics
94032 Passau, Germany
{berl, dedinski, georgiew, demeer}@fim.uni-passau.de

Acknowledgement: This project was partly funded by the German Research Foundation (Deutsche Forschungsgemeinschaft - DFG), contract number ME 1703/4-1, by EPSRC, contract number GR/S69009/01, and by the EURO-FGI - Network of Excellence, European Commission grant IST 028022.

Abstract

Peer-to-peer content-distribution networks have a large user community in the fixed Internet today and cause a noticeable part of Internet traffic. Enabling popular peer-to-peer applications on mobile devices in cellular networks is an interesting opportunity for both, customers and operators. However, when mobile devices join peer-to-peer networks, the networks become heterogeneous (e.g. differing link capacities, CPU power, etc.), leading to disadvantages for all peers in the network. Consequently, mobile devices need some kind of support to be integrated properly in content distribution networks. In this paper, the heterogeneity in peer-to-peer networks is not only considered as disadvantage but also as new opportunity. Mobile devices are identified as providers of advanced mobile features and services, being usually not available in the fixed Internet. Instead of considering mobile devices as bottlenecks, they are regarded as valuable partners in content distribution networks. Mobile features and services are made accessible to fixed peers, facilitating the integration of mobile devices into popular peer-to-peer networks.

Keywords: Mobile peer-to-peer, heterogeneity, cellular networks, content distribution, incentives.

1 Introduction

Peer-to-peer (P2P) content distribution applications are extremely popular in the Internet today, and might become as popular on *mobile devices* (MDs) in the future. Operators of cellular networks are still searching for ”killer applications” to utilize their newly established infrastructure. Integration of mobile devices into existing P2P networks (with a large user community in the fixed Internet) is an important, but difficult topic of the *mobile peer-to-peer* (mobile P2P) research field.

Current MDs often have enough performance to run (adapted) P2P software. However, their participation in P2P networks together with peers from the fixed Internet causes problems. P2P systems apply logical structures on the underlying physical network, the *overlay*. In the overlay, peers and links are considered to be homogeneous with approximately similar properties and capabilities. MDs in contrast, impose *physical heterogeneity* and *heterogeneity of user behavior* within P2P networks.

Heterogeneity in P2P Networks: Due to their mobile nature, MDs have special properties and differ extensively from stationary computers with fixed links. In comparison, the CPU power of MDs is much lower. Programming languages and execution environments, e.g. JAVA or J2ME do not provide the same feature set on MDs. MDs have less storage space, less random access memory, and small displays. Additionally, MDs are not able to manage many TCP connections simultaneously. Most currently available MDs are only able to maintain 3 to 20 of these connections [1], limited by memory and CPU power. One of the most serious differences is however, that MDs depend on small battery life times, limiting time and intensity of the MDs usage. TCP connections, for instance, tend to be battery consuming. In [13] it is described that periodic keep-alive messages of a single open connection use up a batteries energy within few hours. Apart from the MDs constraints, the network access of MDs via air interface differs highly from fixed links. It is usually of variable quality and slower in orders of magnitude. It depends, for instance, on the users movement, the number of concurrent users served from a base station, or the distance of the user to the base station. Uplinks of an air interface have usually less capacity than downlinks. Moreover, there are dead spots where the air link breaks down completely, possibly leading to a change of the MDs IP address. In addition to the physical heterogeneity, also heterogeneity of user behavior emerges when P2P networks are joined by MDs. Users of fixed Internet peers often prefer to be ”always on”, commonly having Internet flat rates. In contrast, users of MDs are expected to prefer remaining offline, due to the MDs limitations. Battery life time has to be saved to keep the MDs operational and flat rates are usually not available or quite expensive.

Effects of Heterogeneity: Heterogeneity in P2P networks leads to disadvantages for both, fixed Internet peers and MDs. In P2P systems, messages are exchanged cooperatively to keep the overlay in a consistent state. Overlays are used to lookup (and route to) either certain peers or content. Both, maintenance and lookup messages are delayed by the varying (and often low) bandwidth of MDs. Moreover, messages get lost when MDs suddenly go offline or have overloaded CPUs, being unable to process the routing. These events lead to temporary inconsistencies within the overlay, causing increased maintenance traffic and repeated lookup messages. All peers are affected by the decreased performance of the P2P network. Fixed peers experience additional disadvantages, when trying to get content from MDs. Most of a download might have finished, long before the last piece of it is transmitted by an MD with slow upload performance [21]. However, especially MDs experience severe disadvantages in heterogeneous P2P content distribution networks. MDs find themselves in severe competitive situations considering content, when joining P2P networks. A peer that shares content, distributes its upload bandwidth among the peers requesting content. A certain amount of peers is served simultaneously, the others have to wait in queues. If the providing peer is popular (or provides popular content), often thousands of peers are waiting in its queues, competing with each other for content. MDs however, do not perform well in these competitions for several reasons. First, in queues often those peers are preferred that provide content in return (tit-for-tat concept). It is difficult for MDs to provide same amounts of content as fixed computers, due to their hardware limitations, restrictions of the air interface, and short online times. Second, MDs are not able to wait in too many queues simultaneously, due to the above described limited ability of managing concurrent TCP connections and limited memory. Fixed peers, in contrast are able to wait in up to hundreds of queues simultaneously to increase their probability of getting content. Third, MDs are often hidden behind firewalls, depending on their mobile operator. Other peers are not able to establish direct connections to these MDs from outside, complicating the P2P communication. Forth, if an MD goes voluntarily or involuntarily offline, e.g. because of a dead spot or low battery power, it is deleted from all queues and has to start its waiting period again. Therefore, MDs are usually waiting in queues for much longer time periods than fixed peers do (cf. Section 4), being heavily discriminated compared to fixed Internet peers.

Several mobile P2P solutions have been suggested in the past to solve the described problems (cf. Section 5). However, MDs are still not widely integrated into popular P2P content distribution networks. MDs are considered to be weak peers that have to be supported without getting anything in return. P2P systems however, are based on balanced cooperative operation, conflicting with this one-sided support. This imbalance of cooperation has not been solved yet. In this paper a solution is suggested, enabling MDs to outweigh the situation. Section 2 identifies MDs as providers of advanced mobile features and services that can be used to compensate support in P2P networks. Section 3 suggests a mobile P2P architecture, making mobile services accessible in P2P networks. Section 4 evaluates the suggested architecture. In Section 5 mobile P2P solutions concerning cellular networks are addressed. Section 6 concludes this paper.

2 Mobile Features and Services

Current mobile P2P solutions (cf. Section 5) tend to underestimate MDs, judging them by their restricted capacities. MDs are considered to be ordinary computers with limited abilities, needing support in P2P networks. Instead of only focusing on quantitative heterogeneity (e.g. bandwidth or CPU power), this paper also considers qualitative heterogeneity (e.g. mobility, features, and services) in P2P networks consisting of fixed and mobile devices.

In recent years, great technological advances in the area of mobile communication have been done. Most of the current MDs (e.g. phones or personal digital assistants) are able to run complex JAVA programs, play music, or show small videos. Some are able to receive TV programs or radio stations. They have high resolution color displays, integrated video cameras, and advanced audio systems, capable of playing sound in good quality. In addition, a number of wireless interfaces is available, involving WLAN, Bluetooth, or infrared. MDs have access to the Internet and are able to communicate via TCP/UDP connections. Some advanced MDs have further features, e.g. GPS or thermal sensors. However, not only the performance and features of MDs have improved over time, also the number of available services has increased, additionally offering new opportunities. Besides the common telephone service, MDs are able to send SMS messages or MMS messages as well as facsimiles or e-mail. Due to a unique ID, MDs are reliably authenticated by their operator, enabling further services. Payment/micropayment can be done, for instance, by calling special service numbers [12].

Most of the MDs features and services are not (or barely) available to computers in the fixed Internet. However, current MDs provide the technical preconditions to share their special features, enabling a variety of *partnership schemes*. On behalf of fixed Internet computers, MDs can send SMS messages, MMS messages or facsimiles, display advertisements on the MD, do micropayments, take pictures from the surroundings, or gather weather information, for instance.

As an example, two partnership schemes are described in detail. In a first example, MDs provide an advertisement service. As compensation for support in a P2P network, MDs show advertisements on their display. Advertisements consist of one or more pictures, banners, or small videos. Pictures, for instance, are displayed during content download. For every transmitted megabyte of data a certain number of advertisements is shown, until the download completes. Companies, which are interested in pushing advertisements to customers of cellular networks, are able to host supporting servers in P2P networks. Another possibility for them is, to let arbitrary peers push advertisements to MDs. Their effort can be compensated by using Google's "pay-per-click" mechanism [2]. A second partnership scheme example is the provision of SMS messages to fixed Internet peers as compensation for P2P support. Here, MDs "pay" for content download with a predefined number of SMS messages per megabyte of data. During the download, the fixed peers transmit text and phone numbers to MDs, which combine them to SMS messages. This kind of SMS delivery service is especially interesting, when no instant delivery is required, e.g. for advertisement SMS messages. In this scheme, users of the fixed network are able to send SMS messages all over the world, if they find MDs providing this service. Current P2P content distribution applications provide advanced lookup mechanisms that can easily be used to match requester and provider of a certain mobile service.

Many other partnership schemes are possible. However, the focus of this paper is not on inventing partnership schemes or evaluating them, but on providing the technology to enable them. The mobile P2P architecture, presented in the next section is a feasibility study, making mobile features and services available in P2P content distribution networks.

3 Architecture

In this Section a *mobile P2P architecture* (MPA) is suggested, which exploits the heterogeneity in P2P networks consisting of mobile and fixed devices. Stationary devices support MDs in performing P2P tasks and MDs offer mobile services in return.

MPA's Components and Setup: The suggested MPA extends existing P2P content distribution networks. Unmodified stationary peers of these networks are called *fixed ordinary peers* (FOPs). If a FOP intends to consume mobile services, it has to install software with extended properties. This extended FOP is called *service peer* (SP) and supports MDs to get mobile services in return. Solely SPs enhance their communication protocols, other FOPs remain unchanged and are unaware of SPs and MDs in the network. Except of their extensions, SPs remain to be FOPs participating in the P2P network as usual. SPs use client-server based protocols to communicate with MDs. When an MD is connected to an SP it does not communicate with FOPs anymore, being separated from the P2P network. MDs are relieved from network maintenance, routing tasks and the competition for content (cf. Section 1). All these operations are transferred to SPs. Via the client-server based communication protocol, MDs provide mobile services to SPs. Figure 1 illustrates the suggested architecture. It

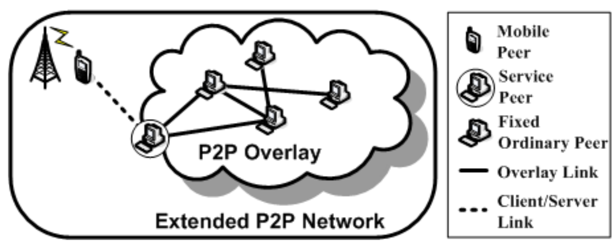

Figure 1: MPA: An extended P2P network involving FOPs, SP and MD

can be observed that the P2P network itself is a component of the MPA. MD's however, are not fully integrated in typical P2P manner. They do neither communicate with FOPs nor with other MDs, after finding an SP. This prevents the effects of heterogeneity, discussed in Section 1. The MPA is easy to establish within an existing P2P network. Installing a single SP instantly enables the participation of MDs in the network, the number of supported MDs is depending on the SPs performance. No further infrastructure or effort has to be done. Additionally, MDs are enabled to participate in more than one P2P network using the suggested MPA. This is possible, because SPs are not restricted to be part of a single P2P network and MDs are not restricted to a single SP. Depending on its performance, an SP is able to participate in several networks simultaneously, filtering and aggregating available resources for its supported MDs. In the following paragraphs the main operation of the MPA is described in detail.

Matching MD and SP: Every MD has to find an SP to participate in the MPA. This problem is shifted to the P2P network itself, using the publish-subscriber principle. SPs publish their availability within the P2P network, by sharing usual content. In a file-sharing network e.g., an SP publishes (shares) a file with a specified name. The file name contains preferences of the SP in its name (consumption of mobile services). The filename ”SP_SMS_Germany.MPA” expresses, for instance, that an SP wants to send SMS messages within Germany. MDs have to perform three steps to connect to the MPA: First, MD's bootstrap like usual peers in the P2P network. Second, they look up files published by SPs, matching their own preferences (offering of mobile services). The P2P network delivers IPs of peers that publish these files (SPs) as response to the lookup. Third, MDs connect to one of the SPs and disconnect from the P2P network. To prevent SPs from being overloaded by too many MDs, they publish their availability in the network only, if they currently have enough resources left. Additionally, SPs store jobs of MDs in queues for later processing, or reject them, if too many requests arrive.

SP-MD communication: The client-server based communication between SP and MD is done as follows: The MD deposits a job at the SP and goes offline, waiting until the job has finished. It periodically contacts the SP (e.g. every 5 minutes), to see if the job is already finished. The SP processes the job, acting as usual FOP towards the P2P network. After finishing, the result is transferred to the MD. During this transfer, the SP consumes mobile services from the MD (e.g. advertisement service or SMS service, cf. Section 2). Specialized protocols are used in the MPA for SP-MD communication, explicitly supporting the air interface. Standard P2P protocols do usually not consider varying delay or bandwidth of MDs. Also the existence of dead spots and IP changes of MDs is usually not considered (cf. Section1). The communication is improved by using suitable transport layer and application layer protocols. Transport protocols for wireless communication are discussed in [20] for instance. The application layer protocol supports the compression of data and data resuming, as it is done in the *File Transfer Protocol* [17]. However, the development of full featured protocols for the SP-MD communication is not topic of this paper.

Free Riding: Another important aspect is to assure fair resource exchange. Free riding SPs might attempt to consume mobile services without providing sufficient support to MDs and vice versa. Besides the opportunity of using closed source software (Skype [10] uses this approach), other mechanisms are possible. MDs assure download of content by providing mobile services interleaved, during the download of content. Ideally, MDs are able to preview parts of the downloaded data. If SPs do not gather requested data for MDs, transferring rubbish instead, MDs stop providing mobile services. Assuring delivery of mobile services depends on the kind of mobile service. Considering advertisement services (cf. Section 2) the MDs user is forced to push a certain key combination displayed in the advertisement. If mobile services are not delivered, SP stop data transmission. Assuring SMS services (cf. Section 2) is a more complex task and has not yet been evaluated. It is suspected that it can be done by using a part of the SMS messages as control messages, sent to a known recipient, confirming the receipt.

The MPA has been implemented prototypically, using JAVA and J2ME language. An MD (Sony Ericsson S700i) has been enabled to participate in the popular eDonkey file-sharing network [4]. An SP has been implemented, extending an usual MLDonkey [7] peer. Two different mobile services have been implemented, an advertisement service and an SMS service, as described in Section 2. Except of a specialized communication protocol (which has been simplistically implemented, using TCP with no further features) the architecture has been realized as described above.

4 Evaluation

The MPA prototype enables MDs to download files from the popular eDonkey P2P file-sharing network. It turned out that the download of files requires only short online times and produces little overhead. Also mobile services are delivered to the SPs properly. During file downloads, MDs send SMS messages or show advertisements on their displays, on behalf of SPs.

In this section the MPA's performance is evaluated in detail, in terms of *on/offline times* and *signalling overhead*. The performance of MPA is compared to an MD which runs out-of-the-box P2P software, directly connecting to a P2P file-sharing network (cf. Section 1). This case is called *mobile ordinary peer* (MOP) in this Section. To provide reference values, also the performance of an FOP is illustrated. The FOPs results can be considered to be the same as the results for SPs, since SPs are extended FOPs (cf. Section 3).

Three different experimental scenarios have been done. In each of them an MP3 file of approximately 2.3 megabytes size has been downloaded by all candidates. The first and second scenario have been deterministic experiments, done in isolated private networks with negligible variations of the results (due to computer clock skews). The third scenario has involved downloads from the real eDonkey network.

A simplified experiment setup has been chosen to get a clear comparison of the different architectures, excluding complex effects. Only bandwidth restrictions of up/download and the maximum number of simultaneous TCP connections have been modeled for MDs. Other issues involving mobility, varying bandwidth, or dead spots have been excluded. More complex conditions however, would not be a disadvantage for MPA in comparison to MOP, because the SP-MD communication is more likely to adapt to it than standard P2P communication (cf. Section 3). All devices have been standard computers with Debian Linux and fixed network connections. The limitations of each device have been modeled by adapting configuration files of P2P software[1]. In the MPA and MOP approach, the upload of the MD has been limited to 3 kilobytes/sec and the download to 6 kilobytes/sec, both values are related to restrictions of GPRS air interfaces. The number of allowed simultaneous download connections has been restricted to 5, related to common limitations of MDs (cf. Section 1). The SPs and FOPs upload has been limited to 10 kilobytes/sec, and the download to 30 kilobytes/sec, both common configuration values.

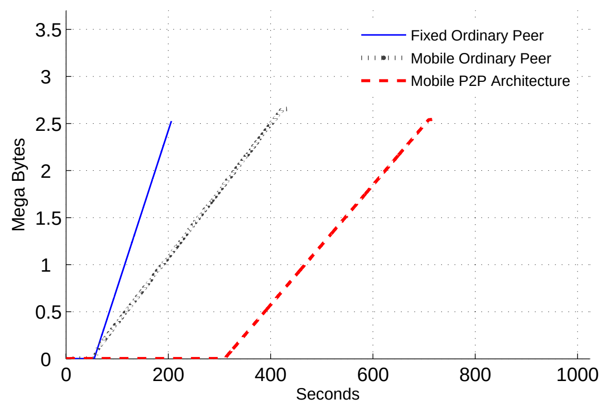

Figure 2: Download of an MP3 without competition

The first experiment assumes a competition-free scenario in the network. A single FOP offers an MP3 file and no other peer than the candidate requests it for download. Figure 2 illustrates the download processes of the three candidates (MPA, FOP, and MOP). Time 0 at the X axis denotes the moment when a download request for the content has been issued. Bootstrapping times and times for looking up files have not been considered in the experiments. It can be observed that MOP and FOP have started their downloads at nearly the same time, after a short waiting period in the queue of the serving peer. As expected, the FOP has completed its download faster, since it has the highest bandwidth. The MPA's MD had to wait for the SP's download first (SP download times are similar to FOP download times), before downloading the requested file from the SP. Although the MPA finished its download last, its advantages become clear in Figure 3, showing online and offline times of the candidates. The MPA's MD had less online time

[1]MLDonkey [7] and aMule [3] have been used to connect to the eDonkey network.

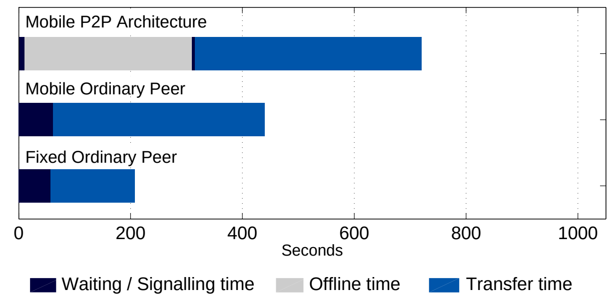

Figure 3: On/offline times without competition

than the MOP. Saving online time is an important issue for MDs (cf. Section 1). At time 0 the MD contacted the SP shortly to initiate the job (illustrated in dark color). After that it went offline (light gray color), waiting for the SP to finish. After 300 seconds (a user configurable value) it contacted the SP and downloaded the MP3 immediately in a short transfer (blue color). MOP (and FOP) otherwise, had to stay online from the initiation of the download until its end. Both had to wait in queues for the download to begin (waiting periods are illustrated in dark color). Another advantage of the MPA becomes

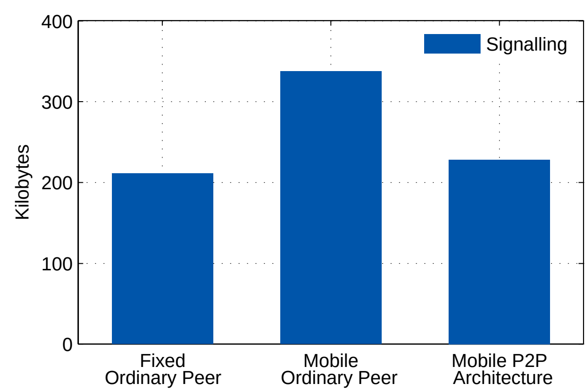

Figure 4: Signalling overhead without competition

clear in Figure 4. It illustrates the signalling overhead of the candidates, in terms of kilobytes transferred in addition to the MP3 data. It can be observed, that the MOP transfers more bytes than the MPA. This MOP overhead has been identified as eDonkey signalling traffic and does not depend directly on the size of the downloaded file. Instead, it depends on the duration of a download, including waiting times.

The second experiment shows, how competitors in the P2P network influence the download performance of each candidate. In contrast to the first experiment, three additional FOPs have been downloading the content in competition to the candidates. The configuration has been the same as above and all of the FOPs have

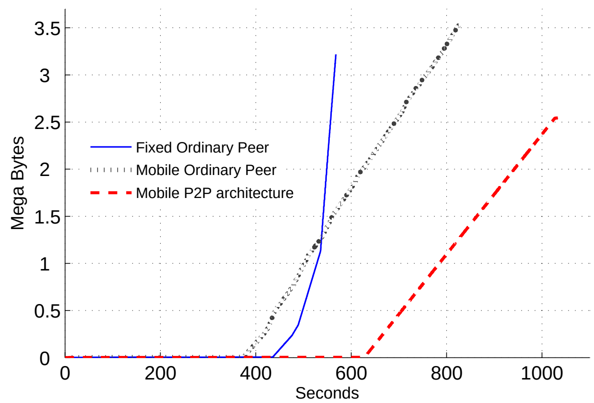

Figure 5: Download of an MP3 with competition

been configured similarly. Additionally, the FOP offering the MP3, has been limited to provide three upload slots only, in order to create a bottleneck. The candidates started their download 30 seconds after the FOPs, resulting in queue waiting times. Figure 5 illustrates the corresponding download processes. It can be clearly seen, that the downloads have been starting considerably later than in the first experiment. The evaluated FOP (similar to the SP) has succeeded first. Again, the MPA's MD started and finished its download last. The

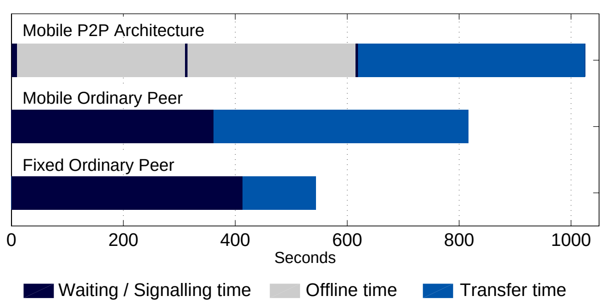

Figure 6: On/offline times with competition

on/offline times shown in Figure 6 illustrate clearly the performance of the MPA. The FOP's and MOP's waiting times have considerably increased (dark color and blue color), compared to the first experiment. In contrast, the MPA online time remained nearly the same (dark color and blue color), only the offline time (light gray color) increased. It can be seen, that the MD checked twice (after 300 seconds and after 600 seconds) if its SP had already finished the job, before downloading the file. A second clear advantage of the MPA is illustrated in Figure 7, where the overhead is compared. Due to the longer waiting times of FOP and MOP, their overhead increased considerably. The overhead of the MPA's MD however, did almost not change compared to the first experiment.

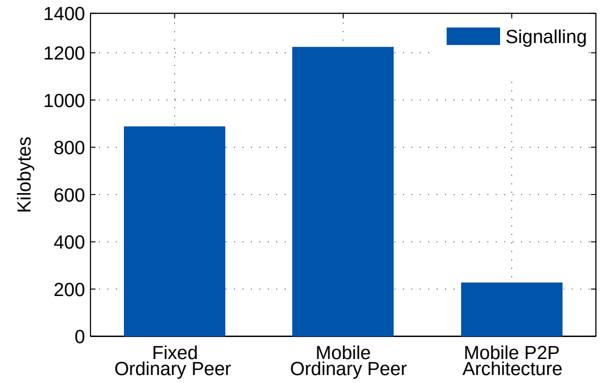

Figure 7: Signalling overhead without competition

The second experiment showed that competition in the P2P network strongly influences the performance of the candidates. Intuitively, it can be expected that a more severe competition situation degrades the performance of the MOP, but not of the MPA's performance. To verify this expectation a third experiment has been done in the real eDonkey network, where every candidate can have up to thousands of competitors. The experiment's setup has been similar to the previous ones. However, the candidates have been directly connected to the Internet. FOP and SP have been connected via fixed links with high bandwith, MOP and MPA's MD have been connected via real GPRS air interfaces. They all downloaded a popular MP3 file of approximately 2.3 megabytes. In Figure 8 it can be seen, that the

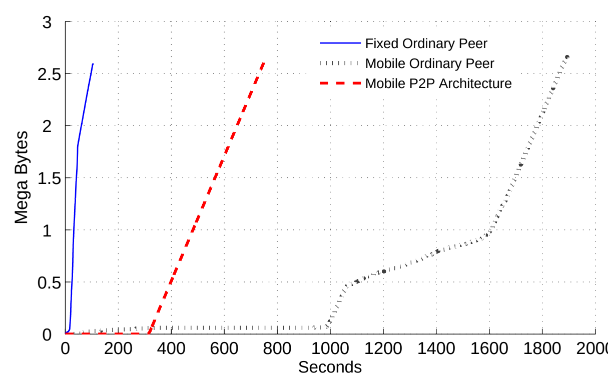

Figure 8: Download of an MP3 in the real eDonkey network

MOP has not been able to compete with the FOP or the MPA. MPA and FOP started and finished their download even faster than in the second experiment, whereas the MOP needed 16 minutes to start and 31 minutes to finish. Although the experiment was repeated several times, the MOP never managed to download the file in less than 25 minutes, spending most of the time waiting in queues. Figure 9 shows that the MPA clearly outperforms the MOP in the third experiment. The MPA's online time has remained very low, profiting from the SP's fast download (similar to the FOP's results). The MOP however, has spent half of its online time waiting in queues. The differences in signalling overhead, illustrated in Figure 10, have not been has high as expected. It is suspected that the results are caused by variable parameters in the experiment, which have not been further evaluated. Examples for these parameters are e.g. different software and different configurations of real peers, to which the candidates connected, or a firewall of the mobile operator, influencing the MOP. The MPA however, produced approximately the same overhead as in the experiments before.

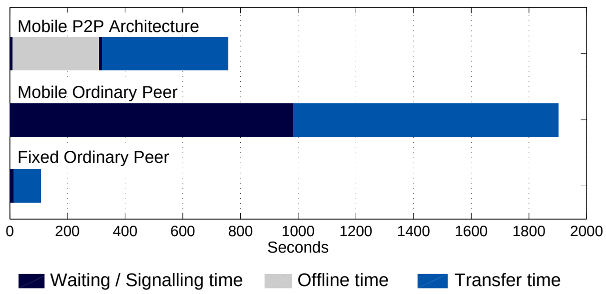

Figure 9: On/offline times in the real eDonkey network

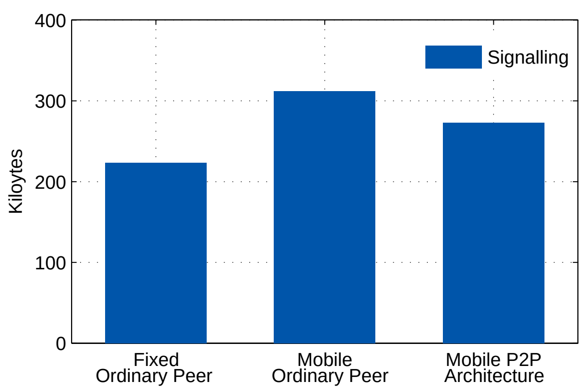

Figure 10: Signalling overhead in the real eDonkey network

5 Related Work

There are several mobile P2P approaches focusing on infrastructure-based cellular networks. They can be distinguished in two different classes.

In the first class, either all peers of a P2P network are involved in the support for MDs, or peers with certain properties (e.g. high bandwidth) are especially determined to provide support. To achieve this, P2P protocols are modified or newly developed. An example for the first class is the *hybrid chord protocol* [22]. It modifies the well known chord protocol to cope more efficiently with the effects of mobility. Peers are divided into *static* nodes and *temporary* nodes. Temporary nodes (nodes with short online times), are relieved from storing object references, improving the overall network performance. Other approaches suggest P2P networks, in which fixed Internet peers are determined to support MDs by aggregating or filtering data for them. In these solutions, MDs are partly or entirely relieved from network maintenance and routing tasks. In [19] *proxy servers* are used to integrate MDs into a P2P architecture. In [14] *surrogate peers* are supporting MDs and the JXME [5] project defines *relay peers* to connect mobile peers to the JXTA [6] P2P environment (nowadays MDs are also able to participate *proxyless* in JXTA). However, due to the protocol change which is applied to all peers, these solutions do not fit to existing P2P networks. These networks already have a large user community in the fixed Internet. Protocol modifications or newly developed protocols are hard to establish in this community, especially when peers are forced to carry additional load in order to support MDs.

Solutions of the second class do not change the protocols of the main user community. Instead, voluntaries within the network (fixed network peers) are changing or extending their protocols to provide support for MDs. An example for the second class is *MobileMule* [8]. It is a project in which users support their own MDs by running a second device in the fixed Internet. However, the MDs are not really integrated in the P2P network. They just remote control the fixed computers, being not able to download or share any content. Another approach requires voluntary support from the operators of cellular networks. To provide this support, operators have to invest in additional hard- and software. The *MoPi architecuture* [15, 16] is an example for this kind of approach. In the project, additional architectural elements (a *cache peer*, a *crawling peer* and an operator driven *index server*) are placed within the operators domain, to integrate MDs to the popular eDonkey network. The MDs communicate mainly with the special components, being separated from the outside P2P network. The P2P protocol is not changed for any other peer. Also MDs use common P2P protocols. Their software is adapted to run on MDs. Examples for adapted P2P software are Mopiphant [9], a client for the eDonkey network, or Symella [11], a client for the Gnutella network [18]. Although this operator driven solution works well, it has its disadvantages. There are legal issues, because the operators architectural elements are acting on behalf of mobile users, who might deal with illegal content. The operator on the other hand is able to eavesdrop and record relayed information. In contradiction to the P2P concept, the architectural elements are centralized solutions, imposing single points of failure and scalability problems.

Another disadvantage is that this solution is based on a change (or extension) of the operators infrastructure, which is a complex and expensive task to do.

None of the described approaches achieved a widespread integration of MDs into popular P2P content-distribution networks. A main weakness of these approaches is the imbalance of cooperation. MDs have to be supported without receiving compensation from them. Even the mentioned operator driven support is not compensated directly, instead it is compensated by the satisfaction of users and a raised utilization of infrastructure. Due to this imbalance, required voluntary support is hard to get in P2P content distribution networks. In this paper in contrast, a solution is suggested which compensates the support of MDs, providing access to advanced mobile features and services of MDs.

6 Conclusion

The integration of MDs into popular P2P content distribution networks is a difficult task. In this paper the problems have been summarized and a new mobile P2P architecture has been suggested as a solution. The main innovation in this architecture is a novel visionary view on MDs. MDs are regarded as providers of advanced mobile features and services instead of considering them as bottlenecks within P2P networks. It is suggested to exploit the heterogeneity of available features and services in P2P networks, consisting of mobile and fixed devices. The suggested architecture shows, that mobile services can be made accessible for fixed peers in P2P networks. Certain fixed peers support MDs in P2P specific operations and consume mobile services in return. This support is provided voluntarily, protocols of other peers remain unchanged. The architecture has been implemented as prototype and been evaluated. The evaluation has shown that MDs are successfully able to participate in P2P networks. They remain offline most of the time, mainly being online to download content. Also different mobile services have been consumed successfully by fixed peers, during the support of MDs.

In future work further partnership schemes between peers in the fixed network and MDs will be evaluated in detail. Additionally the architecture implementation will be extended to a full featured software, being able to operate in different P2P content distribution networks.

References

[1] Current mobile phones and features. https://developer.sprint.com/show_devices.do. (last visited: 04/12/2007).

[2] Google-welcome to adwords. https://adwords.google.com. (last visited: 04/12/2007).

[3] Homepage of the amule project. http://www.amule.org. (last visited: 04/12/2007).

[4] Homepage of the edonkey project. http://www.overnet.org. (last visited: 04/12/2007).

[5] Homepage of the jxta for j2me project. http://jxme.jxta.org. (last visited: 04/12/2007).

[6] Homepage of the jxta project. http://www.jxta.org. (last visited: 04/12/2007).

[7] Homepage of the mldonkey project. http://mldonkey.sourceforge.net. (last visited: 04/12/2007).

[8] Homepage of the mobilemule project. http://mobil.emule-project.net. (last visited: 04/12/2007).

[9] Homepage of the mopiphant project. http://www3.informatik.uni-wuerzburg.de/staff/mopi/mopiphant.shtml. (last visited: 04/12/2007).

[10] Homepage of the skype project. http://www.skype.com. (last visited: 04/12/2007).

[11] Homepage of the symella project. http://symella.aut.bme.hu. (last visited: 04/12/2007).

[12] Infin micropayment. http://www.infin-online.de:2080/minis/mp/index.php. (last visited: 04/12/2007).

[13] S60 platform: Ip bearer management. http://sw.nokia.com/id/190358c8-7cb1-4be3-9321-f9d6788ecae5/S60_Platform_IP_Bearer_Management_v1_0_en.pdf. (last visited: 04/12/2007).

[14] T. Horozov, A. Grama, V. Vasudevan, and S. Landis. Moby - a mobile peer-to-peer service and data network. In *ICPP '02: Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02)*, page 437, Washington, DC, USA, 2002. IEEE Computer Society.

[15] T. Hoßfeld, K. Tutschku, F.-U. Andersen, H. de Meer, and J. Oberender. Simulative performance evaluation of a mobile peer-to-peer file-sharing system. In *Next Generation Internet Networks NGI2005*, Rome, Italy, 4 2005.

[16] J. O. Oberender, F.-U. Andersen, H. de Meer, I. Dedinski, C. Kappler, A. Mäder, and K. Tutschku. Enabling mobile peer-to-peer networking. In *Lecture Notes in Computer Science*, volume 3427, pages 219–234, 2005.

[17] J. Postel and J. Reynolds. File Transfer Protocol, RFC959. *Network Information Center, October*, 1985.

[18] M. Ripeanu. Peer-to-peer architecture case study: Gnutella network. In *Proceedings of the First International Conference on Peer-to-Peer Computing*, pages 99–100, 2001.

[19] H. Sumino, N. Ishikawa, and T. Kato. Design and Implementation of P2P Protcol for Mobile Phones. In *Proccedings of the 3nd IEEE International Workshop on Mobile Peer-to-Peer Computing (MP2P'06)*, Pisa, Italy, Mar. 2006.

[20] Y. Tian, K. Xu, and N. Ansari. Tcp in wireless environments: Problems and solutions. *IEEE Communications Magazine*, 43(3):27 – 32, 2005.

[21] K. Tutschku, T. Hoßfeld, A. Berl, H. de Meer, and J. Oberender. Mobile p2p applications in infrastructure-based mobile networks. Technical report, Department of Distributed Systems, University of Würzburg, Germany, 2007.

[22] S. Zöls, R. Schollmeier, W. Kellerer, and A. Tarlano. The hybrid chord protocol: A peer-to-peer lookup service for context-aware mobile applications. Springer, 2005.