ITC19/ Performance Challenges for Efficient Next Generation Networks 1817-1826
LIANG X.J. and XIN Z.H.(Editors)
V.B. IVERSEN and KUO G.S.(Editors)
Beijing University of Posts and Telecommunications Press

Comparison of Crawling Strategies for an Optimized Mobile P2P Architecture

Tobias Hoßfeld [1], Andreas Mäder [1], Kurt Tutschku [1], Phuoc Tran-Gia [1], Frank-Uwe Andersen [2], Hermann de Meer [3] and Ivan Dedinski [3]

[1] Department of Distributed Systems, Institute of Computer Science, University of Würzburg, Am Hubland, 97074 Würzburg, Germany.
{hossfeld,maeder,tutschku,trangia}@informatik.uni-wuerzburg.de
[2] Siemens Communications, Siemensdamm 62, 13623 Berlin, Germany.
frank-uwe.andersen@siemens.com
[3] Chair of Computer Networks and Computer Communications, University of Passau, Innstr. 33, 94032 Passau, Germany.
{demeer,dedinksi}@fmi.uni-passau.de

Abstract. Mobile networks differ from their wireline counterparts mainly by the high costs for air transmissions and by the mobility of the users. A new entity, denoted as the *crawling peer*, is suggested in order to optimize the resource mediation mechanism for a mobile P2P file sharing application. The crawling peer locates content on behalf of mobile peers. It is placed in the wireline part of the mobile network and thus, does not suffer from the above mentioned restrictions. The crawling peer is part of a comprehensive mobile P2P file sharing architecture [1] which is based on the popular eDonkey file sharing application. The performance of three querying strategies of the crawling peer is investigated with respect to banning at the index servers and the response time of requests, i.e. the time to find a file. The results show that the selection of an appropriate request strategy for the crawling peer maximizes the probability of locating a file while the probability to be banned by an eDonkey index server is minimized.
Keywords: P2P, mobile network architecture, resource mediation

1 Introduction

The last years have seen two success stories in networking. Cellular mobile networks have gained tremendous popularity, e.g. the number of GSM subscribers rose in Germany within ten years from 1.76 million (1993) to 63.5 million (2003) [2]. A similar extreme growth has only been matched by peer-to-peer (P2P) file sharing services like Napster, eDonkey/eMule or BitTorrent. Within the five years since the start of Napster, they have evolved to the most dominant application in the Internet in terms of transmission volume [3, 4]. A continuation of the GSM success story by UMTS was expected but, at least in Europe, is still evolving. This fact comes mainly from the absent of services and applications for this technology [5]. UMTS network operators are currently looking for applications which do both: *a)* exploit, qualitatively and quantitatively, the potential of the UMTS technology and *b)* motivate the user to adopt the new technology. In that way, *mobile P2P file-sharing* is an interesting candidate for such an application.

Mobile networks differ from wireline networks mainly by the limited capacity of radio channels and by the mobility of the users. The high costs of air transmission ask for a minimization of any signalling. The user mobility results in rapidly varying on-line states of users and leads to the discontinued relaying and buffering of signalling information. This can be achieved for example by entities which on behalf of others store content, i.e. *proxies*, or entities which locate information, i.e. *crawlers*.

Therefore, mobile P2P file sharing networks ask for new architecture solutions for these kinds of P2P services. A new entity, the so-called *crawling peer*, is placed in the wired part of the mobile network and locates files on behalf of mobile peers. Research on the mediation performance in P2P systems is fundamental. The crawling peer might be an alternative to highly distributed concepts such as *Distributed Hash Tables*, as used in Chord [6], or *flooding concepts*, as used in Gnutella.

In this paper we investigate the performance of a crawling peer as introduced in [1]. Section 2 describes the Mobile P2P architecture. In Section 3, we measured typical values of real eDonkey index servers which are used as input parameters in our investigation. The considered network and the crawling peer are modeled in Section 4. Numerical results with analytical approximations are given in Section 5 and Section 6 gives a conclusion of our work.

2 Mobile P2P Architecture

The suggested mobile P2P architecture for third generation mobile networks first introduced in [1] is depicted in Figure 2. The suggested concept is based on the architecture of the popular eDonkey P2P file sharing application and was enhanced by three specific entities: the *cache peer*, the *mobile P2P index server*, and the *crawling peer*.

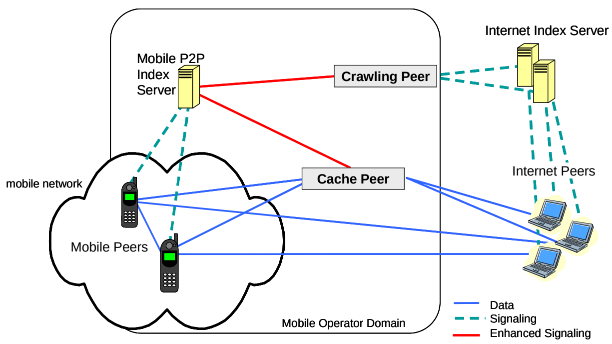

Fig. 1. Architecture concept for a P2P file-sharing service optimized to mobile networks

The *cache peer* is a modified eDonkey peer located in the wireline part of the mobile P2P architecture that can be triggered to download often requested files and then offers these files to the community. The application of the cache peer reduces the traffic on the radio interface [7]. The *mobile P2P index server* is a modified eDonkey index server. It tracks the frequently requested content, triggers the cache peer to fetch it, and forces the mobile peers to download the file from the cache peer, if available.

The *crawling peer* is also located in the wireline part of the suggested mobile P2P architecture and searches content on behalf of other mobile peers. The crawling peer can locate files even when a mobile peer is not online. As a result, the search traffic is shifted

$\mu(\widetilde{F}_i)$ of registered files. The ID of an index server reflects its position in the sorted list, $\forall i, j \in \{1, \cdots, N\} : i < j \Leftrightarrow \mu(\widetilde{F}_i) > \mu(\widetilde{F}_j)$.

The time for answering a search request by an index server is modelled by using the round trip time. Figure 2 shows the cumulative distribution function (CDF) of the round trip times for two public index servers, the largest index server in the eDonkey network with ID 1 and a Chinese eDonkey server with ID 48. The latter has about 12,000 users with 540,000 shared files. The blue curves indicate the measured values which we fit by a lognormal distribution. In detail, the round trip R_i of an index server i is modelled by (1), whereas $\mu(x)$ and $\sigma(x)$ returns the mean value and the standard deviation of the values x, respectively. The resulting CDFs are plotted in Figure 2 as red curves and we can obtain a good match (for more details see [10]) with

$$R_i = \mathrm{DET}(d) + \mathrm{LOGN}(m, s) = \min(\widetilde{R}_i) + \mathrm{LOGN}\left(\mu(\widetilde{R}_i - \min(\widetilde{R}_i)), \sigma(\widetilde{R}_i)\right). \quad (1)$$

4 Model of the Network and the Crawling Peer

We consider a mobile P2P-network as proposed in [1] and as introduced in Section 2. In the mobile network, the users generate a Poisson arrival process of requests for files which cannot be found in the mobile domain. Therefore the requests are delegated to the crawling peer (CP). The request arrival rate is denoted with λ. The CP then asks for the file at the known index servers in $\mathcal{I}$ according to a specific request strategy. In order to increase the efficiency of the search, the CP may ask a number of k servers simultanously. The search stops if either at least one request was successful, since we assume that additional sources – if available – can be found by eDonkey's source exchange mechanism, or, if no source has been found. The file request success probability on an individual index server $i \in \mathcal{I}$ is modelled by the probability f_i, which is derived from the measurements we described in Section 3. It is defined as $f_i = \frac{\mu(\tilde{F}_i)}{\sum_{i\in\mathcal{I}} \tilde{F}_i}$, i.e. according to the distribution of the file registrations at the index servers.

The banning of clients has been introduced lately by the creators of the "lugdunum index server", which is the software platform of choice for the majority of the index servers in the public eDonkey network. The index server has for each requesting client a number of credit points. For each file request, the credit is decreased by normally 16 points, while in turn in each second one point is added. A more detailed description of the banning mechanism can be found on the web [8].

The banning mechanism is modelled as following. Each index server has c_i credit points. Initially, the credits are set to a value of c_{init}, which is around 1000 credits according to the references we found on the web. On each request at i, the credits are reduced by c_r points. Once the crawling peer is banned at an index server, it stays banned forever. This is a worst case assumption since we have no information about the ban time as it is implemented in the public eDonkey network.

The return time from the begin of the request for an index server until the report of the results is modelled with the measured round trip times as introduced in Section 3. The access time to the file location database in the server is neglected.

The goal is now to identify request strategies which deliver good results in terms of the file request success probability p_s and the mean search time μ_s. We define the success