On the Long-term Evolution of the Gnutella Network

Amir H. Rasti, Daniel Stutzbach, Reza Rejaie
University of Oregon

Abstract

The popularity of Peer-to-Peer (P2P) file sharing applications has significantly increased during the past few years. To accommodate the increasing user population, developers gradually evolve their client software by introducing new features, such as a two-tier overlay topology. The evolution of these P2P systems primarily depends on (i) coherency across different implementations and (ii) the availability of these feature among clients throughout the system.

This paper sheds some light on the evolution of client and overlay characteristics in the Gnutella network during a one year period over which it tripled in size, exceeding two million concurrent peers. Our results show several interesting trends including (i) most users quickly upgrade their software which provides great flexibility for developers to react to changes and (ii) as the system has grown in size the top-level overlay has remained properly connected but there is growing imbalance between the two levels.

1 Introduction

Contrary to the common assumption about the limited scalability of unstructured Peer-to-Peer (P2P) file-sharing applications, major P2P file sharing applications (i.e., FastTrack (or Kazaa), Gnutella, and eDonkey) have become significantly more popular during the past few years and now contribute a significant portion of total Internet traffic [1, 5]. For example, over the past 12 months, the Gnutella network has grown in size by a factor of three to more than 2 million simultaneous users. To scale with this growth in user population, different group of application developers have introduced a two-tier overlay topology coupled with more efficient search mechanisms (e.g., Dynamic Querying in Gnutella) in their client software. Given the P2P (and often open-source) nature of these systems, the impact of a newly introduced feature depends on two factors: (i) the coherency (or compatibility) of features across different brands of client software and (ii) the penetration rate and availability of a feature throughout the system. As new features become available, users upgrade their client software and the new features gradually spread throughout the system. Therefore, the penetration rate of new features primarily depends on which brands implement the feature and the ability and willingness of users to upgrade.

In this paper, we examine the evolution of the Gnutella network during the past year over which its overall population has tripled to more than two million simultaneous peers. Toward this end, we explore the evolution of the Gnutella network across two related angles:

- **Client Characteristics** that include changes in the overall population, the distribution across different geographic regions, and the popularity of brands and versions.
- **Overlay Characteristics** that represent changes in various properties of the two-tier overlay topology including balance between the populations at each tier, different angles of node degree, intra-regional bias in peer connectivity, and resiliency to node removal.

Modern Gnutella has incorporated a two-tier architecture where a small subset of peers, called ultrapeers, form a top-level overlay while other peers, called leaf peers, are connected to the top-level overlay through one or multiple ultrapeers (Figure 1(a)). When a leaf peer cannot locate a sufficient number of parents in the top-level overlay, it will promote itself and become an ultrapeer if it believes that it is not firewalled. The goal of this mechanism is to maintain a proper balance between ultrapeers and leaves (15%–17%).
Our main findings are summarized in the following list. Due to the limited space, we only present a subset of our results. Further details can be found in the related technical report [6].

- Gnutella is North America-centric, with a moderate number of European users.
- There is a strong bias towards intra-continent connectivity, especially in continents with smaller population.
- Most users upgrade their client software within 2 months, which is much faster than users upgrade operating systems and other types of software.
- Peers in the top-level overlay can effectively locate the maximum number of neighbors and form a well-connected top-level overlay.
- The ultrapeers-to-leaf ratio has unnecessarily increased, degrading performance of the system. This also implies that leaves are unable to locate available open slots among ultrapeers, and thus unnecessarily promote themselves to become ultrapeers.

Several previous studies have examined characteristics of overlay topologies [3, 4, 7, 12] and participating peers [8, 9] in large scale P2P systems. However, there are at least three important differences between our work and these prior studies as follows: First, all the previous studies (except our earlier work [12]) were conducted more than three years ago on much smaller user populations and before the introduction of the two-tier overlay topology. Second, and more importantly, most of these studies focused on characteristics of P2P systems over a relatively short period (e.g., a few days or weeks) and did not capture any major growth in population. Finally, previous studies have used either partial or distorted snapshots of the system that could significantly affect the accuracy of their results [10]. To our knowledge, our work is the only study that examines the evolution of a P2P file-sharing system over a one year period. This paper builds on our recent work of characterizing graph-related properties of the Gnutella overlay and its short-term dynamics [12], by exploring long-term trends and effects.

The rest of this paper is organized as follows: In Section 2, we briefly present our data collection methodology and tools, and explain the importance of capturing accurate snapshots of P2P system. Sections 3 and 4 present the evolution of client characteristics and overlay characteristics in the Gnutella network, respectively. Finally, we conclude the paper and sketch our future plans in Section 5.

2 Data Collection

To accurately characterize P2P overlay topologies, we need to capture complete and accurate snapshots. By “snapshot”, we mean a graph that captures all participating peers (as nodes) and the connections between them (as edges) at a single instance in time. The most common approach to capture a snapshot is to crawl the overlay. In practice, capturing accurate snapshots is challenging due to the large size and the dynamic nature of P2P systems. Because overlays change as the crawler operates, captured snapshots are inherently distorted where the degree of distortion is proportional to the crawling duration [11].

We have developed a set of measurement techniques into a Gnutella crawler, called Cruiser [10]. Cruiser improves the accuracy of captured snapshots by significantly increasing the crawling speed primarily through two mechanisms. First, it leverages the two-tier structure of the overlay by contacting only ultrapeers. Since leaf peers connect only to ultrapeers, all of their topological information can be captured without contacting them directly. Second, Cruiser significantly increases the degree of concurrency in crawling by running on several machines and opening hundreds of simultaneous connections from each machine.

Cruiser can capture the Gnutella network with 2.2 million peers in around 8 minutes, or around 275 Kpeer/minute (by directly contacting 22 Kpeer/minute). This is orders of magnitude faster than the fastest previously reported crawlers (i.e., 2.5Kpeers/minute in [8])\(^1\). Cruiser captures the following information from each peer it successfully contacts: (i) peer type (ultrapeer or leaf), (ii) brand and version, (iii) a list of the peer’s neighbors, and (iv) a list of an ultrapeer’s leaf nodes. Since the crawler does not directly contact leaf peers, we do not have information about their brand and versions.

\(^1\)Capturing snapshot of a large scale P2P system with a slow crawler might be infeasible since the crawler may never terminate due to ongoing arrival of new peers in the system.
We have captured more than 80,000 snapshots of the Gnutella network with Cruiser between Oct. 2004 and Oct. 2005. To examine the long-term trends in the Gnutella network, we selected nine snapshots roughly evenly spaced during this period\(^2\). To minimize any possible error due to the time-of-day or day-of-week effects, our candidate snapshots were all taken around 3pm PDT on weekdays.

3 Client Characteristics

In this section, we explore changes in several characteristics of Gnutella peers during the past 12 months. Figure 1(b) depicts growth in the overall population of Gnutella peers and its break down between ultrapeers and leaves. This figure shows that Gnutella has tripled in size, with surprisingly linear growth, though somewhat slower during last year’s holiday season (Dec. ’04–Jan. ’05).

Client Location: We also examined the breakdown of ultrapeers across different regions and countries using GeoIP 1.3.14 from MaxMind, LLC. Figure 1(c) shows the distribution of Gnutella clients across four regions, namely North America (NA), South America (SA), Europe (EU), and Asia (AS) that collectively make up 98.5% of the total ultrapeer population. This figure reveals that a majority of Gnutella ultrapeers are in North American (80%), with a significant fraction (13%) in Europe. Furthermore, the user population of different regions have grown proportionally. Gnutella is mostly US-centric and may not properly represent a more international P2P system. We use this information to examine connectivity within each region in the next section.

The distribution of user populations across different countries has also grown proportionally, except for China where client population has dropped significantly (94%).

Clients in US, Canada, and UK make up 65%, 14%, and 5% of the total population, respectively\(^3\). The remaining countries made up less than 2% each, but make up 16% in total. Thus, while the Gnutella network is dominated by predominately English-speaking countries, around one-fifth is composed of users from other countries.

Figure 2(a) shows the effect of different time zones on the distribution of user population of different regions. The population of North American and European clients peak at around 7pm and 11am PDT with 86% and 24%, respectively. This figure indicates that our 3pm snapshots capture roughly average daily population, i.e., not at any of the peaks.

Client Implementation: Figure 2(b) depicts the breakdown of ultrapeers across the major brands that implement Gnutella. This figure shows that the two most popular implementations are LimeWire and BearShare. Overall, the ratio between LimeWire and BearShare has been fairly stable, with LimeWire making up 75–85% of ultrapeers, BearShare\(^4\) making up 10–20%, and other brands making up 3–7%.

Gradual upgrading by users implies that dif-

---

\(^2\)Unfortunately, we did not capture any snapshots during May or June of 2004.

\(^3\)These values are from the snapshot taken on 9/20/05 and are similar to the other values observed during the study period, as shown in Figure 1(c).

\(^4\)BearShare clients support more leaves per ultrapeer, and thus tend to have fewer ultrapeers. Therefore, while our results accurately represent the top-level overlay, they could potentially under-represent BearShare users.
different versions of each brand coexist at any point of time. P2P systems may need to evolve quickly in order to accommodate growing (and even changing) user demand. Otherwise, users may not observe acceptable performance and leave the system. This raises the following fundamental question: “How rapidly and effectively can a widely-deployed P2P system evolve in order to cope with increasing user demand?” Furthermore, how is this evolution affected by the heterogeneity among coexisting implementations (and versions) and the rate of software upgrade by users?

Since LimeWire clients make up an overwhelming majority of ultrapeers, we explored the breakdown among popular versions of LimeWire. Figure 2(c) shows the percentage of LimeWire ultrapeers running each version, revealing that within 2 months of the release of a new version most LimeWire users are running it. This is illustrated by the way the market share of a version increases from 0% to more than 50%, and only decreases when a new version appears. This behavior can be attributed to the automatic notification of new versions coupled with the simplicity of using the P2P system for distributing updates quickly. The quick upgrade by users also implies that new features rapidly become widespread throughout the system. Due to the rapid deployment of new versions, “flag days” are practical in P2P systems where new clients are configured to use a new, incompatible feature on a particular date.

4 Overlay Characteristics

In this section, we turn our attention to the evolution of the two-tier overlay topology during the past year.

Ultrapeer-Leaf Ratio: Figure 4(a) presents the changes in the percentage of ultrapeers during the past year. While this percentage fluctuates, it exhibits an overall increasing trend. We recall that leaf peers become ultrapeer only when they cannot locate a sufficient number of ultrapeers that can accept an additional leaf. This result suggests that the ability of leaves to locate available ultrapeers has not scaled as the system has grown in size. Note that the percentage of ultrapeers seems to have dropped by 1.4% between Sep.–Oct. 2005. This drop seems to be correlated with the increase in popularity of LimeWire client version 4.9 (released in Jan. ‘05). We will revisit this issue when we explore the ultrapeer-leaf degree distribution.

Node Degree: To examine changes in the connectivity of the overlay topology, we examine three different angles of node degree distribution in the two-tier overlay: (i) for ultrapeers, the number of ultrapeer neighbors; (ii) for ultrapeers, the number of leaf children; and (iii) for leaves, the number of ultrapeer parents. To show the evolution of the degree distribution over time, we show each angle on the degree distribution for three snapshots that are roughly six months apart. In the absence of other factors, as the population grows, one expects the distribution to change proportionally across different degree values, i.e., the ratio of peers with different degree would remain approximately constant.

Figure 3(a) shows the distribution of the number of top-level neighbors across ultrapeers in a log-log plot. All three distributions show a strong peak in the range of 20 to 30 neighbors,
with a significant number of peers having less than 20 neighbors. Comparison of these three snapshots reveal that the peak has dramatically grown, while the number of peers with fewer than 20 neighbors has increased only slightly rather than proportionally. This implies that despite the dramatic growth in the total population, ultrapeers with open slots for neighbors quickly locate one another and form a well connected top-level overlay.

Figure 3(b) shows the distribution of the number of leaf children across ultrapeers for three snapshots in a log-log plot. In all three snapshots, there are peaks at 30 and 45 children, corresponding to the maximums set in LimeWire and BearShare, respectively. However, unlike the number of neighbors, the peaks have not significantly increased over time. Instead, the dramatic increases have been in the number of ultrapeers with fewer children. This means that there are proportionally more ultrapeers with open slots for more children. This is the direct result of the increase in the ratio of ultrapeers to leaves as shown in Figure 4(a).

Figure 3(c) shows the distribution of the number of ultrapeer parents among leaves in a log-log plot. In all three snapshots, there is a peak at 1–3 children, with many peers having slightly more parents. While the number of peers with 1–3 children has proportionally increased with the population, the number of peers with more parents only exhibit a minor increase. This seems reasonable given the fact that both LimeWire and BearShare clients attempt to maintain 3 ultrapeer parents by default, and peers with less parents are trying to find 3 parents. It also shows that the number of peers with presumably modified implementations have not increased.

**Intra-Region Bias in Connectivity:** While the neighbor selection is generally random in Gnutella, one key question is whether connectivity in the Gnutella overlay topology is geographically-aware. In other words, whether peers in a certain region are more likely to connect to each other than peers in other regions. This is an important issue because it affects the efficiency of the search mechanism. For each one of the main four regions, Figure 4(b) depicts the percentage of neighbors for all ultrapeers in a region that are located in the same region. If there is no bias towards intra-region connectivity, the percentage for each region should be the same as the percentage of the total population that are located in that region (Figure 1(c)). Figure 4(b) reveals that there is a strong bias towards intra-region connectivity, especially within smaller regions. More specifically, even though 13.3%, 2.8%, and 2.3% of the overall population are located in EU, AS and SA, more than 22.9%, 24.5%, and 16% of their neighbors are within the same region, respectively. This biased intra-region connectivity occurs due to three reasons: First, LimeWire client attempts to maintain at least one neighbor with the same locale setting [2], i.e., at least one neighbor whose user speaks the same language. Second, when peers contact others to identify additional neighbors, they contact more peers than actually needed and select those peers that respond faster. This simple mechanism implicitly leads to bias in connectivity within each region. Third, because users in the same region tend to arrive at the same time of day (as shown in Figure 2(a)), they tend to be looking for neighbors at the same time and are more likely to find one another.

This intra-region biased connectivity in the overlay topology implies that users searching for content are more likely to locate desired content.
among other peers in the same region with the same language and culture. Furthermore, response time to queries will also be faster since geography is a good first-order estimator of latency.

**Resiliency to Peer Departure:** Finally, we examine the resiliency of the Gnutella overlay topology to both random and highest-degree node removal (or failure). Figure 4(c) shows the percentage of ultrapeers that must be removed until the largest connected component contains less than 50% of the remaining ultrapeers (i.e., overlay becomes severely fragmented) over the past year. This figure shows that more than 90% of peers must be randomly removed for the overlay to become severely fragmented. Furthermore, the degree of resiliency has remained relatively constant during the past year. Resiliency to the removal of the highest-degree nodes is clearly worse than random node removal. However, overall Gnutella is growing increasingly resilient to highest-degree removal, though with large fluctuations. Since these results are normalized by total population, the actual number of removed ultrapeers has increased by a factor of 3 (i.e., n·50% in Oct. 2004, n·3·60% in Sep. 2005).

**5 Conclusions**

In this paper, we explored long-term trends in the popular Gnutella P2P file-sharing system. By examining how the system has changed over the course of one year, we can see how the change in scale and the introduction of new software versions has impacted the system. Our results show that ultrapeers remain effective at finding one another and filling out their target degree (30), while leaf peers are experiencing gradually increasing difficulty locating available ultrapeers. We also found that within two months, most users are running the latest version of their P2P file-sharing software.

**References**