

University of Würzburg
Institute of Computer Science
Research Report Series

**Simulative Performance Evaluation of a
Mobile Peer-to-Peer File-Sharing System**

Tobias Hoßfeld, Kurt Tutschku¹, Frank-Uwe Andersen², and
Hermann de Meer, Jens Oberender³

Report No. 345

November 2004

¹ University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
{hossfeld, tutschku}@informatik.uni-wuerzburg.de

² SIEMENS AG
Siemensdamm 62, 13623 Berlin, Germany.
frank-uwe.andersen@siemens.com

³ University of Passau, Chair of Computer Networks and Computer Communications
Innstraße 33, 94032 Passau, Germany.
{demeer, oberender}@fmi.uni-passau.de

Simulative Performance Evaluation of a Mobile Peer-to-Peer File-Sharing System

**Tobias Hoßfeld, Kurt
Tutschku**

University of Würzburg
Department of Computer Science
Am Hubland, D-97074
Würzburg, Germany
{hossfeld, tutschku}@
informatik.
uni-wuerzburg.de

Frank-Uwe Andersen
SIEMENS AG

Siemensdamm 62, 13623 Berlin,
Germany.
frank-uwe.andersen@
siemens.com

**Hermann de Meer, Jens
Oberender**

University of Passau, Chair of
Computer Networks and
Computer Communications
Innstraße 33, 94032 Passau,
Germany.
{demeer, oberender}@
fmi.uni-passau.de

Abstract

Peer-to-Peer (P2P) file-sharing has become the killer application in the wired Internet and might also be highly attractive for mobile networks. In particular since UMTS operators are searching for new applications which do both: *a)* exploit the potential of the UMTS technology and *b)* motivate the user to adopt the new technology.

In this work we are investigating the performance of an eDonkey-based *mobile P2P file-sharing system* by means of time-dynamic simulation. Mobile networks differ from wireline networks by the limited capacity of the radio link and the mobility of the users. P2P networks, in contrast, are overlays which consider the transport network in an abstract way. In a mobile environment, the question arises, whether the abstraction can be maintained and what will be the performance impact if there is any. We will show in detail how the mobile access technology (GPRS or UMTS), the churn behavior of mobile users, the file size of mobile specific content, and special infrastructure entities, such as a cache peer, influences the performance of the suggested mobile P2P file-sharing service.

1 Introduction

P2P file-sharing has become the killer application in the wired Internet and has grown far more rapidly than web browsing in terms of traffic volume [1]. P2P file-sharing might also become 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.

Mobile networks differ from wireline networks mainly by the limited capacity of the radio link and the mobility of the users. P2P networks, in contrast, are overlays which run on top of a transport network. They consider the transport network only in an abstract way. P2P performance is therefore typically considered on overlay level only. In a mobile environment, the question arises, whether the abstraction can be maintained for underlying mobile transport networks, too, and whether there will be a performance impact.

P2P is trading its decentralized nature by increased communication traffic. The peers generate a large amount of signaling traffic for coordinating with each other [2, 3] as well as they produce

high volumes of payload traffic [4]. High application signaling traffic is considered to be too expensive in mobile networks and payload traffic should traverse the air interface only once on its way to the requesting peer. Our approach is to propose a *hybrid p2p architecture* which permits the operator to a) participate in service creation and service control, e.g. in order to keep the arising traffic in their own network, b) to offer value-added services, such as higher performance and presence information, while c) maintaining the characteristic of direct and efficient peer-to-peer interaction between the users, e.g. fast file swapping while minimizing the traffic on the user's uplink.

Currently a number of P2P file-sharing applications are available. Due to its high popularity among users [5], the eDonkey 2000 system¹ has been selected as a candidate for mobile P2P in this study. We assume that compatibility and popularity of an application is of greater importance for the selection than an easy implementation in mobile networks.

The results presented in this paper complement and complete previous work by the authors. The feasibility and performance measurements of a native eDonkey file-sharing service in mobile networks have been investigated for GPRS in [6] and for UMTS in [7]. A mobile P2P architecture overcoming the restrictions of today's existing P2P concepts and mobile networks was suggested in [8]. First performance values for one of the key components of the architecture, the *cache peer*, are also presented in [8] where the caching strategies are evaluated for it.

In this paper, we continue to investigate the proposed mobile P2P file-sharing architecture. We focus on the impact of the mobile environment on the performance and evaluate it by means of extensive simulation studies. Mobile specific influence factors are considered, like the temporal pattern of the mobile subscribers or the access type. We differentiate between the currently available GPRS classes and take a look on the performance improvement by UMTS. Furthermore, we analyze the impact of the file size of mobile specific content like ring tones, games, digital images, or mp3-audio files. The rest of this paper is organized as follows: Section 2 introduces the key elements of the proposed system architecture. We define a simulation model in Section 3 and present the numerical results in Section 4. Finally, Section 5 concludes this work.

2 Architecture for Mobile P2P Systems

The fundamental synchronization and control functions of P2P systems can be classified in two categories: *resource mediation* deals with locating resources while *resource control* grants and schedules priorities and access rights to shared resources.

The strength of pure P2P systems is its decentralization, which results from storing resources on end-user devices at the network edge. End-users of pure P2P systems gain full control on data resource access, which substantiates the high user-acceptance of these architectures. In contrast, the client/server approach offers high centralization in terms of resource control and mediation. Hybrid P2P applications like eDonkey utilize *weakly centralized* resource mediation with decentralized resource control [9].

Unlike in fixed networks, direct data communication in 2.5/3G mobile networks between user devices is more expensive. An IP-based data transfer between two mobile devices is always

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

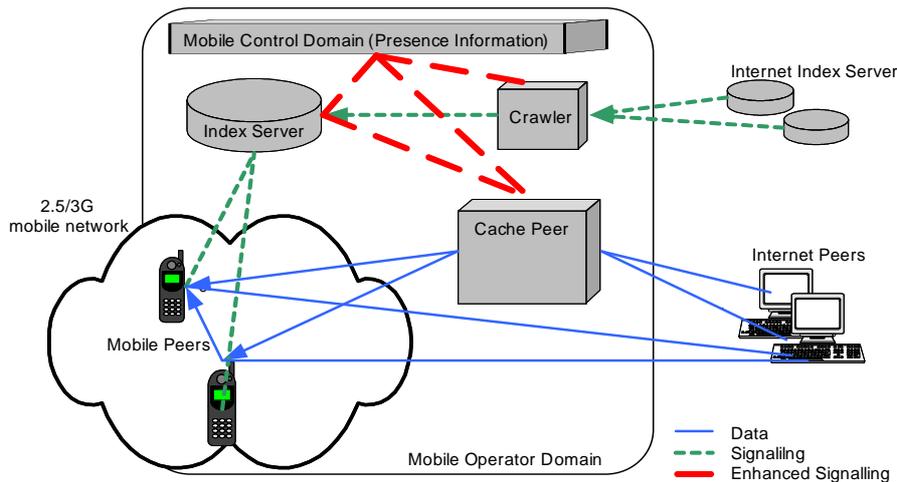


Figure 1: Mobile P2P Architecture Overview

routed through the Gateway GPRS Supporting Node (GGSN) and consumes twice as much network resources, even when both devices are in the same radio cell. Therefore, pure P2P systems are less optimal in 2.5/3G mobile environments, since they cause high amounts of data transfer in the mobile core network.

Our project aims towards integrating P2P technologies into mobile environments. We extended the *hybrid eDonkey* architecture focusing on cache integration. The mobile P2P architecture consists of three components, cf. Fig. 1: a modified *index server* for mediation, a *cache peer* for popular files and a *crawling peer*, which supports mobile peers searching the global community.

The proposed mobile P2P architecture provides file caching in the network. In a mobile environment the resource usage of a peer-to-peer communication is as expensive as two peer-to-cache-peer communications due to passing the air interface two times. Since the buffer of the cache peer is limited, a caching strategy must be chosen [8]. Because of the index server as a centralized mediation entity, statistics about currently accessed files can be collected for the caching strategy.

The index server mediates resources, in this case the shared files, using three operations: registration of shared files, searching for filename patterns and listing of peers that currently share a file. At a *source request* by a peer, the index server returns providing peers for this file. The index server may favor the cache peer. Resources that are already cached can be downloaded from the cache peer first. In any other case, the index server returns the full list of mobile devices that share the file.

Information on recently accessed resources can be aggregated into a list of popular files. The cache peer compares this list to the currently cached items, and then fetches missing resources. For downloading of files, the cache peer uses the same eDonkey mechanism as an arbitrary peer. After the download, the index server is informed about the newly shared file. By introducing the operator run cache peer, the resource control was shifted from the user back to the network. As

a result, the network usage of P2P traffic is efficiently reduced as popular files are downloaded from the cache peer in the network.

The crawling peer supports the integration of the mobile peers into the global community. It searches files on behalf of mobile peers at any other peers and reduces in this way the amount of signalling traffic for mobile peers.

3 Simulation Model

The mobile P2P simulation model consists of the peer model, the resource model, and the network model, cf. Fig. 2. The latter describes the restrictions of the P2P system due the mobile network architecture. The dashed lines in Fig. 2 indicate the interaction between the different model components which are explained in the following. The simulation parameters are summarized in Table 3.

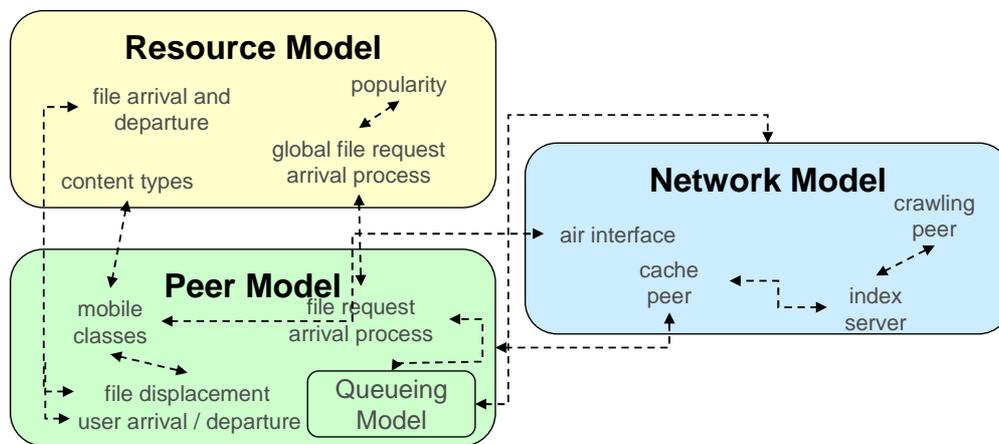


Figure 2: Components of the mobile P2P simulation model

3.1 Peer and Resource Model

The resource model describes the provided files and their popularity determining the file request arrival rate. For each file f the request arrivals follow a Poisson process with rate λ_f . In the considered P2P network, there is a large number N_{files} of files available. Typically, only a small number N_{pop} of very popular files generates a huge amount of traffic [10]. Each peer is initialized with a random contingent N_{pop}^{init} of popular files according to a binomial distribution which is described by the maximal and the average number of popular files. The remaining memory capacity is filled with unpopular files, i.e. the peer utilizes completely its capacity. In our simulation the storage capacity of a mobile peer is either 4MB, 8MB, 64MB, or 128MB and is selected with equal probability for a peer. The storage capacity determines the preferred content types of a peer. If a newly requested file exceeds this capacity, the oldest files which are shared longest are deleted (FIFO) until sufficient memory is available for storing the new file.

We assume that there are mobile specific content types, like ring tones (midi or mp3) or digital images, which are shared in mobile P2P. The file sizes for different content types were measured

Table 1: File sizes and probabilities for mobile P2P contents

| | ring tone | game | image | mp3-audio |
|----------------|-----------|---------|----------|-----------|
| mean [kB] | 8.5762 | 37.9288 | 420.2075 | 4829.3306 |
| std. dev. [kB] | 9.3479 | 26.5833 | 21.3963 | 2305.5083 |
| probability | 33.0% | 33.0% | 27.7% | 6.8% |

in the eDonkey network. We fitted the distribution function for the file size with a lognormal distribution which we applied in the simulation. Table 1 shows the measured file sizes and the assumed file appearance probabilities resulting from the conditional probabilities that a peer with a certain storage capacity shares a mobile specific content type.

In addition to the mobile peers, we also consider *internet peers*. The main differences of these peers are the access type, the maximum number of upload connections, and that the internet peers never leave the eDonkey network. In our simulations we used a ratio of mobile to internet peers of 2 : 1.

In order to reflect the highly fluctuating connection status of mobile peers, we describe their participation in the overlay by an ON/OFF-process. The ON period and OFF period are determined by exponential distributions with means L_{ON} and L_{OFF} . Therefore, the transition rates between these two states are $\frac{1}{L_{ON}}$ and $\frac{1}{L_{OFF}}$. During the ON period, peers participate in the P2P network by providing their own files and requesting other files. With probability p_{new} , a peer shares a new file upon entering the ON state. Since we choose L_{ON} and L_{OFF} to be equal in our simulations, we consider L_{OFF} as the *churn time*.

The eDonkey application maintains an *upload list*, reflecting the simultaneously served peers, and a *waiting list* containing all requesting peers. A peer always utilizes its full capacity in uplink and downlink direction. The uplink bandwidth of a providing peer is equally split among the served peers in the upload list. If the uploading peer cannot utilize the offered bandwidth by the downloading peer, the download peer will fairly share remaining bandwidth between the other uploading peers.

Due to the limited air interface capacity in mobile networks, the effect of bottlenecks are expected to occur more often in both directions than in wireline environments, i.e. on the uplink and on the downlink.

The upload list at a providing peer is limited resulting in a minimal assured download bandwidth at a served peer. The waiting list is unlimited. A newly arriving file request joins the end of the waiting list. It has to be noted that in eDonkey, a file is structured into chunks of 9.5MB and each chunk is downloaded in smaller pieces of 560kB, denoted as download units. After downloading a download unit, the peer also rejoins the end of the waiting list. Immediately after downloading a whole chunk, the peer is registered as a source at the index server and the peer can act as a provider of the data in the chunk.

3.2 Mobile P2P Network Model

The mobile P2P network model includes the description of the access technology which is either GPRS or UMTS. In case of GPRS, the *mobile class* of a peer is characterized by the number of available uplink and downlink slots, cf. Table 2. The mobile classes are assigned to the peers

Table 2: Access types of the peers in the P2P network

| access type of peer | upload bandwidth | download bandwidth | max. upload connections |
|--------------------------------------|------------------|--------------------|-------------------------|
| GPRS class $n \in \{1, 2, 3, 4\}$ | 12 kbps | $n \cdot 12$ kbps | 4 |
| UMTS | 64 kbps | 384 kbps | 4 |
| internet | 128 kbps | 768 kbps | 10 |
| cache | 4 Gbps | 4 Gbps | 400 |

with equal probability. In case of UMTS, a mobile peer has an uplink capacity of 64 kbps and a downlink capacity of 384 kbps, cf. Table 2.

In the eDonkey network, a peer has to be connected to an index server for participating in the network. Thus, the index server knows always the location of all files in the network and immediately notices when a peer goes online. It is assumed that the index server discovers instantaneously when a peer goes offline. The index server returns at most 200 sources to the requesting peer. This value is coherent to the original eDonkey protocol. The returned sources are selected randomly from all known sources at the index server.

If the cache peer shares a file requested by a peer, the cache peer is always returned. In addition, it can be selected in the model that the cache peer is the only returned source and all other sources are hidden, see also Section 4.1.

The cache peer is assumed to be attached to the network with a 4 Gbps link. This value has been selected such that any bottleneck effect at the cache peer can be neglected. The number of parallel upload connections of the cache peer is limited to 400. The cache peer uses the IMU caching strategy [8].

eDonkey peers are able to acquire sources for downloads from other eDonkey nodes. For well shared files every 10 minutes a random peer in the source list of a particular on-going download is asked for its sources. Only sources for missing parts are acquired. In order to reflect the source exchange mechanism, we have implemented periodical source requests. This means that every 10 minutes a downloading peer requests the index server for new sources.

4 Numerical Results

The performance values for the mobile P2P system are considered from the *user's perspective* and the *operator's point of view*. The operator wants its cache peer highly utilized since this facilitates its value-added file-sharing service. The utilization of the cache peer is expressed by the byte-hit-rate which is the ratio of the transmitted download volume of the cache peer and the totally transmitted download volume within the network. In addition, the operator has to consider the *kion-rate (keep in own network-rate)* which is the ratio of the file-swapping traffic volume within the operator's domain to the file-swapping traffic volume within the entire P2P network. The kion-rate describes the efficiency of keeping the traffic in the operator's own network.

For characterizing the performance from user's perspective, the download time is defined

Table 3: Simulation scenario parameters

| parameter | description | standard value |
|-----------------------------------|--|-----------------------------|
| $\frac{N_{pop}}{N_{files}}$ | ratio of popular files | $\frac{20}{50000} = 0.04\%$ |
| N_{pop}^{init} | initial number of popular files | $binom(10, 2)$ |
| λ_{pop} | request rate for each popular file | 4.50 h^{-1} |
| λ_{unpop} | request rate for all unpopular files | 7.13 h^{-1} |
| $\frac{N_{mobile}}{N_{internet}}$ | ratio between mobile and internet peers | $\frac{666}{333} = 2 : 1$ |
| p_{new} | probability to share a new file | 0.10% |
| L_{ON}, L_{OFF} | churn time | 30 min |
| $N_{sources}$ | maximal number of returned sources by the index server | 200 |

as the period of time from the file request at the index server until the successful download including the OFF phases of the peer. The user's objective is to download a file in minimal time with minimal upload volume. If the download time is too large, the peer will cancel its file requests. In this context, we define a *reliable P2P file-sharing service*, if the peer is assured by system design to download a file within acceptable time, e.g. 40min, with some probability, e.g. 90%.

For the performance evaluation of the mobile P2P system, we implemented the model in Section 3 and investigate different scenarios. Table 3 depicts the standard parameters applied for all scenarios. If other values have been used, they will be explicitly stated.

4.1 Performance of the Cache Peer

The performance values of the cache peer with regard to the operator's point of view are given in Figure 3 and Table 4(a). Figure 3 shows the observed download data volume in a scenario when only a one file of 3MB size is shared. The file has an initial diffusion of 0.1% among all peers. In contrast to the other simulations, this scenario comprises 6667 mobile peers and 3333 internet peers.

Figure 3(a) depicts the case when the cache peer is returned by the index server as an ordinary source among other sources. In this scenario, however, the cache is not as efficient as expected. Figure 3(a) shows that the transfer volume of the cache peer (green line) increases as soon as the cache has completed the download. The upload volume of the mobile peer (blue line) reaches immediately very high values and stays on this level even after the cache peers serves requests. The transfer volume of the internet peers (red line) increases immediately but decreases then slowly. With each mobile peer having successfully finished a download, the number of sharing

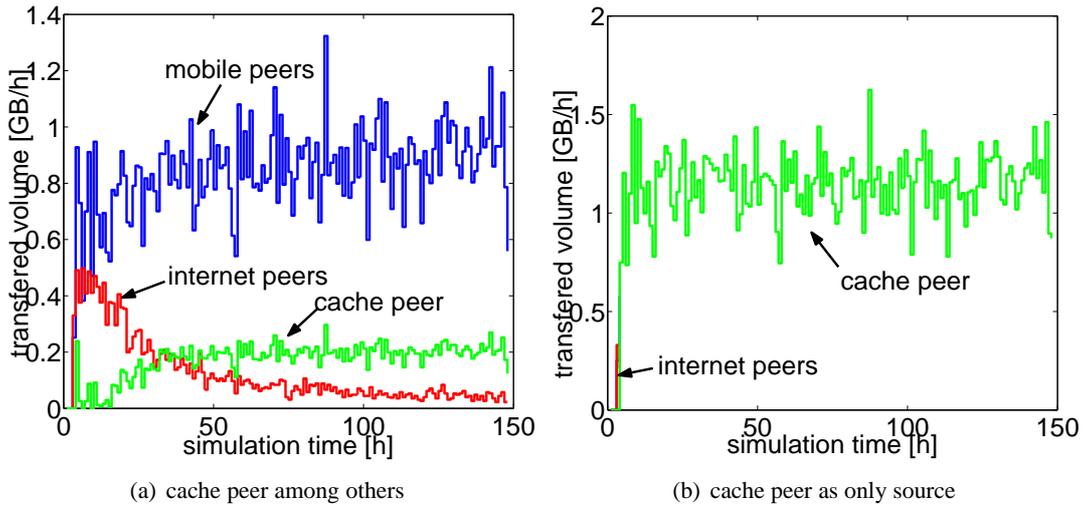


Figure 3: Download volume for a 3 MB file

peers adds, and the probability of a mobile being return as source increases. As a result, the data volume transmitted from the internet peers decreases since the number of mobile peers dominates the internet peers. The lower transfer volume of the cache peer results from the bottleneck in the downlink of the mobile peers. The downlink bandwidth is equally shared as long as the uploading peer can provide sufficient data. With too many active downlink connections at a specific peer, the cache peer is not able to play out its high bandwidth upload capacity. In this way, the cache peer is blocked from improving the service performance.

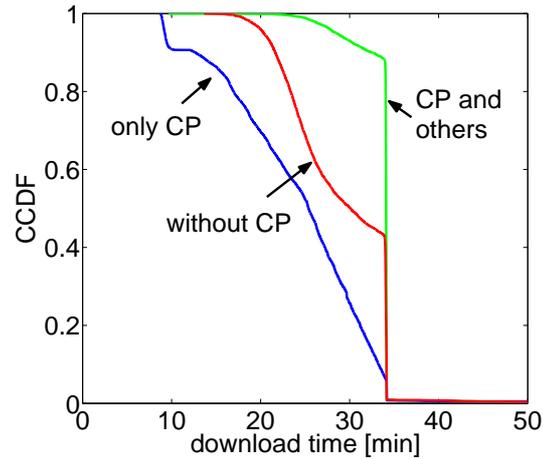
Figure 3(b) depicts the case when the cache peer is returned as the only source. In this scenario, the mobile peers and the internet peers provide the file only at the very beginning of the considered scenario. As soon as the the cache peer has downloaded the file, it serves all further download requests. This behavior demonstrates the real capability of the cache peer. The cache peer controls the download requests for popular files while the architecture still permits peer-to-peer sharing of less popular files.

Table 4(a) shows the numerical values of the cache peer efficiency from provider's point of view. Without using the cache peer, the kion-rate reaches already a high level of 89.0% due the higher number of mobile peers than internet peers. A further increase to almost 100% is achievable if the cache peer is returned as the single source by the index server. The cache peer is serving then all request after it has downloaded the file. In contrast, the byte-hit-rate is poor when the cache peer is returned among other sources. The mobile peers still serve requests for popular files. The byte-hit-rate improves to 99.2%, if the cache peer is returned as the single source.

The user perspective on the cache peer (CP) performance is depicted in Figure 4(b). It shows the complementary cumulative distribution function (CCDF) of the download time for the three scenarios: without using the cache peer (red curve, reference scenario), the cache peer is returned among other sources (green curve), and the cache peer is returned as the only source (blue

| appli- cation | kion- rate[%] | byte-hit- rate[%] |
|---------------------------|------------------|----------------------|
| without c.p. | 89.0 | - |
| among other sources | 89.1 | 15.3 |
| as single source | 99.7 | 99.2 |

(a) byte-hit-rate and kion-rate



(b) Download time

Figure 4: Download of a 3MB file for different applications of the cache peer

curve). Figure 4(b) shows that a naive use of the cache peer, i.e. it is returned among other sources, brings no improvement. The download time does even increase when compared with the reference scenario. The reason is that the waiting list for download requests at mobile sources is nearly empty if the cache peer is used. The index server returns also mobile source and the requesting peer is immediately served. The download time is determined by the uplink of the mobile source. This bottleneck explains the rough decay in the CCDF in Figure 4(b) at about 34 minutes. In this case, the complete file is downloaded from mobile sources. The increase in the weight in CCDF before the decay is a result of the cache peer being slowed down by other mobile peers. The returning of the cache peer as the single source, however, leads to the desired effect of smaller download times, cf. the blue line in Figure 4(b).

Impact on Protocol Design

Typically, the design of P2P mechanisms assumes that all peers are equal. The above presented performance values show that not all peer are equal in the considered mobile P2P file-sharing architecture. The mobile peers, the internet peers, and the cache peer, differ significantly in their throughput values. Neglecting this characteristic may lead to decreased performance. The performance reduction can be overcome in two ways. The index server should return the cache peers as the only source if one is not allowed to modify the original eDonkey protocol. Additional performance can be achieved when the index server also provides a list with the upload access speeds of the providing peers. However, this would require a modification of the eDonkey protocol and might limit the compatibility of the solution. With the upload access speed known, the downloading peer can decide where to transfer data from. With such an extension of the protocol, the peers would automatically prefer the cache peer.

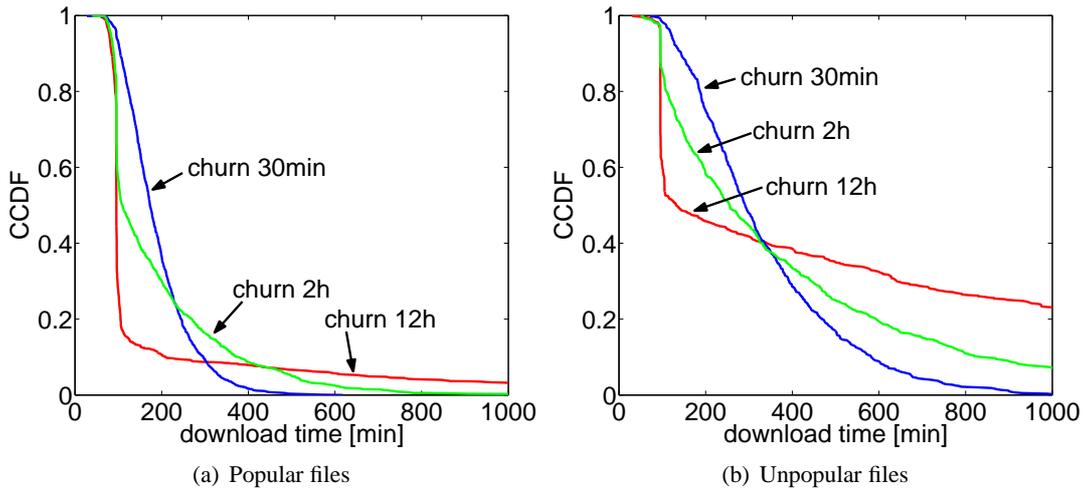


Figure 5: Download time for different churn times

4.2 Impact of Mobility and Temporal Behavior of Mobile Subscribers

Mobile peers are assumed to have a higher churn time, i. e. they change their ON/OFF state more frequently than wirelined peers. This significant characteristic of mobile peers is a result of several reasons such as mobility, roaming, limited battery life time, time-based charging, and area dependent coverage by the operator. We consider three different churn times: 30min, 2h, and 12h. While the first two values fit well mobile subscribers, the latter one describes wireline-like behavior.

Figure 5 depicts the dependence of download time CCDF on the peer's temporal behavior. In the investigated scenarios, the index server returns the cache peer as the only source for popular files. Figure 5(a) shows the CCDF of the download time for popular files. The mobile peers with the longest churn time of 12h (red curve in Figure 5(a)) have the smallest download times. The more the average churn time decreases in Figure 5(a), i.e. 2h (green line) and 30min (blue line), the more the download time increases. In addition, the CCDF of the download time shows exponential decay. This behavior results directly from the exponentially distributed OFF periods of the peers and explains the very long download times.

Figure 5(b) illustrates the CCDF of the download time for unpopular files with respect to the different churn times. One observes the effect of the churn time being even more significant for files with a lower popularity than for more popular files. This result shows the cache peer not only reducing the amount of data transmitted on the air interface but also reducing the impact of the mobility on the service performance. The cache peer increases the reliability of the P2P file-sharing service by increasing the probability that a download is finished within 300 minutes.

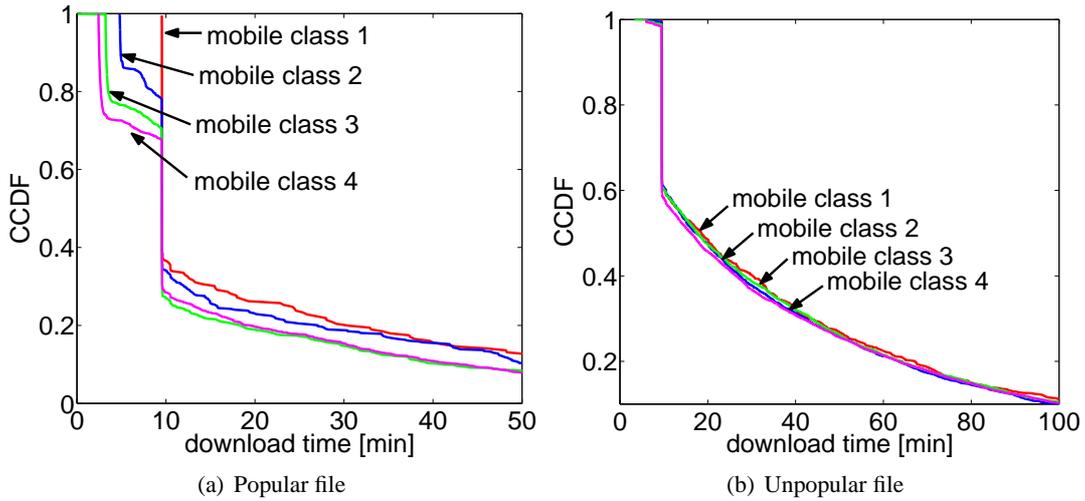


Figure 6: Download of an image for different GPRS classes

4.3 General Packet Radio Service (GPRS)

The access type of the mobile subscribers is assumed to have a high impact on the system performance. As discussed above, the downlink is the bottleneck for the download of popular files since the cache peer has a significant higher upload bandwidth than the downlink bandwidth of the mobile peers. The uplink bandwidth of the sharing peers limits the download bandwidth for unpopular files which are not cached.

In this section we focus on scenarios where the cache peer is returned as single source only. The mean length of the ON/OFF period is 30 minutes. We consider the different mobile classes for GPRS as defined in Table 2.

Figure 6(a) depicts the CCDF of the download time for the four mobile GPRS classes for a popular, cached file of size 858 kB (cf. image category in Table 1). We see that a higher mobile class, i.e. high downlink capacity, results in a shorter download time, as the download time is determined by the downlink bandwidth of the file requesting peer.

The strong decay located at 10 minutes is the limit for the minimal download time of a file of size 858 kB by a GPRS class 1 peer. For the higher mobile classes the minimal download time is correspondingly smaller. If a mobile peer downloads n files in parallel, the download bandwidth of this peer is equally split for each download and results in higher download times. However, the strong decay at 10 minutes for higher mobile classes results from the fact that we have forced the cache peer to become active after 6 days in this scenario. This means that the mobile peers download the files within this period from others mobiles, which explains that the decay is exactly at the limit of class 1 mobiles.

The decay of the CCDF after the minimal class 1 bound shows an exponential shape due to the OFF states of mobile peers being exponentially distributed. For unpopular files this exponential decay is even stronger perceived. Figure 6(b) shows the CCDF for an unpopular file of size 858 kB. No remarkable differences can be observed between the GPRS mobile classes because

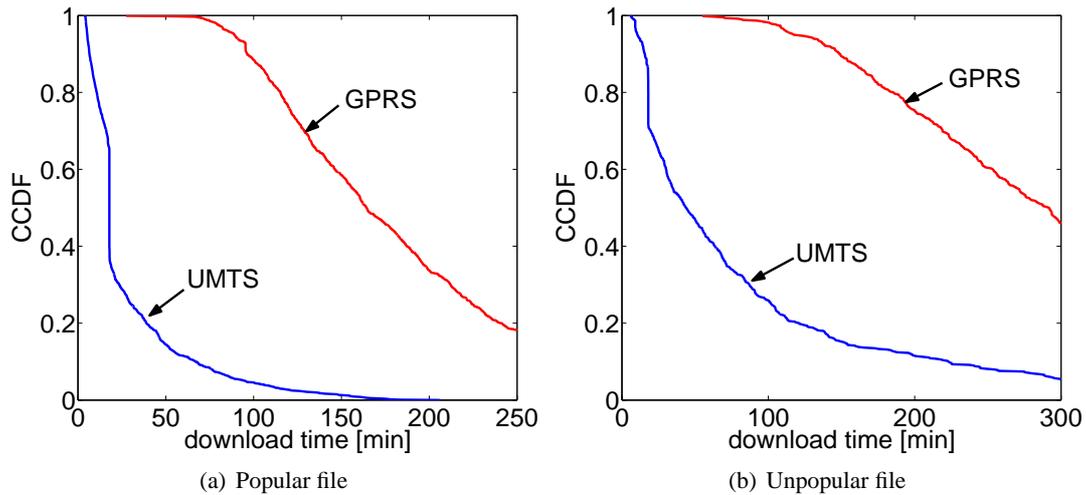


Figure 7: Download of mp3-audio files with GPRS and UMTS

the download time is determined by the upload bandwidth of the sharing peers. It has to be noted again that for unpopular files the uplink of the providing peers is the bottleneck. Therefore only the GPRS mobile class 1 limit is visible in Figure 6(b).

The above shown results demonstrate that the GPRS mobile classes influence significantly the performance of the P2P file-sharing service. These results show that the performance on the overlay level cannot be separated from the access network.

4.4 Improvement by UMTS

In the following we consider that all mobile peers use UMTS as access network. We compare the performance results with the previous case that all mobiles use GPRS. The simulation parameters for the UMTS scenario are the same like in the GPRS scenario in Section 4.3.

Figure 7 shows the CCDF of the download time for popular and unpopular mp3-files of 8 MB. The UMTS subscribers get quite reasonable performance values since the download time exceeds 1 hour only with a small probability. On the other hand, the GPRS subscribers have a much higher download time and the shape of the curve is completely different to the CCDF for UMTS. It seems that there exists a minimal required upload/download bandwidth of the peers for a given file size in order to retrieve a *reliable* P2P system. The shape of the blue curve in Figure 7 is characteristic for the CCDF of the download time in a reliable system, while the red one illustrates the behavior for unreliable systems. This effect becomes even more obvious for unpopular files which are not cached. In Section 4.5 we observe the same indicator of reliable/unreliable systems for different content types, i.e. different file sizes.

The above presented results show that GPRS should not be used for large mp3-files in the context of mobile P2P file-sharing. However, UMTS seems to be a good candidate for enabling mobile P2P file-sharing even for larger size contents. Furthermore, we recognize, due the difference between GPRS and UMTS, the interaction between overlay and access network as having

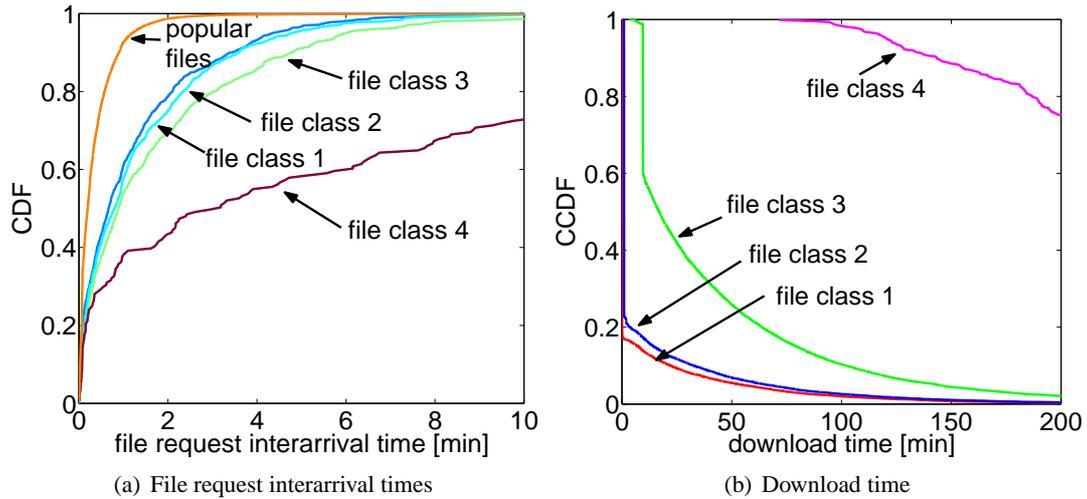


Figure 8: Download of different mobile specific content types

significant impact on the reliability of the mobile P2P file-sharing system.

4.5 Influence of the Content Type

We assume that the file size of typical mobile content will also determine the perceived performance of the mobile P2P file-sharing system. The different file classes considered in our study are summarized in Table 1. The cumulative distribution function (CDF) for the file request interarrival time indicates the resulting load in the network and is plotted in Figure 8(a).

Figure 8 shows the CCDF of the download time for unpopular files. The mobile peers are assumed to be GPRS subscribers of class 4 and the churn time is 30 min. As a result, we observe a similar behavior as in Section 4.4. The system is reliable for small files (class 1 - 3), but larger files of class 4 cannot be supported by the GPRS network. This means there exists a maximal file size which can be exchanged within a acceptable time frame. Otherwise, a large number of downloads cannot be successfully finished, since the download time for a class 4 file exceeds a value of 300 minutes with a probability of 80% for example. However, we see that small files can efficiently be exchanged over the GPRS network. This means that GPRS is useful only in an "Instant Messaging"-mode, i.e. the files are exchanged a single or very few messages. For true P2P file-sharing, UMTS is required. Similar results are observed for popular files.

5 Conclusion and Outlook

In this work, we have investigated the performance of an eDonkey-based *mobile P2P file-sharing system* by means of time-dynamic simulation. The results show that the mobile environment and subscribers have significant impact on the P2P file-sharing service. In particular, the access network and the churn behavior determine the performance of the system. Efficient mobile P2P

file-sharing seems not to be feasible in GPRS environment whereas UMTS is a good candidate for *true* P2P file-swapping. Short churn times of mobile peers, representing high mobility, degrade the service performance strongly. This effect can be reduced if a *cache peer* is used. The application of the cache peer adds a new characteristic to the system. If a cache peer is applied the downlink at the downloading peer will always be the limiting factor, or, bottleneck. If the cache peer is not used or the files are not cached, i.e. they are not popular enough, then the uplink of the providing peer is the bottleneck. The results for mobile specific content types underline that GPRS is useful in an "Instant Messaging"-mode, i.e. the files are exchanged in a single message or within very few messages. The performance results for different content sizes indicate that there is an optimal transmission segment size exists which has to be selected according to the typical content size and the type of mobile access network. The relations among these factors will be investigated in further studies.

Acknowledgment

The authors would like to thank Daniel Schlosser for implementing the simulation.

References

- [1] 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*, vol. 15, no. 6, 2004.
- [2] K. Tutschku, "A measurement-based traffic profile of the eDonkey filesharing service," in *5th Passive and Active Measurement Workshop (PAM2004)*, (Antibes Juan-les-Pins, France), Apr. 2004.
- [3] K. Tutschku and H. deMeer, "A measurement study on signaling on gnutella overlay networks," in *Kommunikation in Verteilten Systemen (KiVS) 2003*, (Leipzig, Germany), Feb. 2003.
- [4] T. Karagiannis, A. Broido, N. Brownlee, and K. Claffy, "Is p2p dying or just hiding?," in *Proceeding of IEEE Globecom 2004*, (Dallas, USA), Nov./Dec. 2004.
- [5] G. Wearden, "eDonkey pulls ahead in europe P2P race." http://business2-cnet.com.com/2100-1025_3-5091230.html.
- [6] T. Hoßfeld, K. Tutschku, and F.-U. Andersen, "Mapping of file-sharing onto mobile environments: Feasibility and performance of edonkey with GPRS," in *WCNC 2005*, (New Orleans, LA USA), 3 2005.
- [7] T. Hoßfeld, K. Tutschku, and F.-U. Andersen, "Mapping of file-sharing onto mobile environments: Enhancement by UMTS," in *Mobile Peer-to-Peer Computing MP2P, in conjunction with PerCom*, (Kauai Island, Hawaii), 3 2005.

- [8] F.-U. Andersen, H. de Meer, I. Dedinski, T. Hoßfeld, C. Kappler, A. Mäder, J. Oberender, and K. Tutschku, “An architecture concept for mobile P2P file sharing services (extended version),” Tech. Rep. 344, University of Würzburg, Nov. 2004.
- [9] F.-U. Andersen, H. de Meer, I. Dedinski, C. Kappler, A. Mäder, 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.
- [10] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak, “Cache replacement policies for P2P file sharing protocols,” *European Transactions on Telecommunications*, vol. 15, no. 6, 2004.