23. Traffic Characteristics and Performance Evaluation of Peer-to-Peer Systems

Kurt Tutschku, Phuoc Tran-Gia (University of Würzburg)

23.1 Introduction

Peer-to-Peer services have become the main source of traffic in the Internet and are even challenging the World Wide Web (WWW) in popularity. Backbone operators and Internet Service Providers (ISP) consistently report Peer-to-Peer-type traffic volumes exceeding 50 % of the total traffic in their networks [42, 337, 372, 556], sometimes even reaching 80 % at nonpeak times [39, 236], see also chapter 22.

Peer-to-Peer services are highly lucrative due to their simple administration, their high scalability, their apparent robustness, and easy deployment. The use of a distributed, self-organizing Peer-to-Peer software might reduce capital and operational expenditures (CAPEX and OPEX) of service operators since fewer entities have to be installed and operated. In a commercial context, high performance Peer-to-Peer means that these services meet tight statistic performance bounds; for *carrier gradeness* this bound is typically 99.999 %, the so called "five nines" concept. Before Peer-to-Peer services or Peer-to-Peer-based algorithms might be released in a production environment, it has to be evaluated whether these Peer-to-Peer-based solutions meet these requirements or not.

This aim of this chapter is to present selected characteristics of Peer-to-Peer traffic and discuss their impact on networks. In addition, the chapter will outline what performance can be expected from Peer-to-Peer-based algorithms and which factors influence Peer-to-Peer performance. First, this chapter discusses in Section 23.2 the relationship of basic Peer-to-Peer functions with performance. Section 23.3 is dedicated to the traffic patterns of popular Peer-to-Peer services. In particular, the characteristics of Gnutella overlays (Section 23.3.1) and of the eDonkey file sharing application in wireline and wireless networks (Section 23.3.2) are investigated. The efficiency of a Chord-like resource mediation algorithm is discussed in Section 23.4. Section 23.5 is devoted to the performance of exchanging resources in a mobile Peer-to-Peer architecture.

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 383-397, 2005.

23.2 A Concept for Peer-to-Peer Performance

A comprehensive description of the performance of Peer-to-Peer systems is a challenging task. The term "Peer-to-Peer" describes not a single architecture but subsumes a rather huge variety of architectures and applications. A single evaluation metric for Peer-to-Peer services is rather impossible. Peer-to-Peer architectures and Peer-to-Peer algorithms have to be evaluated according to the task they accomplish.

All Peer-to-Peer systems have in common that they are highly distributed application architectures where functional equal entities (*peers*) voluntarily share resources. In order to participate in the resource exchange, Peer-to-Peer systems support two fundamental coordination functions: *a) resource mediation* mechanisms, i. e. functions to search and locate resources or entities, and *b) resource access control* mechanisms, i. e. functions to permit, prioritize, and schedule the access to resources. In addition, since Peer-to-Peer is a networked application architecture, the efficiency and the performance of a Peer-to-Peer system has to be evaluated on the network layer (e. g. by describing the load imposed by a Peer-to-Peer application on the network), as well as on the applications layer (e. g. by considering the time to locate a resource).

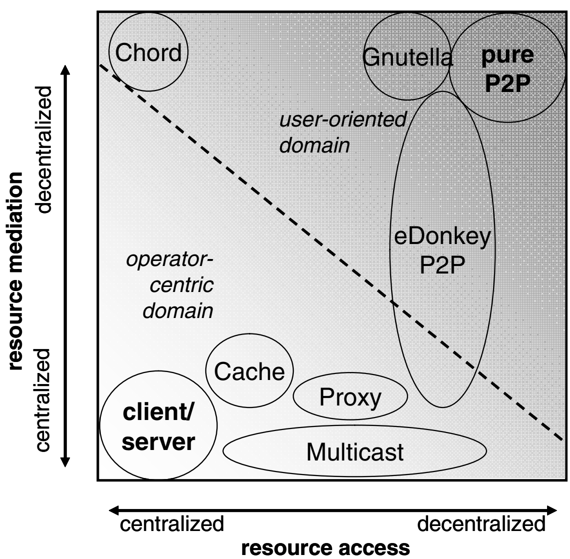


Fig. 23.1: Cartography of Peer-to-Peer applications and content distribution architectures

Figure 23.1 depicts a two-dimensional cartography for comparing Peer-to-Peer systems by their architectural characteristics with other well-established information dissemination mechanisms. The basic Peer-to-Peer control functions *(resource mediation/resource access control)* form the Cartesian space in Figure 23.1. The degree of distribution *(centralization/decentralization)* is used as the range of the axes. The cartography visualizes the architectural options of operators and users for providing information distribution services ("operator-centric" or "user-centric" architectures)[1]. The cartography provides an initial guideline of how to choose the components of a Peer-to-Peer architecture under given application requirements. A specific selection of a Peer-to-Peer-based algorithm will be based on its performance.

The overall performance of Peer-to-Peer services is determined by the combined performance of the two basic control functions. Since the control functions solve different tasks, the algorithms have to be evaluated by separate performance metrics. The *resource mediation functions*, for example, can be evaluated by:

- the needed time to locate a resource (cf. Section 23.4)
- the probability to locate a certain resource
- the amount of communication needed to locate a resource (cf. 23.3.1)

The metrics for *resource access control* are more user-oriented and may comprise:

- the time needed to exchange a resource (cf. Section 23.5)
- the throughput obtained during the exchange of a resource (cf. Section 23.3.2)

The range of the axes of the cartography indicates another constraint for the evaluation of Peer-to-Peer. Decentralization includes scalability, thus the performance of Peer-to-Peer algorithms has to be considered with respect to the *number of entities* participating in the system. Another feature of Peer-to-Peer is the autonomy of the nodes, i. e. the peers may join or leave the system arbitrarily. This leads to the requirements to evaluate Peer-to-Peer algorithms with respect to the *stochastic on-line behavior*, which is summarized under the term "churn", cf. also Section 23.5.

A key reason for the success of Peer-to-Peer systems is their use of application-specific overlays. Peer-to-Peer overlays, however, show a high variability due to the churn behavior of the peers. The *stability of the overlay*, e. g. the life time of overlay connections (cf. Section 23.3.1), and the *consistency of the overlay*, e. g. the probability that an overlay splits, are Peer-to-Peer-specific performance metrics for overlays.

[1] A detailed discussion of Figure 23.1 is provided in [27]

23.3 Traffic Characteristics of Peer-to-Peer-Systems

23.3.1 Gnutella

Gnutella was one of the first successful Peer-to-Peer file sharing applications [335] and sparked largely the wide spread interest in Peer-to-Peer due to its *pure Peer-to-Peer architecture*. The Gnutella service forms an application-specific overlay of Internet accessible hosts running Gnutella-speaking applications like `LimeWire` [388] or `Bearshare` [65]. In Gnutella, the overlay is used for locating files and for finding other peers; the later in order to maintain the integrity of the overlay. The initial version of Gnutella [126] uses a simple flooding protocol combined with a back-tracking mechanisms for locating the resources (files or hosts) in the overlay. While the qualitative evaluation has revealed that Gnutella suffers from scalability problems [518], little is known of quantitative results on the traffic and the dynamics in Gnutella overlays. In particular, time scale and variability of the number of virtual connections have to be characterized [601].

Measurements at an unrestricted Gnutella client have been carried out in March 2002 at the University of Würzburg. The observations (cf. Figure 23.2) reveal that even without sharing files, a Gnutella client is consuming tremendous high amounts of bandwidth for locating resources (files or hosts), reaching up in the order of tens of Mbps.

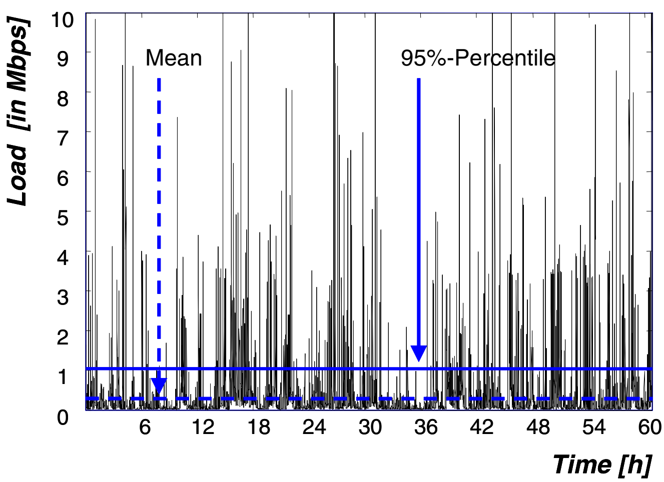


Fig. 23.2: Sum of signaling traffic load

In addition, Figure 23.2 shows that the traffic in Gnutella overlays varies strongly over short timescales. This is mainly due to the use of flooding protocols in Gnutella.

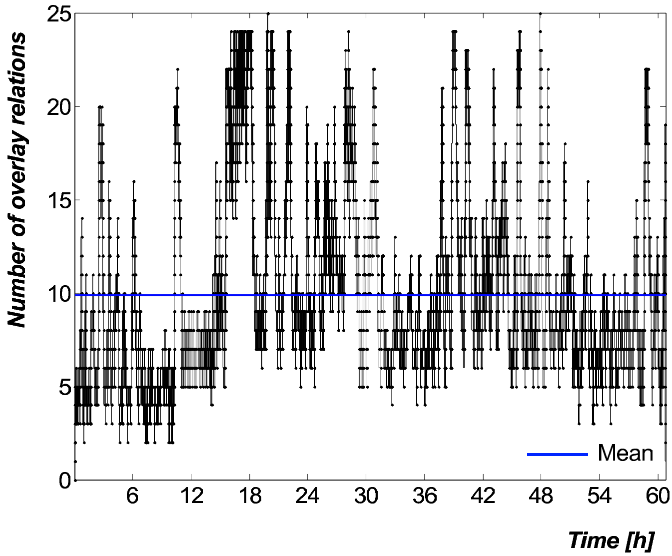


Fig. 23.3: Number of simultaneous overlay relations

In Gnutella, a peer tries to maintain a certain, pre-configured number of overlay connections. Due to the churn behavior of the peer the number of parallel maintained overlay connections can vary significantly, cf. Figure 23.3.

The investigation of the overlay connection holding time in Gnutella showed that the distribution typically has bi-modal characteristic, cf. Figure 23.4.

The modes correspond to a “short” state, where typically host information is transmitted, and to a “stable” mode, where mainly content queries are exchanged. The modes identify the time scales on which a dynamic and adaptive management of Peer-to-Peer overlays and Peer-to-Peer services is of advantage or needed.

23.3.2 eDonkey

The eDonkey Peer-to-Peer filesharing service[2] [410, 589] continues to be one of the most popular file swapping applications in the Internet [616]. The eDonkey system is typically used for exchanging very large files like audio/video CDs or even DVD images, and possesses a hybrid Peer-to-Peer architecture with distinct servers and clients. The eDonkey system makes use of the *multi source download (MSD)* feature, which permits the simultaneous transmis-

[2] This chapter subsumes eDonkey2000 and all its derivatives by the single term eDonkey.

sion of file chunks to a downloading peer. The traffic profile [600] shows that resource mediation traffic (also denoted as "signaling" traffic) and download traffic have significantly different characteristics. Figure 23.5 depicts a scatter plot describing graphically the correlation of the TCP holding time and the size of eDonkey flows.

Each dot in the scatter plot represents an observed eDonkey flow. The brighter dots are identified download flows, the dark dots represent non-download connections. The scatter plot shows that almost all identified download flows are within the same region. In turn, the non-download flows are in an disjunct region of the plot. This graph reveals that download and non-download flows have significantly different characteristics. A Peer-to-Peer traffic model has to distinguish between both types of traffic.

The differences between the types of traffic are underlined in Figure 23.6 and Figure 23.7. The complementary cumulative distribution function (CCDF) of the flow size is depicted in Figure 23.6. Part (a) of Figure 23.6 shows that the download flow size decreases stronger than linear in the log/log plot. That means that the flow sizes don't show a strong "heavy tailed" feature.

An approximation of the observed data with a lognormal distribution achieves a good estimate. The reduced strength of the heavy tail feature is not expected, but can be explained: the download flows are limited due to the segmentation of files into chunks and due to the application of the multiple source download principle. This observation gives evidence that the expected "mice and elephant" phenomenon [479, 73] in eDonkey traffic is not as severe as expected.

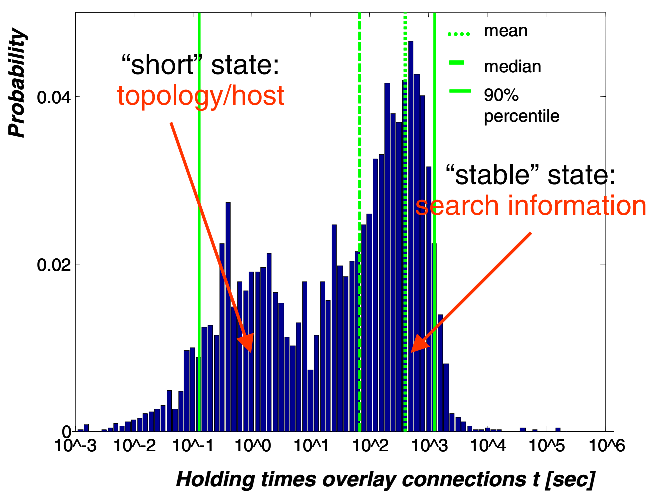


Fig. 23.4: Gnutella overlay connection holding time distribution

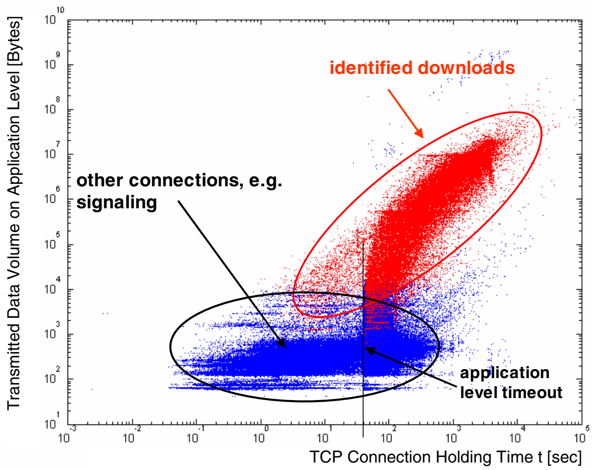


Fig. 23.5: Correlation of eDonkey TCP holding time and flow size

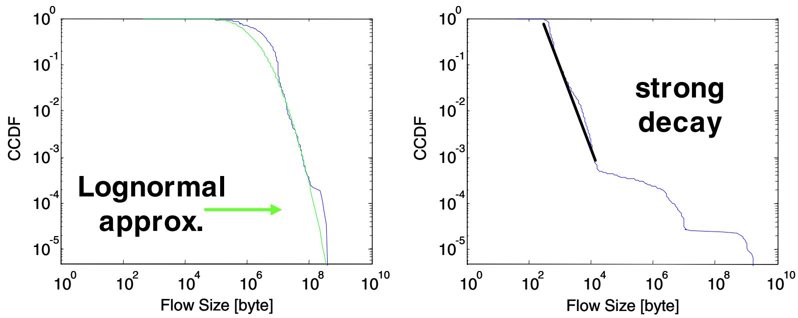
Part (b) of Figure 23.6 depicts the size of non-download flows. The probability that a flow is larger than a given value decreases almost exponentially until a limit (approx. 14 Kbytes). Beyond this limit, the decrease is not regular. This is an expected behavior since non-download flows are typical signaling flows to renew requests.

Figure 23.7 depicts the CCDF of the eDonkey flow holding times on TCP level. The download connection holding time CCDF decrease moderately, cf. Figure 23.7(a), and reminds more of a linear decay in a log/log plot. The CCDF of the holding time of non-download streams, cf. Figure 23.7(b), decreases rapidly as well as un-regularly. This is an expected behavior since non-download connections are short and limited in their sensitivity on TCP flow control.

Mobile Peer-to-Peer Filesharing

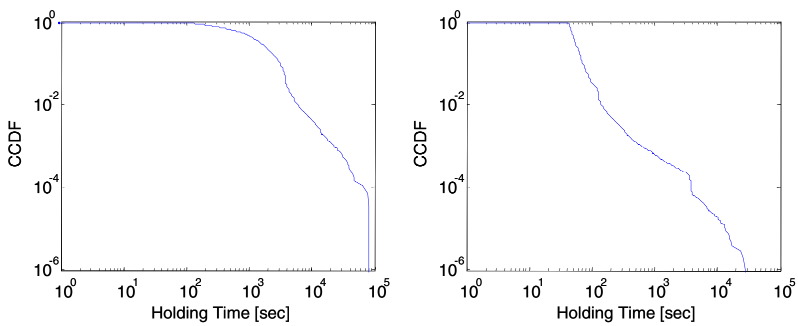
The feasibility and throughput of *mobile Peer-to-Peer* file sharing in infrastructure-based GPRS and UMTS mobile networks is examined by measurements in [298]. The measurements have been carried out in networks of two German GPRS network operators and, for the first time, in a UMTS network. The subject of the empirical investigation was the eDonkey application due to its continuing popularity [616] and its hybrid architecture, which give opportunities for network operators to interfere [456].

The measurements demonstrated that mobile Peer-to-Peer is technically feasible for GPRS technology but stability and throughput are unacceptable low if compared to fixed Peer-to-Peer. Particularly, the direct exchange of large parts of files between two mobile peers and multiple source down-



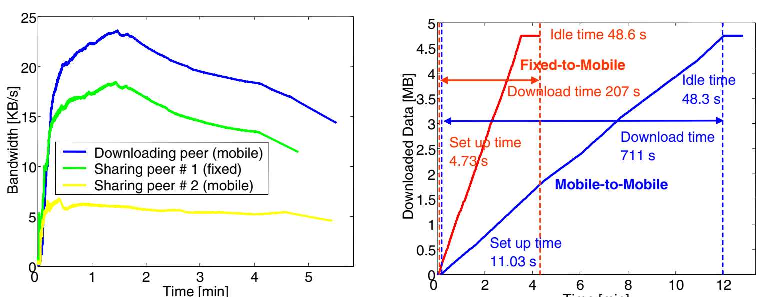
(a) Download, Outbound (b) Non-download, Outbound

Fig. 23.6: CCDF of the observed eDonkey Flow Size



(a) Download, Outbound (b) Non-download, Outbound

Fig. 23.7: CCDF of the observed eDonkey Flow Holding time



(a) Throughput for MSD (fix/mob→mob) (b) Conn. holding times

Fig. 23.8: Performance of mobile Peer-to-Peer file sharing in UMTS

load is not practical in GPRS. UMTS technology, in contrast, is more stable and has superior throughput. It extends the capabilities of the GPRS service into sufficient performance for mobile Peer-to-Peer file sharing. Figure 23.8(a) depicts the observed throughput for a multiple source download for a mobile peer downloading from a fixed peer and a mobile peer (abbreviated as fix/mob→mob). The throughput for MSD reaches a sustained level of 25 KBytes/sec. This is a value which even permits the download of larger files.

The number of traversals of the air interface, however, has to be minimized in order to reduce the traffic and the transmission delay. Figure 23.8(b) compares the TCP connection holding times for an eDonkey file part of the same size for a fixed-to-mobile transmission and for a mobile-to-mobile transmission in UMTS. Figure 23.8(a) reveals that the uplink capacity of the providing mobile peer is the bottleneck and that the connection setup time is almost doubled in the mobile-to-mobile transmission. A reduction of the necessary traversal of the air interface can be achieved efficiently by the application of caches, which has also the advantage of overcoming the asymmetric access bandwidths of mobile stations [297].

23.4 Evaluation of a Peer-to-Peer Resource Mediation Mechanism

A distributed and highly scalable approach for Peer-to-Peer resource mediation mechanisms is the concept of *distributed hash tables (DHTs)* [52]. DHT-based Peer-to-Peer algorithms like Chord [575] only need to store the location of $O(\log_2(n))$ other peers where n is the number of peers in the overlay network. They are furthermore able to retrieve information stored in the distributed network using $O(\log_2(n))$ messages to other peers. This statement, however, is very vague. It tells only the order of the magnitude of the search delay and does not provide sufficient details on the actual search time statistics. As a matter of fact the physical link delay, which is highly probabilistic, strongly influences the performance of searches in a Peer-to-Peer overlay network. Thus, the impact of network delay variation on resource mediation times, i. e. search times, in DHT based Peer-to-Peer systems has to be evaluated. The goal is to prove scalability of Chord rings in order to guarantee quality-of-service demands. An analytical model for Chord has to be deduced in order to evaluate the performance in large peer populations [74].

The phase diagram of the search delay is depicted in Figure 23.9. A particular path i is chosen with probability p_i and consists of i network transmissions T_N to forward the query to the closest known finger and one network transmission T_A to send the answer back to the searching peer. By means of the phase diagram, the generating function and the Laplace transform respec-

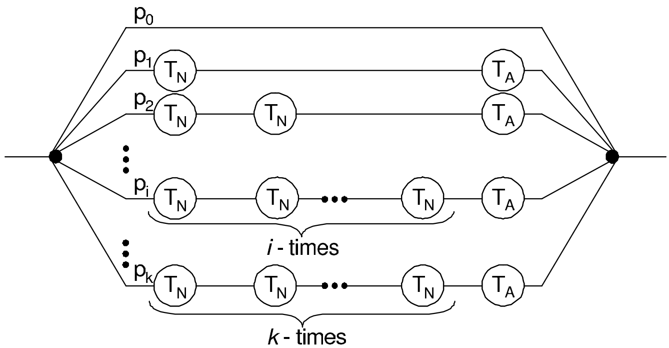


Fig. 23.9: Phase diagram of the search duration T

tively can be derived to cope with the case of discrete-time or continuous-time network transfer delay.

The distribution function of the search delay as seen from a user entering a search query to a peer in the Chord ring is computed. The analysis also gives insight into the quantiles of the search delay.

Figure 23.10, e. g., analyzes different quantiles of the search delay. It can be seen that Chord searches indeed scale logarithmically. One can observe that the search delay bound rapidly increases at smaller values of n, but stays moderate for very large peer populations. The curve is not strictly monotonically increasing as expected since a small decrease can be seen when the population n just exceeds a binary exponential value 2^i. This effect can be explained as follows: once the size of the population crosses the next power of 2, the finger table of each peer grows by one entry. Thus, the mean search duration slightly decreases at this point.

The curve with the 99 %-quantile indicates that 99 percent of search durations lie below that curve. For a peer population of, e. g., $n = 3000$ in 99 percent of all cases the search delay is less then roughly 15 times the average network latency. That is, the curves indicate bounds of the search delay, which can be used for dimensioning purposes. Compared to the mean of the search delay the quantiles of the search delay are on a significantly higher level. Still, the search delay scales in an analogous manner for the search delay quantiles.

Figure 23.11 depicts the 99 %-quantile of the search delay, with the coefficient of variation of T_N as parameter. There are five vertical lines at $n = 512$, 256, 128, 64, and 32 to point out the previously mentioned oscillations at $n = 2^i$. The larger c_{T_N} we choose, i. e. the more variation there is in the network delay, the larger is the 99 %-quantile of the search duration. Therefore, it is more difficult to guarantee Service Level Agreements in networks with larger delay variation. Timeouts, e. g. have to be set to higher values accordingly.

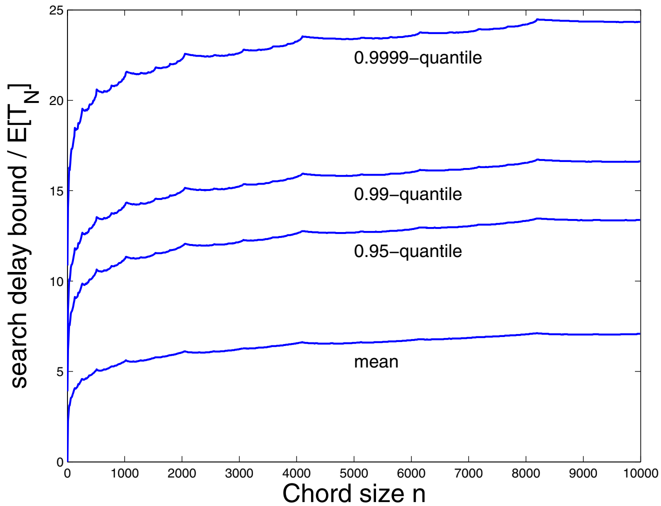


Fig. 23.10: Search delay quantiles

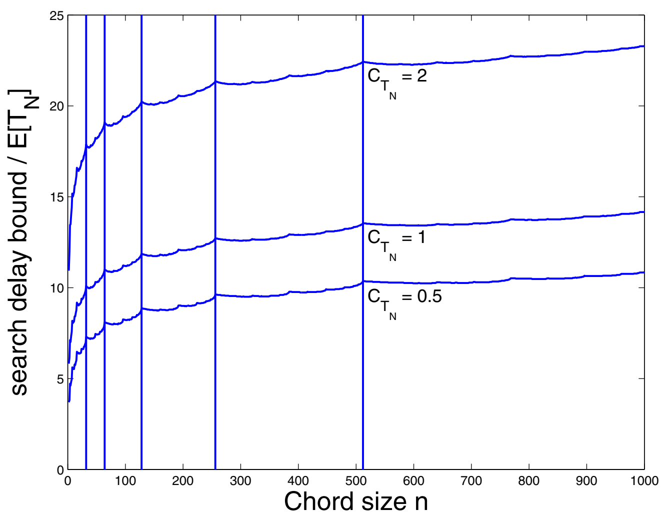


Fig. 23.11: Influence of CoV of T_N on search delay quantile

The numerical results presented above illustrate the dependency of the search duration on the variation of the network transfer delay and analyze the scalability of the Chord-based Peer-to-Peer mediation mechanism. The analysis also gives insight into the quantiles of the search delay, which can be used for system dimensioning purposes. This becomes of particular interest if real-time requirements should be obeyed in case such an algorithms is used to locate users in a Peer-to-Peer VoIP system like Skype [567].

23.5 Evaluation of a Peer-to-Peer Resource Access Mechanism in Mobile Environment

The second key function provided by Peer-to-Peer systems is the group of resource access mechanisms, i. e. algorithms for exchanging files. So far, mainly empirical investigations of their efficiency have been published. A comprehensive measurement-based evaluation of KaZaA [558], for example, is provided in [379] and the empirical performance of the BitTorrent file swapping system [128] is reported in [320, 494]. Two of the few available analytical investigations are presented in [71, 497].

Before deploying resource exchange mechanisms in the wild, it is necessary to validate their capability. In particular, if resources are limited as in mobile networks. Mobile networks differ from wireline networks mainly by the limited capacity of the radio link and the mobility of the users. High overhead for exchanging data 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.

In order to meet these requirements, a hybrid architecture for *mobile Peer-to-Peer file sharing* is proposed in [456] and analyzed in [298]. The suggested architecture, shown in Figure 23.12, is based on the popular eDonkey file sharing network and is enhanced by three specific mobile Peer-to-Peer components: a modified *index server* for mediation, a *cache peer* for storing popular files and a *crawling peer*, which supports mobile peers searching the global community.

The architecture permits the operator to *a)* participate in service creation and service control, *b)* to offer value-added services, 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.

The impact of the node life time, i. e. the churn behavior, on the download time is depicted in Figure 23.13

Figure 23.13 shows the CCDF for downloading files with average size of 5 MBytes and by applying GPRS as the access technology of mobile peers. Figure 23.13(a) shows the CCDF of the download time for popular files, i. e. the *cache peer* is used (for details see [298]). The mobile peers with the longest churn time of 12 h (red curve in Figure 23.13(a)) have the smallest download times. The more the average churn time decreases in Figure 23.13, i. e. 2 h (green line) and 30 min (blue line), the more the download time increases. Figure 23.13(b) illustrates the CCDF of the download time for unpopular files with respect to the different churn times.

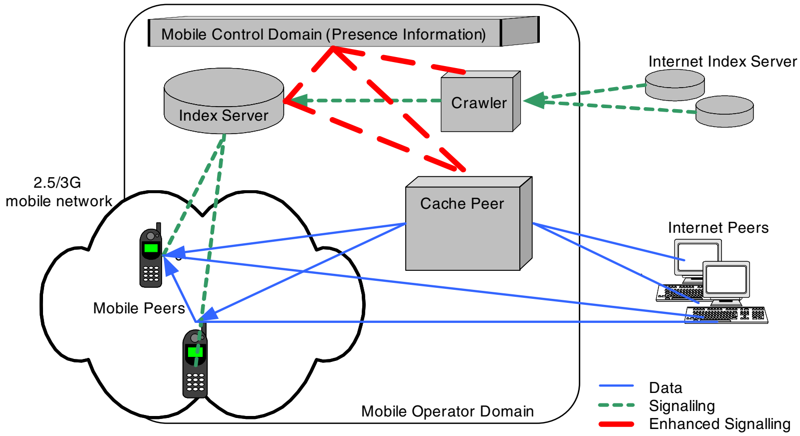
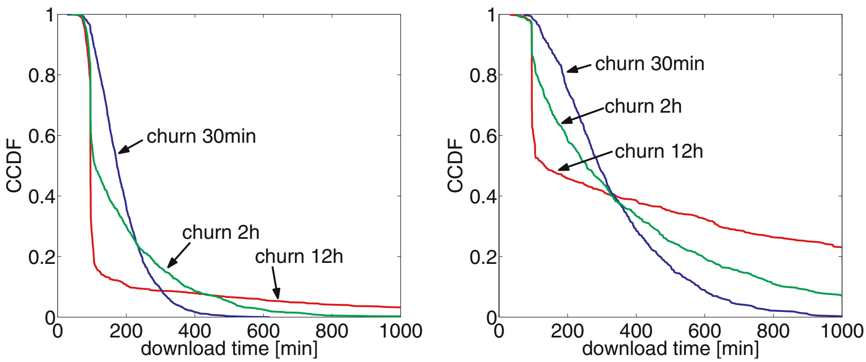


Fig. 23.12: Overview of Mobile Peer-to-Peer Architecture



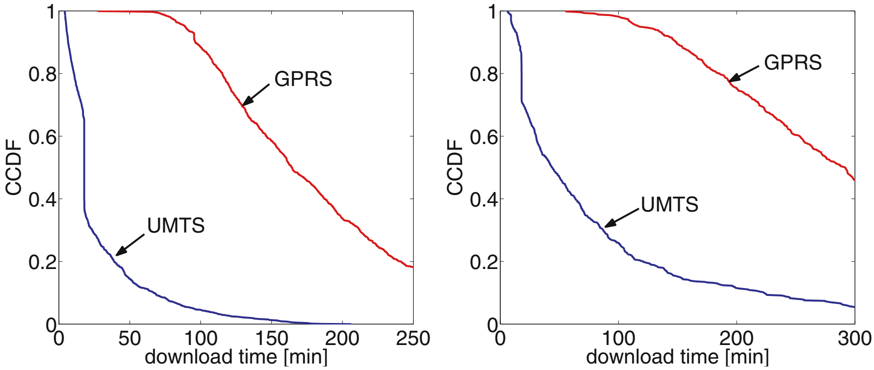
(a) Popular files

(b) Unpopular files

Fig. 23.13: Download time for different churn times

The results shows that churn behavior of the peers has significant impact on the download time of files, however the additional infrastructure entity, the *cache peer*, can reduce this effect.

Figure 23.14 compares the CCDF of the download time for popular and unpopular mp3-files of 8 MBytes. 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 much higher download times 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) Popular files (b) Unpopular files

Fig. 23.14: Download of mp3-audio files with GPRS and UMTS

a file *efficiently*. The shape of the blue curve in Figure 23.14 is characteristic for the CCDF of the download time in an *efficient* system, while the red one illustrates the behavior for *unefficient* systems. This effect becomes even more obvious for unpopular files which are not cached by the cache peer.

The results of Figure 23.14 show that mobile Peer-to-Peer file sharing is almost impossible with GPRS whereas UMTS is a good candidate for *efficient* Peer-to-Peer file swapping.

23.6 Conclusion

Peer-to-Peer architectures and Peer-to-Peer algorithms have to be evaluated according to the task they accomplish. Peer-to-Peer systems support two functions: *a) resource mediation* mechanisms and *b) resource access control* mechanisms.

The performance of a Chord-based Peer-to-Peer resource mediation mechanisms has been assessed in Section 23.4. It has turned out that the network delay has significant impact on the time to locate a resource and that the algorithm scales even for tight delay bounds, in this way fulfilling carrier grade requirements.

The efficiency of a multiple source download mechanism (resource access mechanisms) has been investigated in Section 23.5. It was shown that the download time depends highly on the type of the air interface as well as on the churn behavior of the peers.

Furthermore, it was shown by measurements that Peer-to-Peer traffic is highly variable and that Peer-to-Peer overlay management has to be performed on two timescales (cf. “short” and “stable” state in and Section 23.3.1). Moreover the measurements have revealed that multiple source download mechanisms do not increase the “mice and elephant” phenomenon (cf. Section 23.3.2) and that these mechanisms can perform efficiently even in mobile environments.

In the case of a mobile environment, measurements and the performance evaluation indicate that an optimal transfer segment size exists, which is dependent on the type of access network and the churn behavior of the peers. The determination of this size is for further research.