Mapping of File-Sharing onto Mobile Environments: Enhancement by UMTS

Tobias Hofeld, Kurt Tutschku
Department of Distributed Systems
University of Würzburg
97074 Würzburg, Germany.
[hossfeld|tutschku]@informatik.uni-wuerzburg.de

Frank-Uwe Andersen
Siemens AG
Com
13629 Berlin, Germany.
Frank-Uwe.Andersen@siemens.com

Abstract

Peer-to-Peer (P2P) file-sharing has become the killer application in the Internet with respect to traffic volume which is even surpassing web usage. This characteristic makes P2P commercially attractive to network operators interested in increased traffic. In parallel, the demand for wireless services has caused wireless networks to grow enormously. We assume that P2P file-sharing will be mapped onto mobile environments by its users. This results in a mobile *P2P file-sharing service. In this paper, we examine the feasibility of the eDonkey file-sharing service in GPRS and UMTS mobile networks, detect problems of the interaction between P2P and the mobile network, and outline first solutions to overcome them. The goal is the analysis of feasibility for an Internet-based file-sharing application in mobile networks and to provide real-world measurements. The measurements have been carried out in networks of two German GPRS network operators and, for the first time, for a UMTS network.*

1 Introduction

P2P file-sharing has become the killer application in the wired Internet. It has grown far more rapidly than web browsing in terms of traffic volume [2]. P2P file-sharing might also be highly attractive for mobile networks. UMTS network operators, in particular, are searching for new applications for their systems. So far, applications for these networks are missing which do both: *a)* exploit, qualitatively and quantitatively, the potential of the UMTS technology and *b)* motivate the user to adopt the new technology. *Mobile P2P file sharing* might be an interesting candidate for such an application. To get an impression of the behavior of P2P in mobile networks, we present case-by-case measurements of mobile P2P for GPRS and UMTS networks.

P2P applications, however, have also some downsides. P2P is trading its decentralized nature by increased communication traffic. In particular, the peers generate a considerable amount of signaling traffic for coordinating with each other [7, 8]. High application signaling traffic is considered to be too expensive in mobile networks. This shows the importance of traffic measurements for optimizing mobile P2P in the sense of an operator supported service, e.g. caching strategies to reduce bandwidth or signaling traffic [12].

The aim of this paper is to examine the feasibility of mobile P2P and to give an insight how a general P2P architecture works in a mobile cellular environment. We detect problems of the interaction between P2P and mobile networks, e.g. restrictions because of the air interface, and describe how obstacles, such as network address translation (NAT) or firewalls, can be overcome. Finally, this paper measures and analyzes the characteristics of mobile P2P using GPRS and UMTS transmission technology and gives first empirical performance values.

The broader scope of work is to use the experience gained from our measurements in order to identify mobile P2P specific problems. These are addressed by an architecture proposal recently published [6].

Currently a number of P2P file-sharing applications are available. Due to its current popularity among users [11], the eDonkey 2000 system[1] is used as a candidate for mobile P2P in this study. We assume that the popularity of an application is of greater importance for the selection than an easy implementation in mobile networks. Our investigation of mobile P2P is based on the measurements of GPRS-based mobile P2P services since this service is widely available and

[1] In this paper, we subsume eDonkey 2000 and its derivatives, e.g. eMule, mlDonkey, by the single term "eDonkey".

the architecture of the fixed network core of GPRS and UMTS are almost identical. The major findings for the GPRS service are subsequently extended for the UMTS radio bearer type.[2]

2 P2P Architecture

The eDonkey file-sharing service [1] belongs to the class of hybrid P2P architectures comprising two components: the eDonkey client and the eDonkey server[3]. The eDonkey client is used to share and download files. The eDonkey server operates as an index server for file locations and distributes addresses of other servers to clients. The consuming client may operate in a *multiple source download* mode, i.e. it issues two or more requests in parallel to different providing clients. The uploading client keeps the outstanding requests in a list of current downloading requests. Then, the user data is transmitted in several parallel TCP connections from the uploading peers to the requesting peer. The upload management of a peer maintains an upload queue which consists of two lists, the waiting list and the uploading list. The uploading list holds the exchange requests which are currently served. Each served request gets typically an equal share of the upload capacity which may be restricted to a given limit. A download request is served as soon as it obtains an upload slot, i.e. it moves from the waiting list to the uploading list. The complex scoring mechanism of eDonkey decides which request is served next. One important factor of the scoring system is the "high ID/low ID"[4] mechanism to ensure fairness for peers before or behind a NAT or a firewall. A high ID increases the score whereas a low ID reduces it. A peer gets a low ID if it located behind a firewall or a NAT since other peers cannot initiate new connections to this peer. This results in an unfair behavior as the peer does not answer file requests and thus does not share its content.

3 Mobile Network Characteristics

General Packet Radio Service (GPRS) is the current, GSM based infrastructure and the confluence of mobile telecommunications and IP data networking. GPRS data rates depend on the overall number and ratio of voice and data users in a cell and the supported data rates of the mobile station (MS). GPRS applies dynamic bandwidth allocation which is mainly based upon granting circuit-switched voice traffic priority, including the option to stop data communications in favor of voice calls. The combination of uplink/downlink channels depends on the *class* of the mobile terminal. A *class 8* mobile station, for example, is limited to 1 uplink and 4 downlink channels which yields theoretical data rates of 13.4 kbps for the uplink and 53.6 kbps for the download link. In GPRS, the air interface is the lossy part of the link. The eDonkey (v.0.40f) application uses TCP for transmitting user data. As a result of the mobile environment, TCP suffers from packet retransmissions due to packet losses [4]. All IP traffic is centrally directed through the GGSN network element. For any two MS exchanging IP data between them, the entire path up to corresponding GGSN(s) needs to be traversed twice, even though the SGSN is the same. This results in high delay times. Figure 1 shows the increasing transfer delay of an IP packet on the data path between two terminals in an unloaded network, identifying the crucial parts of the overall path (based on measurements performed by Siemens). It has to be considered that the first packet in a packet stream between two mobiles experiences a significantly higher delay than the following ones because of temporary block flow (TBF) setup times [3]. GPRS brings IP-based services to the mobile mass market and has paved the way for UMTS networks.

Universal Mobile Telecommunications System (UMTS) networks differs from GPRS networks, among other things but mainly, by the use of Wideband Code Division Multiple Access (W-CDMA). Currently, the UMTS networks that have been rolled out permit an uplink bandwidth of up to 64 kbps and a downlink capacity of 384 kbps for packet data transmission in unloaded conditions. The core network architecture of Release99 UMTS networks matches widely the core architecture of GPRS networks [9].

4 Problems of Mobile Peers Using eDonkey

Some German mobile operators assign private IP addresses and shield the mobile peers by firewalls and NAT. Such peers would be assigned low IDs resulting

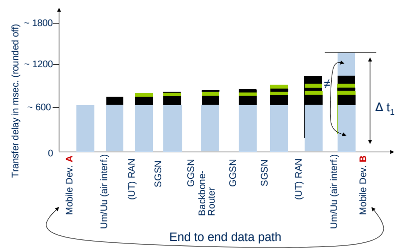


Figure 1. Delay of an IP packet on the path between two terminals

[2]The work, presented here, extends previous measurements [10].

[3]The terms "client" and "peer" are exchangeable in the context of eDonkey.

[4]The eDonkey *ID* identifies peers and is assigned upon registration of a client at an index server.

in a discrimination in the upload queue. To avoid this, a consistent address space is required which can be realized artificially by applying a virtual private network (VPN). Using a VPN means that all peers and the index server must be included in the VPN. In our measurements, we used a Point-to-Point-Tunneling Protocol (PPTP) based VPN.

It is interesting to examine how expensive a VPN is as a solution. The costs are expressed by the protocol overhead, the download time, and the received bandwidth on application layer. The overhead of an encapsulated data packet via VPN is at least 28 Byte [5]. The application of a VPN might also lead to an increased number of packets due to fragmentation.

Is a multiple source download possible in a mobile network?
It may be conceivable that the performance of a multiple source download significantly differs from a single source download with a small (or zero) number of users in the waiting list of the upload queue due to increased overhead for coordinating multiple sources and the limited bandwidth in mobile environments.

Is the performance influenced by the content type?
Regarding the peer's mobile equipment, a set of content types seems to be typical for mobile P2P users. Today, mobile handsets support multimedia, e.g. polyphony ringing tones, self-recorded audio files or pictures and small movies from integrated digital camera. These file types are reflected by different distributions of the file size. On the other hand, the memory capacities are limited up to several megabytes. We investigate the performance of a mobile P2P service with respect to the file sizes of the different content types, e.g. to answer the question whether is it practical do download mp3-audio files.

5 Measurement Scenarios

The **GPRS Measurements** took place in between Dec. 2003 and Feb. 2004, see also [10]. We selected two German GPRS operators, A and B. Operator A assigns global IP addresses to mobiles which enables arbitrary direct communication between peers. Provider B uses a firewall which denies mobile-terminating TCP connections, except connections to an external VPN gateway. Two alternatives have been considered during the measurements. First, the eDonkey application uses GPRS as a bit pipe connection to the public Internet: the public Internet scenario ("pub"). Secondly, the file-sharing application resides in a VPN and uses virtual connections for exchanging information: closed network scenario ("vpn"). The physical access of a peer can be either Ethernet for fixed network access (max. 100 Mbps) or GPRS for mobile access

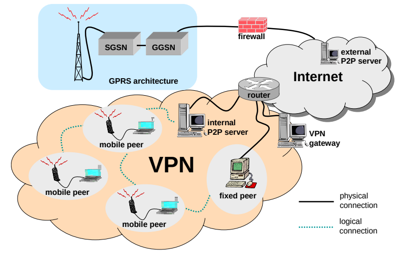
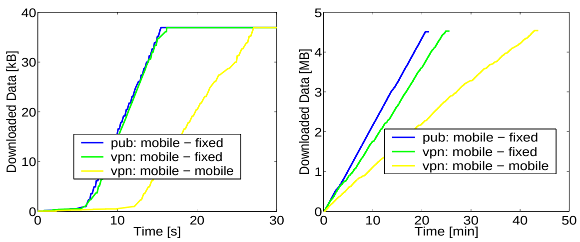


Figure 2. Network architecture for a mobile P2P file-sharing service

Table 1. Parameters for measurements

P2P application	eMule 0.40f
mobile phone	Siemens S45/S55/ME45; multislot-class 8; as modem via RS232
operating system (peers)	Windows 2000 (SP4)
packet capture software	WinDump 3.6.2 (using PCap 2.3)
VPN gateway	PoPToP v1.1.3 - a freeware PPTP server running under SuSE Linux 8.2, kernel version 2.4.20
internal eDonkey server (non-public)	eserver 16.43-i686 (Lugdunum)
external eDonkey server (public)	207.44.200.40:4242 with more than 50,000 users and 2,900,00 files

(max. 53.6 kbps). The fixed peers, the internal eDonkey server, and the VPN gateway are located within a LAN of the university which is connected to the German Research Network by 100Mbps. Figure 2 shows the closed network architecture for a mobile P2P file-sharing service over GPRS. This architecture differs from the public Internet scenario by the application of a VPN. In the closed network scenario, it is not possible to communicate to entities (peers, index servers) which are not connected to the VPN. The VPN scenario is only used with operator B. The mobile P2P clients consist of a mobile phone which is used as a modem and a laptop running Windows2000. The used mobile phones (Siemens S45, S55, and ME45) support the GPRS multislot-class 8. A complete packet trace was captured for every mobile or fixed peer during the measurement campaign. The external P2P server in the public scenario is a well-known eDonkey server with a fixed IP address. The internal P2P server is part of the LAN at the Department of Distributed Systems. The used software and hardware for the measurements are summarized in Table 1. In order to investigate a single source download, a peer provides a unique file that is yet unknown to the eDonkey network. That



(a) Game (b) Song

Figure 3. SSD in GPRS: Transmitted data

way, it can be assured that the number of sources to download from is exactly one. A multiple source download is realized by downloading a popular file which is shared by at least two peers.

The investigated file types represent a typical set of files with typical sizes that seems to be interesting for mobile users [10]: ring tones (avg. size: 6,830 Byte), games for java-capable mobiles (39,114 Byte), digital camera images (483,525 Byte), and mp3-audio files (4,726,618 Byte).

The **UMTS Measurements** took place in Aug./Sep. 2004. We selected only the operator A since this is still the only one that assigns global IP addresses to its subscriber's mobiles. It should be noted that that due to the relatively new service, the load conditions of the UMTS cell were relatively low and thus we encountered relatively low delay and high bandwidth. The general measurement setup matches the one for the GPRS measurements, except that we used the Vodafone Mobile Connect UMTS PC-card as the modem for the Windows2000 laptop.

6 Results

6.1 Feasibility of Mobile P2P in GPRS Networks

First, we consider a single source download (SSD) fixed-to-mobile and mobile-to-mobile of files using operator B for the VPN scenario and the public scenario (despite the latter causes low IDs). The downloading peer has mobile access. The physical access of the sharing peer[5] is chosen to have fixed (Ethernet) or wireless (GRPS) connectivity.

Figure 3(a) illustrates the amount of downloaded data in [kB] over time in [s] for downloading the game; the legend classifies the scenarios according to the scheme "[network scenario]:[access type of downloading peer] - [access type of sharing peer]". The behavior of the mobile downloading peer and the fixed sharing peer are similar and independent of the use VPN because of the small file size. In contrast, Figure3(b) shows that it takes indeed more time to transmit the song with the VPN. The download time is increased for direct downloading a file from a mobile peer, as the uplink of the sharing mobile peer is the bottleneck and limits the download bandwidth. After having done the measurements for operator B, we performed them for operator A. The results are the same with respect to download time, transmission rate, and packet loss.

[5]The terms "sharing peer" and "serving peer" are exchangeable in this work.

6.1.1 TCP Retransmissions and Aborted Downloads

Packet retransmission may occur in GPRS through packet loss on the air. We observed for downloading a mp3 file (fixed-to-mobile) very small retransmission probabilities (averaged over 10 file exchanges) of 0.26% using the VPN and 0.44% without the VPN.

A download is detected to be aborted if no more data is send from the sharing peer to the downloading peer for at least 10 min or if the GPRS connection hangs up. If the sharing peer has fixed access, no aborts were observed. However, if all involved peers use mobile access, we noticed a significant abortion rate of downloads. It should be noted that only single source download is used, here. However, the number of aborted downloads for large files (mp3) differs significantly between operator B and A, cf. Table 2. In order to explain the aborted downloads, we investigated the exchange of the same file by FTP between the mobiles. Again, we noted a higher success rate for operator A. A reason for this observation cannot be derived directly by our measurements. The most likely explanations are errors in early software implementations of mobile handsets and network infrastructure.

6.1.2 P2P Setup Time, Download Time, Idle Time

The *P2P setup time* is defined as the time period from the observation of the first TCP SYN packet to the first TCP packet containing user content. The *download time* is the time interval from the observation of the first TCP to the last TCP packet containing user content. The *idle time* is considered as the time from the last TCP packet containing user content until the observation of a TCP FIN or TCP RST packet for this connection.

Table 2. Success rate for GPRS

appl.	downl.	sharing	operator	success
eDonkey	mobile	mobile	B	$0 = 0 : 5$
FTP	mobile	mobile	B	$0.5 = 4 : 8$
eDonkey	mobile	mobile	A	$0.6 = 3 : 5$
FTP	mobile	mobile	A	$0.75 = 6 : 8$

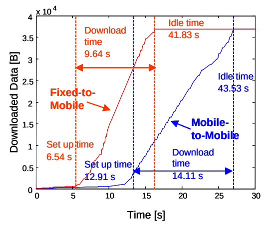


Figure 4. Intervals a game

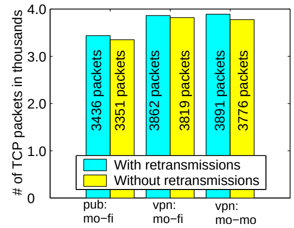


Figure 5. Num. of TCP packets

Figure 4 depicts the above introduced time intervals for single downloads (fixed-to-mobile and mobile-to-mobile) of a game. Since in the mobile-to-mobile case, the air interface has to be passed twice, the setup time is twice as much as in the fixed-to-mobile case, see also Figure 1. The download time is determined by the minimum of the download bandwidth of the requesting peer and the upload bandwidth of the sharing peer. Therefore, the download time is significant larger in the mobile-to-mobile transfer. The idle time is independent of the connection type and dominated by a timeout mechanism of the eDonkey application.

6.2 Overhead due to VPN

The overhead introduced by using a VPN is described by the increased data volume due to the PPTP header information and by the higher number of transmitted packets due to segmentation. Figure 5 shows the number of transmitted TCP packets for three scenarios. As expected, using a VPN leads to a slightly higher number of transmitted packets.

Figure 3(b) reveals that the same amount of user data is transmitted, however, the download takes longer due to the transmission of additional header information.

6.3 Multiple Source Download (MSD)

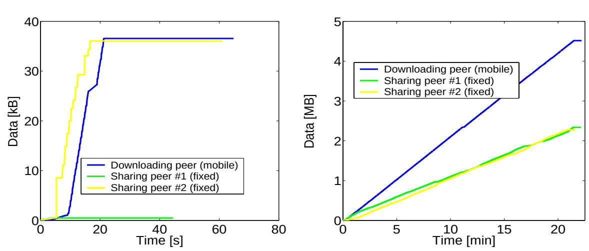
First, we consider two peers sharing the complete file and a single peer requesting the download. Figure 6 shows the uploaded data volume of the two sharing peers with fixed access and the total downloaded data volume of the requesting mobile peer. MSD does not become effective for small files, like the game, cf. Figure 6(a). In this case, the requesting peer receives in one download connection almost all of the requested data. Contrary to this, we observe for large files an efficient MSD, cf. Figure 6(b). The requested data volume is equally split between the two sharing peers.

In the second scenario, we investigate the influence of the access type of the requesting peers on the MSD mechanism, while one sharing peer has mobile access and the other one has fixed access. Figure 7(a) shows the MSD for a downloading mobile peer. This case reveals the asymmetry of the mobile equipment, see Section 3. The mobile downloading peer has four slots for downloading data. The mobile sharing peer can only use one slot due to his uplink restrictions. The remaining downlink capacity of the mobile downloading peer is utilized by the fixed sharing peer.

Figure 7(b) depicts the MSD behavior for the same scenario with a fixed downloading peer issuing requests to all sharing peers. The fixed sharing peer serves this request with high throughput. The mobile sharing peer is also serving the file request, immediately. However, he provides the minimal amount of data eDonkey transmits for request (which is in eDonkey three blocks, each of 180 kB). The downloading peer completes the file after receiving the data from the mobile sharing peer. A redirection of the download request to another peer which can serve the request faster would reduce the download time.

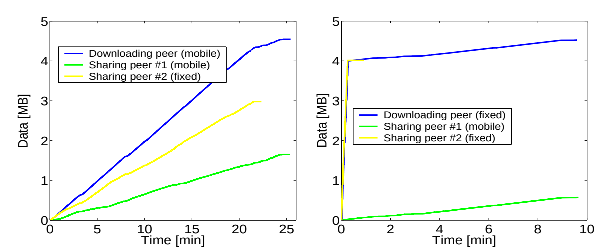
6.4 Evaluation of Mobile P2P via GPRS

The measurements in the previous section have demonstrated the feasibility of mobile P2P file sharing in GPRS with respect to fair bandwidth sharing for MSD, packet retransmission, VPN overhead, and connection setup times. The performance with respect to throughput and service stability for long files (cf. Table 2), however, is poor. The UMTS service, in contrast, is promising improved throughput and high stability. This will be validated by selected measurements



(a) (b)

Figure 6. MSD: two peers - fixed access



(a) (b)

Figure 7. MSD: large file - different access

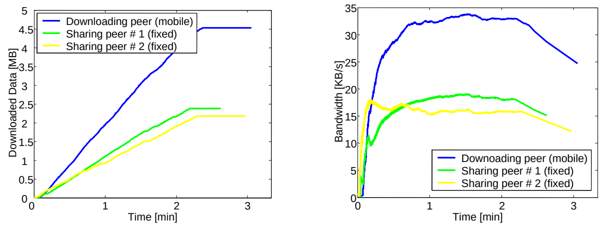
in the next section.

6.5 Expected Performance of Mobile P2P in UMTS Networks

Figures 8(a)-11(a) depict the amount of downloaded data over time for exchanging a large mp3 song file either by Single Source Download (SSD, Figure 8) or by Multiple Source Download (MSD, Figures 9-11). Figure 8(a) shows that there is only small differences in the download time between using a VPN or a direct connection. In addition, the figure clearly reveals the strong asymmetric bandwidth split in UMTS between uplink and downlink. The uplink of the sharing peer is the bottleneck in a mobile-to-fixed file exchange. This observation is acknowledged in the case of a MSD by a mobile peer from two fixed peers, cf. Figure 9(a). The case of a MSD by a mobile peer from a fixed and mobile a peer it shown in Figure 10(a). Here, both sources are used and transmit data. This example demonstrates that MSD is feasible and fair in UMTS in heterogenous (mobile/fixed) environment. Figure 11(a) depicts the case of an all mobile MSD. Both providing peers are equally used. The limiting capacity is the uploading link of both providing peers.

6.5.1 Throughput

Observed throughput values for SSD and MSD in UMTS are given in Figures 8(b)-11(b). The throughput for a SSD, cf. Figure 8(b), reaches values up to 23 Kbytes/sec (without VPN) and is sightly smaller with using a VPN. For a mobile-to-fixed file exchanged, we observed a throughput of up to 7KBytes/sec which is sustained by the system. An interesting observation can be made in the case of a heterogenous download by a mobile peer from a fixed peer and a mobile peer, cf. Figure 10(b). Whereas the providing mobile peer supports an almost constant upload bandwidth (which is the uplink capacity), the upload bandwidth of the providing fixed peer varies. The variation influences the download throughput for the receiving mobile peer. This case shows clearly that the download capacity in



(a) Downloaded Volume (b) Throughput

Figure 9. MSD in UMTS (mob.:fix./fix.)

UMTS is not fully utilized and is well sufficient for supporting MSD. The equal bandwidth sharing in an all mobile file exchange is shown in Figure 11(b). Both providing mobile peers upload with the same bandwidth.

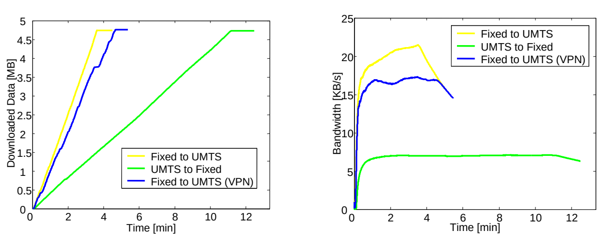
The throughput values of UMTS are far higher than the ones for GPRS [10] and demonstrate the strength of this technology. The obtained bandwidth appears to be sufficient for true mobile P2P file sharing.

6.5.2 P2P Setup Time, Download Time, Idle Time

These times are shown for typical SSD file exchanges (fixed-to-mobile and mobile-to-mobile) by UMTS in Figure 12. Again, for the mobile-to-mobile exchange, the setup time is twice as much as for the fixed-to-mobile case. The setup times are slightly smaller than in the GPRS example and indicates that this time is mainly related to the common core network architecture (cf. Section 3) of GPRS and UMTS and to the implementation of the eDonkey application. The idle times are in the same order of the times in the GPRS example, showing that this interval is determined by the application.

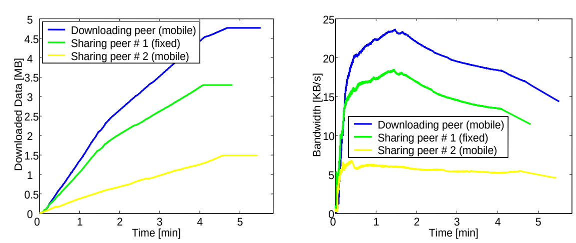
6.5.3 TCP Retransmissions and Aborted Downloads

We observed very small retransmission probabilities for downloading a mp3 file (SSD, fixed-to-mobile, averaged over 10 exchanges) of 0.43% (without the VPN) and



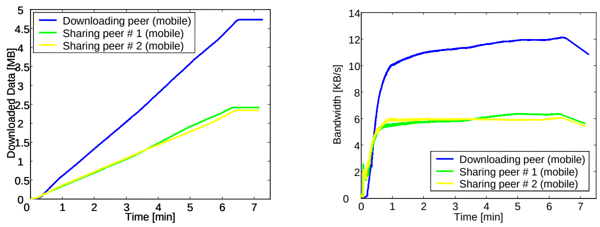
(a) Downloaded Volume (b) Throughput

Figure 8. SSD in UMTS



(a) Download Volume (b) Throughput

Figure 10. MSD in UMTS (mob.:fix./mob.)



(a) Downloaded Volume (b) Throughput

Figure 11. MSD in UMTS (mob.:mob./mob.)

0.13% (with VPN), which are in the order of retransmission in GPRS.

In contrast to GPRS, we have not observed any aborted file transmission in UMTS. It appears to be very stable despite its early state of deployment.

7 Conclusions and Outlook

In this work, we provided first measurements on the performance of a mobile P2P file-sharing service. The measurements were carried out in real-world networks for two different GPRS operators and one UMTS provider. We demonstrated that mobile P2P is technically feasible for GPRS technology but stability and throughput are unacceptable low if compared to fixed P2P. Particularly, the direct exchange of large parts of files between two mobile peers and multiple source download is not practical in GPRS. GPRS is well suited for exchanging small contents with "*Instant Messaging*"-like P2P applications, i.e. small files are transmitted in a single or few parts. UMTS technology, in contrast, is more stable and has superior throughput. It extends the capabilities of GPRS service into sufficient performance for mobile P2P file sharing. However, the number of traversals of the air interface has to be minimized for both technologies (cf. Figures 4 and 12) in order to reduce the traffic and the transmission delay. This could be achieved by the application of a cache, which has also the advantage of overcoming the asymmetric access bandwidths of mobile stations [6]. Multiple source download is not required for small files. As mentioned above, large parts of files should also not be transmitted. This characteristic indicates

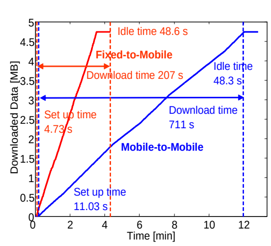


Figure 12. UMTS: Setup and Download Times

that there seems to be an optimal segment size for MSD which depends on the total file size and the capacity of the access of the sharing peers. In addition, sharing peers should be selected with respect to their throughput and responsiveness.

In future studies, we will perform additional measurements in order to obtain more comprehensive statistical characterizations of mobile P2P file-sharing and, in particular, to investigate mobile P2P for the UMTS radio bearer type.

References

[1] eDonkey2000 Home Page. `http://www.eDonkey2000.com/`.

[2] N. Azzouna and F. Guillemin. Experimental analysis of the impact of peer-to-peer application on traffic in commercial IP networks. *European Transactions on Telecommunications*, 15(6), 2004.

[3] C. Bettstetter, H.-J. Vogel, and J. Eberspacher. GSM Phase 2+, general packet radio service GPRS: Architecture, protocols and air interface. In *IEEE Communications Surveys*, volume 2, 1999.

[4] K. Brown and S. Singh. M-TCP: TCP for mobile cellular networks. In *ACM Computer Communications Review*, 1997.

[5] D. F. et al. *RFC 2784 - Generic Routing Encapsulation (GRE)*, 2000.

[6] F.-U. Andersen and H. de Meer and I. Dedinski and C. Kappler and A. Mäder and J. Oberender and K. Tutschku. . An Architecture Concept for Mobile P2P File Sharing Services. In *Workshop at Informatik 2004 - Algorithms and Protocols for Efficient Peer-to-Peer Applications*, Ulm, Sep. 2004.

[7] K. Tutschku. A Measurement-based Traffic Profile of the eDonkey Filesharing Service. In *5th Passive and Active Measurement Workshop (PAM2004)*, pages 12–21, Antibes Juan-les-Pins, France, Apr. 2004.

[8] K. Tutschku and H. deMeer. A Measurement Study on Signaling on Gnutella Overlay Networks. In *Fachtagung - Kommunikation in Verteilten Systemen (KiVS) 2003*, Leipzig, Germany, Feb. 2003.

[9] J. Schiller. *Mobile Communications*. Addison-Wesley, 2003.

[10] T. Hossfeld and K. Tutschku and F.-U. Andersen. Mapping of File-Sharing onto Mobile Environments: Feasibility and Performance of eDonkey with GPRS. In *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, New Orleans, LA, USA, Mar. 2005.

[11] G. Wearden. eDonkey pulls ahead in europe p2p race. `http://business2-cnet.com.com/2100-1025_3-5091230.html`.

[12] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak. Cache replacement policies for p2p file sharing protocols. *European Transactions on Telecommunications*, 15(6), 2004.