A measurement study supporting P2P file-sharing community models

R. Bolla a, R. Gaeta b, A. Magnetto b, M. Sciuto a, M. Sereno b,c

a Dipartimento di Informatica, Sistemistica e Telematica, Università di Genova, Via Opera Pia 13, 16145 Genova, Italy
b Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy

ABSTRACT

Knowledge of emergent properties of existing peer-to-peer file-sharing communities can be helpful for the design and implementation of innovative peer-to-peer protocols/services that exploit autonomicity, self-configuration, resilience, scalability, and performance. It is well known that the performance of this class of peer-to-peer applications depends on several parameters that represent the topological structure of the overlay network, the users’ behavior, and resource dynamics. Estimation of these parameters is difficult, but it is crucial for analytical models as well as for realistic simulation of peer-to-peer applications.

In this paper, we present an active measurement-based study designed to glean insights on the above parameters within the Gnutella network. The measurement software that we developed is able to collect topological information about the overlay network topology in a few minutes; in a second step, it contacts the users discovered during the topological measurement in order to acquire a novel dataset regarding the shared resources.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The support of media exchange communities, where large groups of users with common interests exchange self-made multimedia documents (e.g., music, video, presentations, etc.), is one of the most successful and interesting Internet service typologies to date. Peer-to-peer (P2P) technologies are one of the most innovative attempts to support these communities. In fact, P2P file-sharing applications currently represent a large fraction of overall Internet traffic [3] due to their high popularity among Internet users. Several papers have analyzed the characteristics of P2P systems as well as the peculiarities of the network traffic generated by such applications and their impact on the network. Estimation of system parameters, even though it undoubtedly represents a complex task (due to factors such as the large size of the P2P overlay networks), can be beneficial for many research activities, like, among the others:

- The design of innovative P2P services and applications that exploit the advantages of existing programs and the characteristics of their communities, such as autonomicity, self-configuration, resilience, scalability, and performance.
- The modeling of P2P file-sharing applications, since analytical models and simulators must be fed with realistic values for the input variables.
- The development of traffic engineering strategies, which can support an efficient use of network resources, and appropriate strategies for the management of the network traffic originated by P2P file-sharing applications.

In this paper, we present an active measurement study of the characteristics of peers and files in the real Gnutella overlay network. We focused our attention on this P2P community since the Gnutella overlay network is unstructured and completely decentralized, thus making it a self-configuring, self-organizing and self-managing structure; furthermore, it represents an extremely popular and long-lived community.

We provide results regarding system parameters that have been the subject of previous studies with the goal
of investigating the evolution and stability of the Gnutella community over time. On the other hand, the most relevant part of the paper presents the following original results:

- The effects of network address translation (NAT) on both the overlay network connectivity and the degree of connectivity of peers (single or reciprocal).
- The problem of edge distortion represented by connections among ultra-peers that are established or closed during a crawl, and that are likely to be reported only by one end of the connection.
- The percentage of peers and ultra-peers and their geographical position.
- The behavior of users, involving the length of their connections and disconnections, their rate of ingress into the network and the number of files they share.
- A classification of resources based on their popularity profiles, along with the probability distribution that files will belong to a popularity class.
- The rate of births of distinct categories of files in the system.
- The evolution of each file’s popularity.

One of the goals of our work is to provide useful tools for modeling a P2P file-sharing system. To this end, for the most interesting measured distributions, we provide the best-fitting curve (chosen among the exponential, lognormal, Weibull and Pareto distributions) to represent the measured phenomena.

The outline of the paper is the following: in Section 2, we briefly summarize previous empirical findings from measurement-based investigations of P2P communities. Section 3 describes the architecture of the software that we developed, which has been used to capture snapshots of the Gnutella overlay. Measurement results are then presented in Section 4, where we discuss the difficulties that we faced in collecting information from a wide area overlay network; Section 5 reports our analysis on the properties of the overlay topology, Section 6 concerns the behavior of users, while in Section 7, we describe the peculiarities of files shared by users. Finally, Section 8 synthesizes connections throughout our work and outlines several developments for current and future investigation.

2. Related works

In the past, many crawlers operating on the Gnutella network have been proposed, with the aim of acquiring snapshots of the overlay network at many points in time. One of the first incarnations was developed by Jovanovic et al. from the University of Cincinnati [5]. The work was aimed at studying the overlay network topology, by acquiring the lists of neighbors by peers, in order to reconstruct the network using graph theory. This crawler could contact 1000 peers on average, rebuilding graphs with approximately 4000 relationships between neighbors. A few months later, Ripeanu et al. developed another Gnutella crawler, whose application was important for achieving results regarding the size of the network [9]. In May 2001, Saroiu et al. from the University of Washington, published a comparative measurement work comparing Napster and Gnutella [11]. A relevant study, although different from the one presented here, is the passive distributed crawler gnuDC [15], designed to establish connections with the network nodes in order to collect information on their traffic. As the new version (0.6) of the Gnutella protocol was implemented in the most popular software clients (Limewire, BearShare, Shareaza, Phex, etc.), the measurement crawlers were also improving their efficiency, allowing for the discovery of overlay network topology, with faster acquisitions (by at least two orders of magnitude). In 2005, the first crawler for the Gnutella 0.6 network was realized by R. Rezaje, D. Stutzbach and S. Zhao from the University of Oregon. It is composed by two modules: an overlay network component and a content crawler, working on shared resources. The software is designed to work via one master and six slave components that operate in parallel. Each component contacts a number of ultra-peers, depending on its CPU load. The crawler is extremely powerful, able to contact up to 140,000 peers per minute. In [14], a study on the complexity of churning in different P2P systems is presented. Reza et al. investigated the properties of user dynamics with a focus on the session lengths for users of Gnutella, BitTorrent and Kad. They find lognormal distributions to be best for modeling the session lengths of Gnutella users, though this analysis was restricted to 1 day of measurement only. They provided results based on a longer period for users of BitTorrent. However, in the case of a particular shared file (i.e., a Linux ISO image), the results were potentially biased, given that the contents generally shared over P2P file-sharing networks are music or video files. In [6], the authors present an interesting measurement study regarding query behavior in the Gnutella network. Their results primarily concern the rate of queries, with a focus on the influence of geography, day and time of day. The primary work regarding the characteristics of files in the Gnutella system is [16], where Reza et al. examined several properties of files. They presented results both for the static and dynamic properties of resources. Finally, an innovative contribution was recently published [1], analyzing the topological properties of a P2P streaming overlay and their behavior over time.

3. The Gnutella crawler

According to the results presented in [12,13], the accuracy of captured network snapshots depends on the crawling speed. In principle, a perfect snapshot is captured if the crawling process is instantaneous and complete (that is, if all peers are contacted). However, neither of these conditions is generally met for the following reasons: rapidly changing topology, and unreachable peers. Two studies presented a distributed crawling technique that aims to mitigate these problems (see [12,13]).

To obtain accurate network snapshots, we used a distributed crawler architecture, similar to the one described in [12], but modified with an innovative two-phase crawling technique, as explained in the following section.
3.1. Gnutella 0.6 features

Gnutella 0.6 clients (i.e., clients, compliant to version 0.6 of the Gnutella protocol implementation) use a two-tier overlay structure that divides peers into two groups: ultra-peers and leaf-peers [4,7]. Fig. 1a depicts the overlay structure of the actual Gnutella version: the ultra-peers form a top-level overlay while the leaf-peers are connected to the top-level overlay through one or multiple connections. Our crawler exploits this two-tier structure by only crawling the top-level peers (i.e., ultra-peers). Since each ultra-peer maintains the list of its adjacent ultra-peers and the list of the leaf-peers connected with itself, this feature enables the crawler to reduce the number of contacted peers.

3.2. Master–slave crawler architecture

The crawler employs this type of architecture in order to achieve a high degree of concurrency to use the available PCs. For our experiments, we used 10 off-the-shelf PCs, divided into one master and nine slaves. A master process coordinates multiple slave processes. The slave processes are independent crawlers that crawl different portions of the overlay topology. Each slave asynchronously requests ultra-peer addresses from the master according to the slave crawling speed and inserts these received addresses into a queue (the slave queue). The slave queue is used to keep the slave processes busy; it is filled by the slave requests and cannot be saturated. We set the master queue length at 80,000 addresses. To increase the degree of concurrency, each slave process uses a multi-threaded architecture where the slave queue is shared among all the slave threads. The multi-threading level of the slave processes used in the current experiment is 220. The slave processes report back to the master process with the following information: ultra-peer neighbors, and the content of headers such as X-Try, X-Try-Ultrapeers. Fig. 1b depicts the master–slave crawler architecture.

3.3. Two-phase crawling

In order to reduce the crawling time, we use a two-phase crawling strategy. The first phase is performed by the master process. During this phase, the master crawls through the overlay topology in order to collect an initial global ultra-peer addresses queue. During our experiments, we collected approximately 80,000 ultra-peer addresses. This queue will be the starting point for the second phase in which the crawling is performed by the slave processes that obtain ultra-peer addresses from the master process. The aim of this architecture is to speed up the crawling.

3.4. Timeout management

In order to avoid long delays in the case of unresponsive ultra-peers, we tune a timeout initial value equal to 40 s. In case of contacts that terminate with a timeout, we did not find it convenient to implement a retry policy, which has been used in the past simply to capture information about resources, as explained further below. With these optimizations, the crawler is able to collect snapshots of the Gnutella overlay network, exploiting the flooding features of the Gnutella protocol. The average duration of each crawling experiment (topological crawling) is 624 s, with an almost negligible standard deviation.

In addition to the architecture described above, we developed a second crawling software that uses a similar master–slave architecture for capturing information about the shared resources. It first creates a list of reachable ultra-peers, and then the master sends the addresses to the slaves, which request the information using a load adaptive policy.

Each slave tries to connect the ultra-peers, opening multiple connections in leaf mode and sending them standard Query messages. The slave then waits for the reply messages (QueryHits) that can be originated by the contacted ultra-peers or by some of their leaf-peers; the addresses of unresponsive ultra-peers are sent back to the master process, which puts these ultra-peers in a retry queue. For every

![Diagram](image1.png)
(a) Gnutella overlay topology

![Diagram](image2.png)
(b) The master-slave crawler architecture

Fig. 1. Figure (a) depicts the two layer hierarchy of the Gnutella overlay, composed by top-level (ultra-peers) and second-level (leaf-peers) users. Figure (b) shows the crawler architecture.
received QueryHit message, each slave stores the following information: IP address, port and SID of the owner (even if peers often declare their private address), along with size, hash and extension of the file. At the end of the crawling phase, the information gathered by the slaves was merged in order to remove duplicated data.

The information about the shared resources, which will be investigated in the following sections, concerns about 500,000 servents (both ultra-peers and leaf-peers) on average for each snapshot, depending on the number of users replying to the “Query” messages.

4. Crawling accuracy and data set

The accuracy of the crawling experiments is strictly dependent on the percentage of non-responding peers, that is, peers whose identity has been discovered during the exploration of the overlay network but that deny (or excessively delay) their reply to a connection request from the measurement software. Before we present results from our experiments, we discuss the issue of non-responding peers and describe the data set we collected.

4.1. Non-responding peers

In our experiments, peers are identified by the IP address of the machine used to connect to the Internet, as well as by the port number selected by the Gnutella client in order to implement application layer message exchange. We immediately observed that several peers share the same IP address but use different port numbers; this is probably due to the widespread deployment of network address translation (NAT) techniques. To quantify this phenomenon, we ran a few pilot experiments by taking a snapshot of the Gnutella network to compute the distribution of the number of ultra-peers sharing their IP addresses with other ultra-peers.

Table 1 presents results from one of these samples that lead us to observe that approximately 53% of ultra-peers possess a unique IP address, while the remainder share their IP address with other ultra-peers. In this experiment, we observed up to 2000 ultra-peers that are known to the Internet through the same IP address.

The accessibility of NATed ultra-peers can be problematic, and it may negatively impact the accuracy of crawling experiments, because non-responding ultra-peers would increase the time required for the crawling due to frequent time-out expirations. To investigate this issue, we successively ran additional experiments designed to compute the distribution of the number of responding ultra-peers as a function of the number of ultra-peers that share the same IP address. The results of this second exploratory experiment are summarized in Table 2.

We observed that approximately 25% of the total number of known ultra-peers did not respond to connection messages from our crawler. Within this population of non-responder ultra-peers, approximately 38% was represented by ultra-peers whose IP addresses were unique. We can thus conclude that NAT is primarily responsible for the uncooperative behavior of ultra-peers during the crawling experiments. Unfortunately, contacting all of the ultra-peers that have the same IP address but different port numbers may yield an unacceptable increase of the crawling time, which can affect the statistical characteristics of the collected data (as previously pointed out by Stutzbach et al. [13]). Since there is no easy method for identifying NATed ultra-peers, and because knowledge of how many ultra-peers have the same IP address but different port

<table>
<thead>
<tr>
<th>Number of different ports (p) per IP address</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10 (11–2000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of IP address</td>
<td>584,500</td>
<td>105,347</td>
<td>21,645</td>
<td>6764</td>
<td>3936</td>
<td>3648</td>
<td>3849</td>
<td>3309</td>
<td>1470</td>
<td>143</td>
</tr>
<tr>
<td>Number of ultra-peers</td>
<td>584,500</td>
<td>210,694</td>
<td>64,935</td>
<td>27,056</td>
<td>19,680</td>
<td>21,888</td>
<td>26,943</td>
<td>26,472</td>
<td>13,230</td>
<td>1430</td>
</tr>
<tr>
<td>%</td>
<td>53.7</td>
<td>19.3</td>
<td>5.9</td>
<td>2.4</td>
<td>1.8</td>
<td>2.0</td>
<td>2.4</td>
<td>2.4</td>
<td>1.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2

Number of responding ultra-peers as a function of the number of ultra-peers sharing the same IP address.

<table>
<thead>
<tr>
<th># of ultra-peers sharing their IP address</th>
<th>Number of responding ultra-peers</th>
<th>Fraction of non-responding ultra-peers (%)</th>
<th>Fraction of multiple responders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>110,822 365,597</td>
<td>9.6</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>71,022 181,042 11,858</td>
<td>6.2</td>
<td>1.03</td>
</tr>
<tr>
<td>3</td>
<td>46,731 90,783 7443 828</td>
<td>4.1</td>
<td>0.71</td>
</tr>
<tr>
<td>4</td>
<td>26,472 45,980 4552 576 128</td>
<td>2.3</td>
<td>0.45</td>
</tr>
<tr>
<td>5</td>
<td>14,480 32,275 2895 405 145 70</td>
<td>1.2</td>
<td>0.30</td>
</tr>
<tr>
<td>6</td>
<td>9882 32,244 1740 330 60 66 36</td>
<td>0.8</td>
<td>0.19</td>
</tr>
<tr>
<td>7</td>
<td>7539 33,607 1001 196 77 28 0 35</td>
<td>0.6</td>
<td>0.11</td>
</tr>
<tr>
<td>8</td>
<td>3928 26,328 888 176 24 32 16 8 32</td>
<td>0.3</td>
<td>0.19</td>
</tr>
<tr>
<td>9</td>
<td>1269 12,591 639 108 9 27 0 9 18 27</td>
<td>0.1</td>
<td>0.07</td>
</tr>
<tr>
<td>10</td>
<td>580 1980 420 50 0 20 0 0 0 0 20.05</td>
<td>0.05</td>
<td>0.04</td>
</tr>
</tbody>
</table>

...
numbers can only be collected at the end of the experiment, we designed our crawler to contact an IP address only once. This choice led our crawler to neglect information that could be gained by contacting all ultra-peers behind the same IP. However, Table 2 shows that only approximately 3% of the total number of responses come from multiple responses, in this case from ultra-peers sharing the same IP address.

4.2. Data set description

As described in Section 3, the crawling experiments can provide two distinct data sets: (i) snapshots reporting information about the overlay topology and (ii) snapshots concerning the resources shared by users. In the following section, we analyze a group of successive measurements, acquired periodically at a constant rate, with the term battery. The duration of the battery derives from the collection time of each snapshot and from the constant time interval between successive acquisitions. We identify each snapshot with a progressive number (ID). Because the average acquisition time shows a negligible variance, and because the time between the acquisitions is constant, it is possible to infer the date and the hour of collection from the snapshot ID.

The results concerning the overlay topology and the behavior of users have been obtained by the analysis of one set of snapshots that reports topological information. We denote this group of acquisitions as top_battery. These crawling experiments were collected between 04/02/2007 and 04/12/2007: during this period, we collected 1244 snapshots of the Gnutella overlay network. The average duration of crawling experiments was 624 s (with an almost negligible standard deviation). Each crawling experiment was started after a constant wait time of 70 s after the end of the previous collection.

Information about the shared resources comes from two batteries of separate experiments, denoted as res_battery1 and res_battery2. The res_battery1 was performed between 09/05/2007 and 11/22/2007, and it is composed by 583 snapshots. It represents the main source of data for our results. The specific analysis presented in Section 7.2 required a second shorter set of measurements (res_battery2), that comprised 205 snapshots captured between 07/28/2007 and 09/03/2007.

5. Topological and static properties

In this section, we present results regarding the number of peers joining the Gnutella overlay network, the topological structure of the overlay, the phenomenon of edge distortion, and the characterization of network proximity among peers of the Gnutella network. Throughout the following, we denote the set of responding ultra-peers as \( RU = \bigcup_{k=0}^{k_{\text{max}}} RU_k = \bigcup_{k=0}^{k_{\text{max}}} LP_k \) where \( RU_k \) is the set of responding ultra-peers whose number of connections in the top-overlay is equal to \( k \) and \( LP_k \) is the set of responding ultra-peers connected to a leaf-peer. The values of \( k_{\text{max}} \) and \( l_{\text{max}} \) are snapshot dependent. Furthermore, the set of leaf-peers is denoted as \( L = \bigcup_{n=1}^{n_{\text{max}}} U_n \) where \( U_n \) is the set of leaf-peers whose number of connections to ultra-peers is equal to \( n \). Also in this case, the value of \( n_{\text{max}} \) is snapshot dependent. All results are computed from the data collected in the top_battery.

5.1. Gnutella overlay network size

The plots of Fig. 2 shows the typical daily behavior for the top_battery.

The number of discovered ultra-peers ranged from 608,846 to 1,111,009, while the number of responding ultra-peers ranged from 257,844 to 503,249. On the other hand, the number of leaf-peers ranges from 2,655,127 to 5,325,162. It is interesting that these values are higher than previous accounts [13] that reported up to 158,345 responding ultra-peers and 873,130 leaf-peers for the February 2005 snapshots (data taken from [13, page 53, Table 1]). Furthermore, by mapping IP addresses to autonomous system identifiers, we also characterized the geographical localization of peers. In Table 3, we report data from the snapshots with the highest (snapshot no. 617) and the lowest (snapshot no. 572) number of peers referring to the snapshots in the interval 572–708. We limit our analyses to two sample snapshots because similar phenomena are reproduced all over the entire battery. It can be noted that, regardless the time of the day when the snapshots were taken, approximately 60% of the total number of peers and more than 70% of ultra-peers are localized in North America (U.S. and Canada), while non-European countries (excluding North America) constitute about 10% of the total number of participating peers.

5.2. Gnutella overlay network topology

The topological structure of the Gnutella overlay network can be characterized by estimating the distributions of three variables: (i) the number of connections among ultra-peers in the top-level overlay; (ii) the number of connections an ultra-peer keeps to leaf-peers; (iii) the number of connections a leaf-peer maintains to ultra-peers.

For each snapshot, we denote the set of ultra-peers with \( k \) responding neighbor ultra-peers as \( RN_k \) (of course, \( RU = \bigcup_{k=0}^{k_{\text{max}}} RN_k \)). For each snapshot, we compute the fraction of ultra-peers whose number of connections in the top-overlay is equal to \( k \) as \( p_k = \frac{|RN_k|}{|RU|} \). The fraction of ultra-peers whose number of responding neighbor ultra-peers is equal to \( k \) as \( m_k = \frac{|RN_k|}{|RU|} \), while the fraction of ultra-peers with \( k \) leaf-peers connected as \( l_k = \frac{|RN_k|}{|RU|} \). We also compute the fraction of leaf-peers connected to \( k \) ultra-peers as \( u_k = \frac{|RN_k|}{|RU|} \).

Fig. 3a displays the distribution \( \{p_k\} \) on a logarithmic scale for two reference snapshots and for the February 2005 snapshot analyzed in [13]. It can be noted that the results presented in 2005 [13, page 5, Fig. 4] are quite similarly reproduced by our measurements, although the size of the Gnutella network has increased since then.\(^1\) In particular, Rejaie et al. showed in [13, page 2, Fig. 1] that the Gnutella overlay size was increasing. Moreover, they described a set of users (around 1.2 M) much lower than the one that we recently observed in our work (around 5 M).
ticular, no power-law fitting of the experimental data can be performed; rather, a Poisson-like fitting could be sought for the distribution $f_{rnk\,g}$ of the connections among ultra-peers restricted to the responding neighbors (Fig. 3b).

We also present the distributions $f_{lk\,g}$ (Fig. 3c) and $f_{uk\,g}$ (Fig. 3d). Also in these cases, preliminary results were confirmed by our measurements, revealing that most of the ultra-peers maintained about 30 connections to leaf-peers. In addition, more than 95% of the leaf-peers are connected to no more than five ultra-peers.

Fig. 2c presents the result of a time-dependent study of both the average number of connections among ultra-peers and of the average number of connections to leaf-peers that we defined as $p = \sum_{n=1}^{n_{\max}} k \cdot p_k$ and $l = \sum_{h=1}^{h_{\max}} h \cdot l_h$, respectively. We considered these two measures for the snapshots in the interval 572–708 that corresponds to 1 day of repeated network measurements. The comparison between data plotted in Fig. 2c and numbers reported in Fig. 2a and b reveals that, although the number of peers follows the typical daily behavior, these average measures remain practically constant during the observation period.

5.3. Edge distortion

The authors of [13] discussed the problem of edge distortion represented by connections among ultra-peers that are established or closed during a crawling experiment that are likely to be reported only by one end of the connection. In our study, we investigated this issue by computing, for each ultra-peer in a snapshot, its reciprocity coefficient as the fraction of connections that responded to ultra-peers that are also reported by the other end. To quantify the effect of edge distortion for ultra-peer $u$, we compute in each snapshot the number of bi-directional connections that responded to ultra-peers in the top-overlay network, denoted as $b_u$. We define the average fraction

| Table 3 |
| Geographical localization of peers computed for snapshots number 572 and 617. |

<table>
<thead>
<tr>
<th>Snapshot no. 572 (11:57 a.m. 4/6/2007, GMT)</th>
<th>Snapshot no. 617 (7:37 p.m. 4/6/2007, GMT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Total (%)</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>US</td>
<td>50.19</td>
</tr>
<tr>
<td>GB</td>
<td>7.50</td>
</tr>
<tr>
<td>CA</td>
<td>6.84</td>
</tr>
<tr>
<td>AU</td>
<td>4.44</td>
</tr>
<tr>
<td>DE</td>
<td>2.95</td>
</tr>
<tr>
<td>NL</td>
<td>2.89</td>
</tr>
<tr>
<td>PL</td>
<td>2.71</td>
</tr>
<tr>
<td>FR</td>
<td>2.55</td>
</tr>
<tr>
<td>BR</td>
<td>2.26</td>
</tr>
<tr>
<td>JP</td>
<td>2.12</td>
</tr>
<tr>
<td>TR</td>
<td>1.70</td>
</tr>
<tr>
<td>SE</td>
<td>1.09</td>
</tr>
<tr>
<td>IT</td>
<td>1.05</td>
</tr>
<tr>
<td>Other</td>
<td>11.71</td>
</tr>
</tbody>
</table>

Fig. 2c presents the result of a time-dependent study of both the average number of connections among ultra-peers and of the average number of connections to leaf-peers that we defined as $p = \sum_{n=1}^{n_{\max}} k \cdot p_k$ and $l = \sum_{h=1}^{h_{\max}} h \cdot l_h$, respectively. We considered these two measures for the snapshots in the interval 572–708 that corresponded to 1 day of repeated network measurements. The comparison between data plotted in Fig. 2c and numbers reported in Fig. 2a and b reveals that, although the number of peers follows the typical daily behavior, these average measures remain practically constant during the observation period.

5.3. Edge distortion

The authors of [13] discussed the problem of edge distortion represented by connections among ultra-peers that are established or closed during a crawling experiment that are likely to be reported only by one end of the connection. In our study, we investigated this issue by computing, for each ultra-peer in a snapshot, its reciprocity coefficient as the fraction of connections that responded to ultra-peers that are also reported by the other end. To quantify the effect of edge distortion for ultra-peer $u$, we compute in each snapshot the number of bi-directional connections that responded to ultra-peers in the top-overlap network, denoted as $b_u$. We define the average fraction
of reciprocal connections for ultra-peers connected to \( k \) responding neighbors as

\[
rc_k = \frac{\sum_{u \in RN_k} b_u}{k \cdot |RN_k|}.
\]

We then compute the average fraction of reciprocal connections as

\[
\overline{rc} = \sum_{k=1}^{k_{max}} m_k \cdot rc_k.
\]

We computed \( \overline{rc} \) for all of the snapshots in the battery. The values of this measure range from 0.85 to 0.88, which indicates a rather high degree of edge distortion. Since in [13] it is argued that the degree of distortion is a function of the crawl duration relative to the rate of change in the overlay, we conducted a second experiment. We randomly selected a set of 100 ultra-peers from a pilot snapshot, and then we repeatedly contacted them, as well as their neighbors in the top-level overlay network. Since the size of the set of contacted ultra-peers was rather small, these limited crawls took only a short time to complete (max. 10 s). Of course, only a subset of the selected 100 ultra-peers may respond, and for them, we computed the average fraction of reciprocal connections for the responding ultra-peers that we denote as \( \overline{rc} \). We obtained values of \( \overline{rc} \) ranging from 0.90 to 0.95, thus confirming that faster snapshots would yield more accurate results. Nevertheless, we observed that the snapshots that we collected always contain some ultra-peers that are reported as neighbors by very large numbers of other ultra-peers, but that do not advertise any neighborhood when they reply to our crawler. This suggests that edge distortion is a problem not only related to crawling duration. That is, other factors may play a role in edge distortion (e.g., Gnutella clients that do not update their neighborhood lists frequently, or that implement non-standard strategies to manage these lists).

5.4. Network proximity

We investigated the relationship between the overlay topology of the Gnutella network and the underlying IP topology. In particular, we introduced measures that attempted to quantify the mismatch between the overlay and the IP topologies. This issue has been investigated by Ripeanu et al. [8]. That paper demonstrated that there is a strong mismatch between the two topologies. In other words, the Gnutella network does not use any locality information to build and maintain its overlay topology. Locality awareness is an important issue at the system level in order to optimize performance of the applications, and to allow for faster responses to user requests for both locating and transferring a resource. Exploiting locality is also beneficial for Internet service providers (ISP): P2P friendly solutions are currently being sought, and many P2P solutions propose the minimization of transit costs incurred by providers by exploiting traffic locality. In this case, locality awareness in the creation and maintenance of the application network topology leads to economical benefits for the ISPs that provide access to customers who are primarily interested in using P2P based applications. We introduce two measures to capture the mismatch between the two topologies. For ultra-peer \( u \) in Autonomous System (AS) \( as_u \), we denote as \( sn_u \) and \( sl_u \) the number of its neighbor ultra-peers in the top-overlay network and the number of its leaf-peers that belong to \( as_u \). We define
the average fraction of neighbor ultra-peers in the same AS for an ultra-peer with \( k \) connections in the top-level overlay as

\[
lnk = \frac{\sum_{u \in R_k} sn_u}{k \cdot |LU_k|},
\]

and the average fraction of leaf-peers in the same AS for an ultra-peer with \( k \) leaf-peers as

\[
lslk = \frac{\sum_{u \in L_k} sl_u}{k \cdot |LP_k|}.
\]

We then compute the average fraction of close connections as

\[
lsn = \sum_{k=1}^{k_{max}} p_k \cdot lsnk, \quad lsl = \sum_{k=0}^{k_{max}} l_k \cdot lslk.
\]

We computed the values of \( lsn \) and \( lsl \) for all the snapshots of the top battery, and we found that only 3.05–3.88% of the ultra-peers are connected to other ultra-peers that belong to the same autonomous systems, while only 3.18–3.7% of the leaf-peers attached to an ultra-peer belong to the same autonomous system of the ultra-peer. This confirms that Gnutella does not exploit locality awareness in building and maintaining the overlay topology. This contribution confirms the result presented in [8]. However, we must point out that, although the conclusions reported here and in [8] are quite similar in spirit, the two papers address different versions of Gnutella. Each version is characterized by a different architectural structure (the flat structure of Gnutella 0.4 vs. the two-tier structure of Gnutella 0.6).

6. Users’ behavior properties

We studied the behavior of users in the Gnutella community, taking into account several aspects of network topology. First of all, we faced the issue of estimating the length of the users’ connections. The term session is used to indicate a set of consecutive snapshots in which the same ultra-peer is repeatedly connected or disconnected. A connected session is characterized by the presence of the same user in a set of consecutive snapshots, without considering the user’s activity (e.g., query, data transfer, etc.). The standby session of the ultra-peer \( UP \) is identified by the set of snapshots between two connected sessions of such a user. The lengths of the connected sessions have been already studied, with different approaches, in previous works [6,13], while the duration of the standby periods have never been reported in literature, though these results may have important implications for users’ models.

6.1. The duration of sessions

It is well known that users in a file-sharing system tend to join the network for long periods of time, in order to download the contents in which they are interested. These long periods are frequently composed of shorter connected and standby sessions, which might, in theory, follow periodic behaviors. In the following, we provide the CCDF concerning the duration of the whole set of known sessions. Besides, we computed the time periods between successive connections of the same user, in order to provide the distribution of the standby sessions’ lengths.

In this paper, we present results about the duration of ultra-peer sessions because our measurement technique allows us to analyze the connectivity of users only at the first hierarchy layer. The results that we show derive from the analysis of the top battery. We take into account each peer \( UP(ip, port) \), connected for at least one snapshot during our observation, in order to identify its periods of activity and standby. The length of a connected session for the ultra-peer \( UP \) is simply defined by \( L_{UP} = ID^{up}_f - ID^{up}_i + 1 \), where \( ID^{up}_i \) and \( ID^{up}_f \) are the ID of the first and last snapshots of the connected session. Fig. 4a shows the CCDF of the duration of the connected sessions on a logarithmic scale. We did not account for all of the sessions already active in the first snapshot, nor for those still alive in the last measure. The figure indicates that 90% of sessions lasted less than 11 consecutive snapshots (which means a continuous activity period of about 2 h). This result is congruent with the conclusions concerning the size of resources shared in the network, which we discuss in Section 7. In fact, most of the files in the system have a small size, typically about 3.5 MB, which require only a short time to be downloaded.

As a second step, we analyzed the behavior of ultra-peers concerning their standby sessions. We performed
such an analysis by considering the \((ip, port)\) pair as the key in order to identify the same user in different measurements. Such a measurement technique assumes that users tend to use the same pair \((ip, port)\) in joining the network in different time instants. We are aware that this choice could bias the final results, but to the best of our knowledge, this is the only way to perform such type of measure for the Gnutella protocol. In [6], the same assumption has been made by authors. In Fig. 4b, the distribution of the temporal duration between different logins of the same user has been plotted. The figure reveals that, in their whole uptime, 90% of users tend to join the network in less than about 43 h (roughly equivalent to 250 snapshots). Along with the information about the connected sessions, we found that users generally login for short time periods, that is, just long enough to download the contents in which they are interested, though they also join the overlay network quite frequently. In both the reported cases, we computed the best-fitting distributions of the experimental profiles along with an absolute error of less than 0.14. Both of the profiles are shown in Fig. 4a and b, while the parameters of the fitting distributions, to properly model the measured behaviors, are reported in Table 4. The empirical curves result to be approximated by lognormal profiles.

6.2. Estimating the arrival rate of ultra-peers

The conclusions presented in the previous section suggest an easy method for roughly evaluating the arrival rate of users in the Gnutella network. The system can be modeled as a queue with infinite servers, where the service time (duration of the connected sessions) has been measured in Section 6.1 while the inter-arrival rate of customers is unknown. By taking into account the average duration of the connected sessions, along with the average number of users in the network, it is possible to derive an estimate of the arrival rate of customers by applying Little’s law.

The suggested approach can be assumed as correct under the following considerations: (i) such a rate concerns ultra-peers only. We are not able to provide the arrival rate of leaf-peers, as we do not have knowledge about the average time that they spend in the system. (ii) The set of users that we are studying is a subset of the total set of ultra-peers. However, we believe that such an estimate could be roughly extended to all the ultra-peers. (iii) The system can be considered as stable in the in our chosen measurement period (10 days); in principle, the application of such an approach over longer periods could return misleading results, as P2P file-sharing systems are characterized by large evolutions over long time scales. Taking into account the considerations presented in Section 5 regarding the number of ultra-peers in the network, the average number of ultra-peers results \(N_{UP} = 869,259\). Such a number has been computed by averaging the measurement results represented by the solid line (known UPs) in Fig. 2a. Regarding the average time spent by customers in the system, we derive such a result from the distribution plotted in Fig. 4a. The resulting average session duration (ultra-peers’ uptime) is \(T_{UP} = 12.34\) snapshots = 8563.94 s. Applying Little’s law, we can estimate the average arrival rate of ultra-peers in the Gnutella network as

\[
\lambda_{UP} = \frac{N_{UP}}{T_{UP}} = 101.5 \text{ ultra-peers/s.}
\]

At this stage, the empirical distribution of the process cannot be accurately determined by our measurements; however, we could estimate the average arrival rate by applying Little’s law. Such a result can be combined with the conclusions on the duration of sessions (in Table 4), in order to properly model the users’ behavior in a P2P file-sharing system.

6.3. The sharing activity of peers

In our work, we also studied the presence of files in the network from the viewpoint of users, measuring the number of files shared by peers. This study derives from the analysis of the dataset res_battery1, which concerns the resources shared both by peers and ultra-peers. An important remark about this measure is that we only have knowledge of servents (using Gnutella jargon) sharing at least one file (as they reply with "Query Hit" messages to our crawler). In our analysis we cannot take account of servents that do not provide information about their contents, so we are not able to produce results about free-riders. A second remark concerns the percentage of users replying to Query messages originated by our crawler; we verified that this portion has an average population of 513,270 users.

Fig. 5a shows the CCDF using a log–log scale, in order to illustrate the files shared by users across three distinct snapshots. This result shows that 90% of peers share less than 200 files on average, while 9% of users contribute a number of files ranging between 200 to approximately 500 replicas. Only 1% of the total number of peers makes available more than 500 files. The body of the distribution does not indicate a divergence, while significant deviations are clearly visible in the tail. Fig. 5b shows the maximum

<table>
<thead>
<tr>
<th>Type of session</th>
<th>Fitting distribution</th>
<th>Final set of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Lognormal</td>
<td>(\mu = 1.20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\sigma = 1.25)</td>
</tr>
<tr>
<td>Standby</td>
<td>Lognormal</td>
<td>(\mu = 3.42)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(\sigma = 2.00)</td>
</tr>
</tbody>
</table>

2 This number of users (both peers and ultra-peers) refers to the experiments on the shared resources and correctly exceeds the number of responding ultra-peers provided in Section 4, which concerns the experiments on the overlay topology.
number of files shared by users and can clarify the behavior of users sharing many contents. The figure shows that 0.0001% of users makes available more than 2000 files; although such a percentage is extremely limited, it refers to more than 340 peers of our analysis. However, the number of shared resources never exceeds 6200 files.

We estimated deviations among the three measures by performing (and comparing) their best-fitting distributions. The empirical profiles are fitted by a lognormal distribution, with the following parameters: $\mu = 3.47 \pm 0.02$, $\sigma = 1.44 \pm 0.01$. The reported (almost negligible) standard deviations take into account the distortions among distinct observations, proving the consistency of our measures.

7. Shared resources properties

In this section, we present three different analyses pertaining to the resources shared by peers. Similarly to [13], we identified files using the content hash (SHA1) returned by the contacted peers that represents the most reliable way to unequivocally identify files. Nevertheless, we are aware that this approach does not solve the issue of different versions/variants of the same file: using this approach, such files are inevitably identified as distinct elements. Section 7.1 presents the analysis of static properties of files regarding the number of shared files in the network, considering both replicas and distinct file hashes. These measures allow researchers to investigate file replication in the Gnutella network. We found that most of the files have just one replica, while the tail of the distribution suggests that behavior is accurately fitted by the Zipf's curve. Furthermore, we report on the size and the type of shared files, with the aim of updating results published in [12,13] nearly 2 years ago.

Section 7.2 is focused on the properties of newly created files shared by users in the Gnutella network. We characterize files in different classes of popularity depending on the number of replicas that we compute during our observation period; we also estimate the probability that newly created files could belong to a particular class of popularity. Furthermore, we provide the average rates of birth of newly created files in the network, referred to as different classes of popularity.

Section 7.3 presents a study on the dynamics of file popularity in order to characterize files whose popularity is increasing or decreasing over time (i.e., becoming more or less popular).

7.1. Static properties of files

We first provide results regarding the amount of contents shared in the overlay network. Next, we focus on the distribution of file types and size and we present results extracted from one reference snapshot. We found that our conclusions were fairly consistent across our whole set of observations.

7.1.1. The number of files shared in the network

We first turn our attention to the amount of contents shared in the overlay network. In particular, we provide an estimate of both the number of unique hashes (files) and of the total number of files. As we are primarily interested in the contents that generate the most traffic in the network, we excluded from our analysis all of the files...
with only one replica, which did not receive downloads during our observation interval. Interestingly, contents such as these accounted for approximately 50% of the total number of unique files in the system. The average number of distinct hashes in the Gnutella network, for files that received at least one download during our observation, is 6,271,570. By taking into account the replicas of every file, the resulting total average number is 36,362,400. We also measured the number of replicas for every file across many snapshots in order to provide a distribution for modeling file replication; to this end, we focused on the number of files that reached at least two replicas in the observation window. Fig. 5c shows the CCDF of the number of replicas. The results plotted in Fig. 5c shows that 85% of files have one replica only, even if we know that they will reach at least two replicas in our observation period. From the analysis of the sequence of snapshots, we can roughly estimate that fewer than 1,000,000 files out of 6,271,570 on average in the Gnutella network have more than two replicas.

Moreover, we can also determine that the average number of single resources is roughly two times the 6,271,570 files. Such an estimate shows that only a small percentage of files are shared by many users, and that most of them receive at most one download. Furthermore, Fig. 5c confirms that the correct distribution for modeling file replication should exhibit a long tail, highlighted by the linear behavior in a log–log scale. By fitting the tail of the empirical distribution, we found that file replication in Gnutella is still described by the linear shape of a Zipf-like profile, with slope \( a = 1 \). Our recent result, along with the measurement studies performed since 2001 [10], confirmed the stability of important features of the Gnutella community.

7.1.2. Results about the type and the size of files

We also report data on the distribution of size and file types of shared resources in the Gnutella network. The results in this section should be considered as an update of those presented in [3,16] almost 2 years ago.

In Figs. 6a and b we depicted the CCDF distribution of file size in both linear–log and log–log scale; Fig. 6a indicates that most of the files are a few megabytes in size (in particular, the curve has a knee between 1 and 10 MB). Furthermore, 10% of them are smaller than 1 MB in size, while a very small fraction of files are larger than 1 GB. Our measurements are consistent with the conclusions published in [16], wherein authors found that the average file size is 3.544 MB, which is the typical dimension of a few minutes long MP3 file.

We also computed the distribution of files across different types of formats, in order to understand which contents were the most popular, that is, the most shared and requested within the Gnutella community. Fig. 6c depicts the contribution of the top-20 file types in the total number of files. The majority of files comprises MP3 files, while the remaining file types account for a small fraction of all files. MP3 audio files are still extremely popular in the Gnutella network, making up about 80% of all files. The histogram of the amount of contributed bytes by each file type (which has not been reported) highlights the load of video files: although they represent only 1% of the total number of files, they account for 23% of the overall amount of shared data in bytes. Such conclusions are congruent with the results provided in [16,3], which derive from measurements conducted almost 2 years ago with respect to our current set of observations. Therefore, we can claim that, regarding the sharing of contents, the Gnutella

![Fig. 6.](image-url)
network is a relatively stable community that did not show
evidence of major modifications over a long time frame
(i.e., almost 2 years).

7.2. Characterizing the new resources

We approached the issue of the birth of new files in the
network from two different viewpoints. (i) The probability
that a new file will reach a certain popularity value will be
analyzed in Section 7.2.1; (ii) the rate of births for different
classes of files identified by their maximum reached popu-

larity will be derived in Section 7.2.2.

7.2.1. Classes of popularity

We describe our study of the probability that new re-
sources will reach a certain level of popularity after their
birth. We performed this study in order to characterize
the transfer of files, as a function of the interest that they
will arouse in users of the system. We introduce a metric
to define the popularity of one resource across the entire
series of snapshots. We assign a parameter $\psi$ to each
new resource, where $\psi$ is defined as the maximum number
of users sharing the file in the available sequence of snap-
shots.\textsuperscript{3} Such a metric is independent of variations in the
number of users between different snapshots, and, further-
more, it is characterized by two important properties: (i) it
allows a comparison between different resources, as the
snapshots have similar sizes, and (ii) its results are partic-

ularly meaningful for files born during our measurement
period, as it provides an estimate of their diffusion rate
in the network.

We consider classes of popularity to be defined as fol-

ows: popularity class $n$ identifies resources whose $\psi$ val-

ues are such that $n = \lfloor \log_2 \psi \rfloor$. In other words, class $n$
defines the new resources whose number of replicas falls
within the range $(2^{n-1}, 2^n]$.

We evaluated the level of popularity that new files
reach in a 15-day period, and we repeated the same anal-
ysis, increasing the length of the time interval up to 2
months. We performed our analysis on the dataset res_bat-
tery1, and we also exploited res_battery2, with the aim of
verifying whether new resources in res_battery1 were, in
fact, novel for the Gnutella community. We have studied
the behavior of approximately 10,000 new resources by
measuring the class of popularity that they reached after
a certain amount of time (about 15, 30, 45 and 60 days).
We took into account only file classes with $n \geq 4$, for
which we could derive an analytical expression fitting
the measured behavior.

In Fig. 7a, we plot the calculated probabilities. We re-
ported single profiles for each time interval with the aim
of highlighting the similarity among the different behav-
iors. More precisely, the distribution describing the 15-
day observation period has a minor load in the tail because,
by taking into account a short time interval, the probability
that a file will attain a high diffusion rate is quite limited.
For all the distributions, we could verify some regularities,
reporting that the ratio between the number of resources
in contiguous classes was close to 2. Such a conclusion
means that the probability that a file will belong to a pop-
ularity class is halved when the file class $(n)$ is increased,

at least for classes with $n \geq 4$.

Assuming that this behavior is reproduced over longer
(greater than 60 days) time intervals, we could find an ana-

lytical expression describing the probability that a new file
belongs to one particular popularity class:

$$P(class = n | n \geq 4) = 2^{\left\lfloor \log_2 \frac{1}{2.02} \right\rfloor}.$$ \hfill (1)

In Fig. 7a, we also depicted the curve in Eq. 1, which clearly
fits the empirical profiles.

Furthermore, we repeated the same measure for files
born at the beginning of our observation, while taking into
account the whole measurement time for the files’ diffu-

sion. Also, in this case, we could verify a regular behavior
from the 4th up to the 12th class (as shown in Fig. 7b).
In the successive classes, the empirical profiles are lower
than the theoretic profiles. We justified these deviations
because, over long time scales, the behavior of file replica-

tion is not just determined by the file diffusion, but it is
deeply biased by the file deletion processes, which de-
crease the probability that one file could reach a large
diffusion.

This study of file popularity classes spurred us to sug-

gest the following meaningful categories of files:

- The Negligible files, which do not receive downloads and
remain confined in their original place (i.e., the peer that
first created the resource). Such files belong to the class
$n = 0$.

- The Rare files, with a limited number of replicas, being
classified by $n \in [1, 3]$.

- Significant files, which have exceeded the threshold of
eight replicas, meaning $n \geq 4$, in at least one snapshot.
Such files represent the 0.2 per million of observed
resources.

In addition to the analysis concerning the Significant
files, we also investigated the class of Rare files, although,

in such cases, we could not verify many interesting
behaviors. The analysis of the probability that such files
belong to one popularity class remains difficult, as the
Rare files are characterized by extremely discontinuous
behaviors. They do not reach high numbers of replicas
and, moreover, they often leave the overlay for long time
periods. Hence, it is not possible to distinguish whether a
file represents a new resource for the users or, otherwise,
if it was already known before our observations. An accu-
rate analysis could be performed by applying the same
technique that we developed for the Significant files,

but taking into account a higher number of observations.
In the next section, we present an insight into the rate of
birth of new files, belonging to the categories of Rare and
Significant files.

7.2.2. The rate of birth of new files

For modeling purposes it may be assumed that the
birth of Negligible, Rare and Significant files are described

\textsuperscript{3} $\psi$ can also be intended as the maximum number of replicas reached by
that file over the whole set of measures.
by independent processes, whose distinct parameters can be computed separately. Because files in the Negligible category will not generate any data transfer, the birth rate for this category would be of little interest. Therefore, we focused on the Rare and the Significant resources. The measurement methodology must take into account the finite horizon of the observation, which inevitably biases the measure. In particular, two main effects must be considered:

(i) The knowledge of the system in the first snapshots of the network is inadequate for identifying new resources; hence, a transient observation period is needed to establish a set of known resources. A limited knowledge of the files shared in the network requires that many resources be classified as new, even if they could have belonged to the system in time preceding our measures. Consequently, the birth rate of new resources is overestimated (being biased by the limited knowledge of the network). Furthermore, in the first observations, each snapshot could describe only a limited part of the global network, as many sharing peers might be disconnected. In other words, if a resource has never been listed in our snapshots, it does not necessarily mean that it is a new resource. The initial transient snapshot sequence is invaluable for building out our knowledge of the shared resources. However, it cannot be considered to be a valid observation period for measuring the birth rate of files.

(ii) On the other hand, the last snapshots of the observation horizon could be too short to allow for the diffusion of files. Such a temporal limit could mislead the forecast of popularity that a shared resource would be able to reach, biasing the classification of resources as Negligible, Rare or Significant (because of this, the birth rate of new resources is underestimated). Also, in the last observations, the measure must not be performed. This final period of observation must be considered for allowing the diffusion of resources after their births.

For each snapshot, we counted the number of different files that were never collected before that snapshot and divided it by the time elapsed since the previous snapshot. To reduce the distortion caused by the limited observation period, we computed the rate of Rare and Significant file births for each snapshot and averaged their birth rate in the set of snapshots whose id is in the interval $i = [100, 350]$. The last set of measures for $i > 350$ is taken into account in order to allow for file diffusion. As already stated, in this last region (see Fig. 7c) the rate decreases because files do not have enough time to reach their predicted popularity. Only the fastest ones can reach the

![Fig. 7.](image-url)
popularity value for their reported file class. We repeated the same procedure for the class of Rare files, and we found the same steady-state interval for snapshots. As a final result, we computed the rate of birth of new files for the selected categories. For the Rare files, the measured average rate is 3977.07 files/h. The birth rate for the class of Significant files is only 18.65 files/h. The global rate of files (taking into account both Rare and Significant files) derives from the algebraic sum of the single rates, resulting 3995.72 files/h. The reported results prove that accurate modeling of resource birth should consider files as partitioned in multiple categories, each of them depending on a popularity profile that determines its diffusion in the network.

7.3. Analyzing file popularity dynamics

For this analysis, we exploit the set of snapshots in res_battery1: a such dataset allowed us to explore dynamic properties over short timescales (days and weeks) up to the entire measurement period (2 months). We examined the popularity variations for files belonging to classes with \( n \geq 6 \) (i.e., reaching at least 32 replicas during their lifetime) yielding a sample in the dataset whose size is equal to 239,729. For this analysis, we define the popularity of a file as the number of replicas of that file in the system, normalized by the total number of files present in the relevant snapshot. Such a definition eliminates the effects of a varying peer population across different snapshots and is analogous to the one used in [16].

We performed a basic analysis concerning the popularity changes of every file, taking into account two simple parameters: (i) \( \delta \), defined as the change in popularity of a file between the beginning and the end of the measurement period, and (ii) \( \delta_{Mm} \), defined as the gap between the maximum and the minimum value of popularity for every file.

We first consider \( \delta_{Mm} \), in order to understand how popularity variations can affect files in a measurement period across 2 months. Fig. 8a shows the probability density function of \( \delta_{Mm} \) (in logarithmic scale) for the whole set of files, which provides an evidence about the predominant small variations in the popularity of files (the first bar value is 96.74%). Taking into account the entire set of files, \( \delta_{Mm} \) is distributed between 0 and 695, where only 2.14\% of the files (corresponding to 5132 files) are characterized by \( \delta_{Mm} > 10 \). Such popularity values, reported in Fig. 8a, have been multiplied by \( 10^6 \), in order to make the numbers easier to manage. Taking into account the fact that the number of users replying to query messages was approximately 513,270 (on average), it is possible to estimate that a popularity variation \( \delta \) is equivalent to a gap of \( \delta/2 \) in the number of replicas. Such an estimate means that \( \delta_{Mm} = 10 \) roughly corresponds to five replicas only. We then assumed that the set of files with \( \delta_{Mm} < 10 \), which represents 97.86% of the total, can be classified as constant.

In order to identify popularity profiles, we introduced the following parameter: \( \rho = \delta/\delta_{Mm} \), where \( \rho \) ranges in the interval \([-1, 1]\). This parameter is a good estimator of the shape of popularity curves. For \( \rho \approx 1 \), we are in the case of increasing profiles. In fact, the popularity gap between the maximum and the minimum points is close to the gap between the first and the last measures, which results positive, being \( \delta > 0 \). In the same way, \( \rho \approx -1 \) identifies decreasing curves. The case of \( \rho \approx 0 \) can indicate both constant and variable profiles; however, \( \rho \approx 0 \) coupled to high values of \( \delta_{Mm} \) can highlight interesting profiles, such as sinusoidal, concave or convex curves.

By inspecting the selected profiles, we found that files with significant increasing values of popularity were properly characterized by \( \rho > 0.5 \), while decreasing values of popularity have \( \rho \leq -0.5 \). According to the reported classification metric, we could identify the number of files with increasing (957 of 239,729) and decreasing (410 of 239,729) profiles. As a second step, we tried to provide a deeper investigation into the set of 3765 (1.57%) files characterized by \( \delta_{Mm} > 10 \) and \(-0.5 < \rho < 0.5 \), with the aim of identifying files with typical profiles. We analyzed all profiles and can now claim that they can be properly included in the class of files with constant popularity, which sometimes experiences wide oscillations around the mean. Even significant values of \( \delta_{Mm} \) do not indicate interesting behaviors; they simply result from large oscillations, which bias the popularity profiles.

Lastly, we conclude that in modeling a large P2P file-sharing systems, file popularity profiles can be taken into account for a small set files only. More that 99\% of files report constant behaviors that can be modeled by aggregating their popularity in classes. About 1\% of files are characterized by increasing and decreasing profiles, which...
8. Conclusions and further developments

In this paper, we presented an active measurement study regarding the characteristics of peers and files in the Gnutella file-sharing community. We developed a parallel crawler that captured accurate snapshots of this large scale P2P system, in which each peer is annotated with its available files.

The analysis presented here addressed several issues concerning the typical characteristics of a P2P file-sharing community, such as overlay network topology, the behavior of users and the characteristics of shared files. We reported results that represent an update for the conclusions drawn by several authors in previous work on this topic, as well as a set of novel results that, to the best of our knowledge, have not yet been discussed.

Crawlers are designed for one specific application, and because of this, adapting them to be used with other P2P communities is not painless. For instance, if we were to develop a crawler for the BitTorrent [17] community, we would be required to heavily modify the architecture and the internal details of our software. This is because BitTorrent is tracker-centric and most of the information on the set of active peers should be obtained by repeated queries to the file tracker. Last but not least, the overlay network that we explored in Gnutella is used for managing queries for shared resources, while in BitTorrent, active peers organize into an overlay network for data transfer. The constant search (due to the for-tat strategy employed by BitTorrent peers) for high download/upload rates might make this overlay network even more dynamic than Gnutella, which would require to devise more sophisticated approaches to reduce the crawling time.

On the other hand, P2P communities in which participants are able to provide information about their neighborhood are better suited to be studied with reasonable modifications to our Gnutella crawler. In particular, we recently started to design and plan measurements for P2P-based IP television (IPTV) networks. To this end, we started to develop a parallel, multi-threaded crawler of the PPlive application that is one of the most widespread IPTV platforms. We leveraged the experience that we gained in developing our Gnutella crawler by designing the PPlive crawler using the same software architecture and most of the analysis software. Nevertheless, the peculiarities of IPTV applications forced a major re-design and implementation of the internal communication modules. IPTV applications are now often proprietary, thus forcing researchers to perform complicated reverse engineering of the network traffic by passive sniffing of controlled clients. Finally, as in BitTorrent, IPTV peers organize into an overlay network for data (video and audio) transfer which we found in preliminary experiments to be extremely dynamic due to the gossiping algorithms employed by peers to discover neighbors from whom they are downloading (and for whom they are uploading) data chunks. For this reason, specific solutions to speed up the crawling are currently being developed.

Acknowledgements

This work has been partially supported by the Italian Ministry for University and Research (MIUR) within the frameworks of the PROFILES (PRIN) and of the RECIPE (PRIN) projects.

References


R. Bolla received the “Laurea” degree in Electronic Engineering in 1989 and the Ph.D. degree in Telecommunications in 1994 from the University of Genoa, Genoa, Italy. Since September 2004, he has been Associate Professor of Telematics at the University of Genoa, where he is with the Department of Communications, Computer and Systems Science (DIST). He has been PI in a number of relevant research projects and contracts in the telecommunications field. He has co-authored over 130 scientific publications in international journals and international conference proceedings. His current research interests are in: (i) resource allocation, routing and control of IP QoS (Quality of Service) networks, (ii) modeling and design of TCP/IP networks, (iii) P2P overlay network traffic mea-
surement and modeling, (iv) advanced platforms for software routers, (v) performance and traffic monitoring in IP networks, and (vi) advanced mobility management in heterogeneous wireless networks.

**R. Gaeta** was born in Torino, Italy in 1968. He received his Laurea and Ph.D. degrees in Computer Science from the University of Torino in 1992 and 1997, respectively. He is currently Associate Professor at the Computer Science Department of the University of Torino. His research interests include the use of formal models for dependability analysis of time critical systems and the performance evaluation of peer-to-peer computing systems and large scale sensor networks.

**A. Magnetto** was born in Rivoli in 1984. He received his first level Laurea degree in Computer Science in 2006 from the University of Torino. He is currently Master Student at the University of Torino, where he is developing a peer-to-peer architecture to support real time streaming. His main research interests include network modeling/simulation, with particular interest in peer-to-peer overlay networks.

**M. Sciuto** received the “Laurea” degree in Physics in 2001 from the University of Genoa, Italy. Since January 2005, he has been with the Department of Communications, Computer and Systems Science of the University of Genoa, working with Prof. Raffaele Bolla and Prof. Franco Davoli, as a Ph.D. candidate. He just submitted his Ph.D. Thesis “Analysis, characterization and modeling of application layer traffic in large scale overlay networks”, that he is going to defend on the next 28th of April. Since April 2008, he is a Research Assistant at the Department of Communications, Computer and Systems Science of the University of Genoa. He was a visiting Ph.D. student starting from July 2006 to March 2007 at the Computer Science Department of the University of Canterbury in Christchurch, New Zealand, where he worked with Prof. Krys Pawlowski. During 2004 and 2005, he worked for the Metropolitan Area Network of Piacenza, Italy, as responsible for network operations. As from 12/2001 until 02/2004, he worked as a network engineer in GARR (the Italian Academic and Research Network, www.garr.it) where he played an active part in traffic monitoring projects. His main research interests include: (i) active/passive network monitoring and traffic characterization in high speed networks and (ii) network modeling/simulation, with particular interest in peer-to-peer overlay networks.

**M. Sereno** was born in Nocera Inferiore, Italy. He received his Laurea degree in Computer Science from the University of Salerno, Italy, in 1987, and his Ph.D. degree in Computer Science from the University of Torino, Italy, in 1992. He is currently Full Professor at the Computer Science Department of the University of Torino. His current research interests are in the area of performance evaluation of computer systems, communication networks, peer-to-peer systems, sensor networks, queueing networks and stochastic Petri net models.