On the Long-term Evolution of the Two-Tier Gnutella Overlay

Amir H. Rasti, Daniel Stutzbach, Reza Rejaie
Computer and Information Science Department
University of Oregon
{amir, agthorr, reza}@cs.uoregon.edu

Abstract— Peer-to-Peer (P2P) file sharing applications have witnessed a dramatic increase in popularity during the past few years. To accommodate the rapid growth in user population, developers introduced new features in their client software, in particular a two-tier overlay topology. The effect of the two-tier overlay topology in a widely-deployed P2P system primarily depends on the availability and coherency of its implementations among participating clients throughout the system.

This paper sheds some light on the long-term evolution of such a two-tier overlay topology in the Gnutella network during a 15-month period over which the system quadrupled in size, exceeding three million concurrent peers. Our results show two interesting phenomena including: *(i)* During this period, the two-tier overlay has repeatedly begun to lose its balance. However, proper modifications in major client software coupled with the rapid upgrade rate of users, has enabled the developers to maintain the overlay's desired properties. *(ii)* Despite its random connectivity, the Gnutella overlay exhibits a strong bias towards intra-continent connectivity, especially in continents with smaller user populations, that has not changed as the system scaled.

I. Introduction

Contrary to common assumptions about the limited scalability of unstructured Peer-to-Peer (P2P) file-sharing applications, the top-three P2P file sharing applications (*i.e.*, FastTrack or Kazaa, Gnutella and eDonkey) have witnessed a dramatic increase in their popularity during the past few years. For example, the number of simultaneous users in the Gnutella network has quadrupled during the 15 months measurement period. Furthermore, some studies report that the popular P2P file sharing applications make a significant contribution to total Internet traffic [1], [6].

To scale with this rapid growth in user population, major P2P file sharing applications adopted a two-tier overlay topology along with more efficient search mechanisms (*e.g.*, Dynamic Querying [4] in Gnutella). In this two-tier overlay architecture, a small subset of participating peers promote themselves to become *ultrapeers* in a demand-driven fashion and form a *top-level* overlay. Other peers, called *leaf peers*, connect to the top-level overlay through one or multiple ultrapeers (Figure 1). The two-tier architecture attempts to dynamically maintain the following two properties in order to scale with the number of peers while ensuring short pairwise distances between peers as they join/leave the system: *(i)* a proper balance between ultrapeers and leaf peers, and *(ii)* a well-connected top-level overlay where each ultrapeer has a configured number of neighbors. To achieve these goals, participating peers collectively implement two mechanisms: First, a *promotion/demotion* mechanism that determines when a leaf should be promoted to become an ultrapeer and vice versa. Second, an *ultrapeer discovery mechanism* that enables either ultrapeers to find a neighbor or leaf peers to locate a parent in the top-level overlay with available open slots for neighbor or child peer, respectively.

The properties of the two-tier overlay in a widely-deployed P2P system depend not only on the portion of peers that support this feature but also on the coherency (or compatibility) of implementations (and configuration parameters) among participating peers. These properties can be further aggravated in open-source P2P applications since users can arbitrarily change their software. This raises the basic question of: *how can such a fluid two-tier overlay topology effectively accommodate such a rapid increase in peer population despite the heterogeneity of client software while maintaining a short pairwise distance among peers?*

This paper, presents our ongoing investigation to answer the above question by empirically examining the long-term evolution of the Gnutella two-tier overlay topology during the last 15 months over which the user population has more than quadrupled. Using accurate snapshots of the Gnutella overlay, we characterize the following three angles of its long-term evolution: client, graph-related, and geographical properties of the overlay. We explore potential correlation between different observed characteristics and take the steps to identify some of the underlying causes.

Our results illustrate two important points: First, as the Gnutella network has experienced this dramatic increase in user population, the two-tier overlay has repeatedly begun to lose its balance. However, proper modifications in major client software coupled with the rapid upgrade rate of users, has enabled the developers to maintain the overlay's desired properties. Second, despite its random connectivity, the Gnutella overlay exhibits a strong bias towards intra-continent connectivity, especially in continents with smaller user populations. Furthermore, this bias has not changed as the population has quadrupled. *The main contribution of this paper is to illustrate the long-term evolution of a two-tier overlay in a widely-deployed P2P system while it has coped with a significant increase in user population.* While it is extremely difficult to pinpoint the underlying causes of every observed
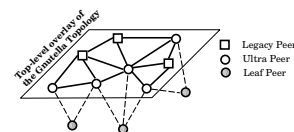

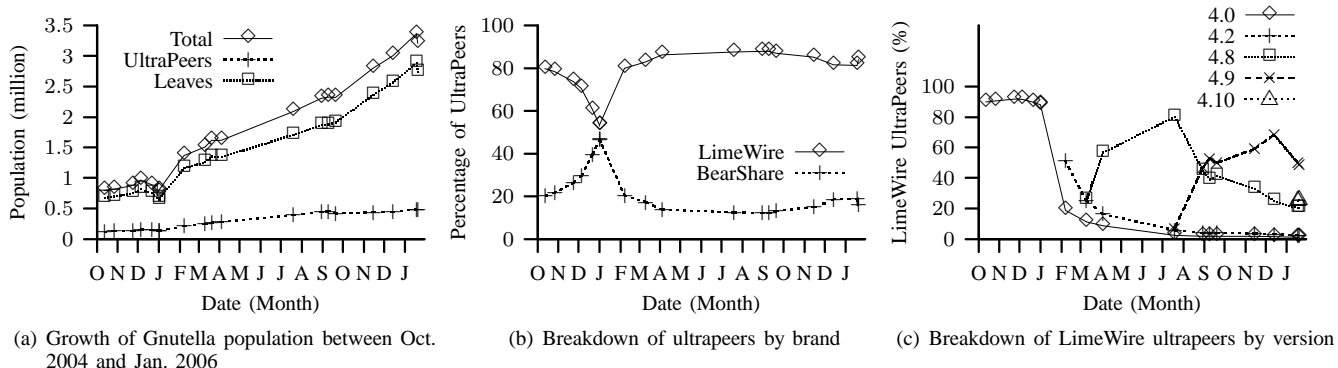Fig. 1.   Gnutella's Two-Tier Overlay Topology

(a) Growth of Gnutella population between Oct. 2004 and Jan. 2006

(b) Breakdown of ultrapeers by brand

(c) Breakdown of LimeWire ultrapeers by version

Fig. 2. Evolution of client properties

characteristic in a large P2P system, this study sheds some light on how P2P overlays evolve.

Related Work: There has been a wealth of research on P2P systems during the past few years. Most of these studies rely on simulation while little is known about the characteristics of widely deployed P2P systems such as Gnutella. A few previous studies have examined characteristics of overlay topologies [3], [5], [7], [13] and participating peers [8], [9] in large scale P2P systems. However, there are two important differences between this work and these prior studies as follows: First, all the previous studies (except our earlier work [13]) were conducted more than three years ago on much smaller user populations and before the introduction of the two-tier overlay topology. Second, previous studies have used either partial or distorted snapshots of the system that could significantly affect the accuracy of their results [11]. To our knowledge, this work is the only study that examines long-term trends of the two-tier overlay topology of a widely-deployed P2P systems.

In our earlier work [13], we characterized graph-related properties of the Gnutella overlay topology across several snapshots (spanned over a few months) in order to provide representative results. This paper complements our earlier work by focusing on long-term trends in the two-tier overlay topology.

The rest of this paper is organized as follows: In Section II, we explain the importance of capturing accurate snapshots of P2P system and briefly present our data collection methodology, our measurement tool and dataset. Section III presents the evolution of overlay properties in the Gnutella network. Finally, we conclude the paper and sketch our future plans in Section IV.

II. 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 instant 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 [12].

We have developed a set of measurement techniques into

a Gnutella crawler, called *Cruiser* [11], which improve the accuracy of captured snapshots by significantly increasing the crawling speed primarily through two mechanisms. First, it leverages the two-tier structure 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 crawler (2.5Kpeers/minute in [8]). Cruiser captures the following information from each peer it successfully contacts: *(i)* peer type (ultrapeer or leaf), *(ii)* brand and version of client, *(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.

DataSet: We have captured around 20,000 snapshots of the Gnutella network with Cruiser between Oct. 2004 and Jan. 2006 [1]. To minimize any possible error on our long-term analysis due to the time-of-day or day-of-week variations in overlay characteristics, we select 18 comparable snapshots that are taken around 3pm PDT on weekdays scattered during our 15-month measurement period [2].

III. Evolution of Overlay Properties

This section, presents the evolution of the two-tier overlay over a 15-month period. In the following subsections, we examine the evolution of three aspects of the Gnutella overlay topology: *(i)* the composition of participating clients, *(ii)* graph-related properties, and *(iii)* geographical properties.

A. Client Properties

Figure 2(a) illustrates the growth in the population of Gnutella network during the past 15 months, and the breakdown of participating peers between the two levels of the overlay. This figure shows that the population has quadrupled during this period. The growth in population has been surprisingly

---

[1] Unfortunately, we did not capture any snapshots during May or June of 2004.

[2] While we do have a huge number of snapshots, the number of *comparable* snapshots is significantly smaller.
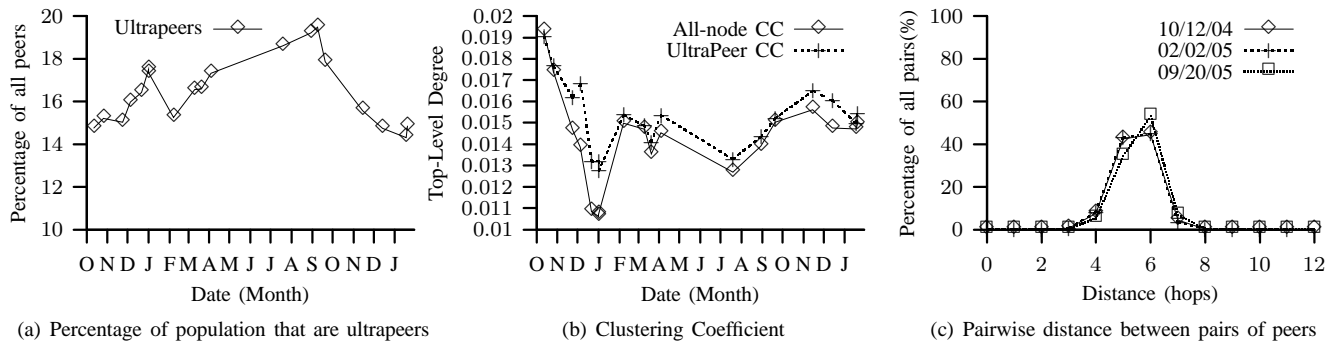
Fig. 3. Evolution of graph properties

(a) Percentage of population that are ultrapeers   (b) Clustering Coefficient   (c) Pairwise distance between pairs of peers

linear with a noticeable dip over the 2004–2005 winter holiday season.

Now, we explore the different varieties of Gnutella clients in use and observe how users upgrade their software as new versions are released. 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[3] making up 10–20%, and other brands making up 3–7%.

Gradual upgrading by users implies that different versions of each brand coexist at any point of time. P2P systems may need to evolve quickly in order to accommodate growing 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?"*

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 quickly 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.

B. Graph-related Properties

We now turn our attention to the evolution of different graph-properties of the overlay topology.

Ultrapeer-Leaf Ratio: A key property of the two-tier overlay is the balance between the population of ultrapeers and leaves. We know that each ultrapeer attempts to maintain 30 leaf children, and each leaf tries to maintain 3 ultrapeer parents.

Given the number of ultrapeers in the system, $|U|$, and the number of leaves, $|L|$, we can reason that there are $30 \cdot |U|$ slots available for leaves, of which $3 \cdot |L|$ are in use. If the ultrapeer-promotion mechanism is working well, and leaves can efficiently locate parents with open slots, then we would see few open slots (δ), i.e., $(30 - \delta) \cdot |U| = 3 \cdot |L|$. For $\delta = 0$, fulfilling this equation yields a mix of 9% ultrapeers and 81% leaves. However, if δ is very small, this indicates that the system is working very hard to keep the balance perfectly despite constant churn in the system. To allow some flexibility, in practice the target percentage of ultrapeers is slightly more than this minimum of 9%, in order to provide some resiliency against dynamics.

Figure 3(a) presents the change in the percentage of ultrapeers during our measurement period. As the population has grown, the percentage of ultrapeers have increased and reached two clear peaks (on Jan. and Sep. 2005), but has dropped back to the expected value (around 15%) in both cases. In Gnutella, leaf peers become ultrapeers only when they cannot locate a sufficient number of ultrapeers that can accept an additional leaf [10]. This increase in the percentage of ultrapeers illustrates the inability of leaves to locate available ultrapeers as the system has grown in size. However, the problem has been apparently addressed in the newer version of the client which led to the drop in the percentage of ultrapeers. There seems to be a correlation between the drop in percentage of ultrapeers in Sep.–Oct. 2005 and the increase in popularity of LimeWire version 4.9, shown in Figure 2(c) and discussed earlier.

Node Degree: To investigate changes in the connectivity of the overlay topology, we examine three different angles of the 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[4]. To show the evolution of the degree distribution over time, we have examined each angle of the degree distribution for all candidate snapshots. However, for clarity of the presented results, we show only four evenly spaced snapshots. The presented trends were similar across other snapshots except where noted.

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

---

[3]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.

[4]We limit the range of node degree to 500 in these graphs. This range includes all but a small percentage of peers ($<0.1\%$) with a higher degree.

(a) Degree distribution from ultrapeers to ultrapeers (b) Degree distribution from ultrapeers to leaves (c) Degree distribution from leaves to ultrapeers
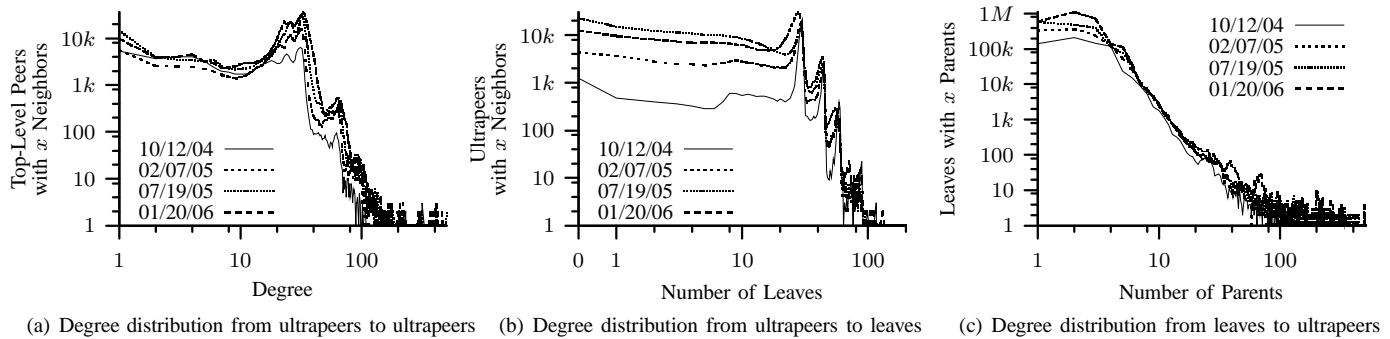
Fig. 4.   Different angles of degree distribution

degree would remain approximately constant. Figure 4(a) shows the distribution of the number of top-level neighbors across ultrapeers for four snapshots in a log-log plot. All four 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 snapshots reveals 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 continue to quickly locate one another and form a well connected top-level overlay.

Figure 4(b) shows the distribution of the number of leaf children across ultrapeers for four snapshots in a log-log plot. In all four snapshots, there are peaks at 30 and 45 children, corresponding to the maximums set in LimeWire and Bear-Share, 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 unnecessary increase in the percentage of ultrapeers as illustrated by the two peaks in Figure 3(a). However, the increasing trend in the number of ultrapeers with open slots has reversed in the most recent snapshots as a result of drop in the percentage of ultrapeers during recent months. Note that the number of peers with fewer children has dropped between the two most recent snapshots in figure 4(b) (*i.e.*, 7/19/05 and 1/20/06).

Figure 4(c) shows the distribution of the number of ultrapeer parents among leaves in a log-log plot. In all snapshots, there is a peak at 1–3 parents, with many peers having slightly more parents. While the number of peers with 1–3 parents has proportionally increased with the population, the number of peers with more parents only exhibits a minor increase. This seems reasonable given the fact that both LimeWire and BearShare clients attempt to maintain 3 ultrapeer parents by default whereas peers with fewer parents are trying to find 3 parents. It also shows that the number of peers with more parents, presumably due to modified implementations, have not increased.

Clustering Coefficient: To examine the degree of clustering in the overlay topology, Figure 3(b) depicts the evolution of the clustering coefficient during our measurement period. Comparing this figure with the population of ultrapeers (Figure 2(a)) shows the clustering coefficient is inversely related to the population of ultrapeers. Since the degree distribution among ultrapeers is relatively fixed, as the number of ultrapeers increases, the top-level overlay becomes more sparse (*i.e.*, a smaller percentage of the possible edges exist), resulting in a lower clustering coefficient.

Pair-wise Distance: The distribution of pair-wise distances among pairs of peers is another interesting aspect of the overlay topology that determines the maximum useful scope for proper reachability in some search mechanisms. Figure 3(c) depicts this distribution between *all* pairs of participating peers for three snapshots during our measurement period[5]. This figure illustrates that the significant growth in the population of peers has led to only a minor increase in the distances between peers. This is not surprising because of the logarithmic effect of population on the distances between peers in randomly connected graphs.

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 5(a) shows the percentage of ultrapeers that must be removed for the largest connected component to contain fewer than 50% of the remaining ultrapeers (*i.e.*, the overlay becomes severely fragmented). This figure shows that more than 90% of peers must be randomly removed from the overlay for it 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. Overall Gnutella is growing increasingly resilient to highest-degree removal. Since these results are normalized by total population, the actual number of removed ultrapeers has increased by a factor of 3 (i.e., $n \cdot 50\%$ in Oct. 2004, $n \cdot 3 \cdot 60\%$ in Sep. 2005).

C. Geographical Properties

While neighbor selection is largely a random process 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 other peers in their region.

Client Location: To characterize this property, first we examined the breakdown of ultrapeers across different regions and countries using GeoIP 1.3.14 from MaxMind, LLC. Figure 5(b) shows the distribution of Gnutella clients across four

---

[5]Since the required processing for pair-wise distances is expensive ($O(n^2)$), we only conducted this analysis for these three snapshots.

(a) Percentage of peers removed to cause severe fragmentation

(b) Breakdown of ultrapeers by region

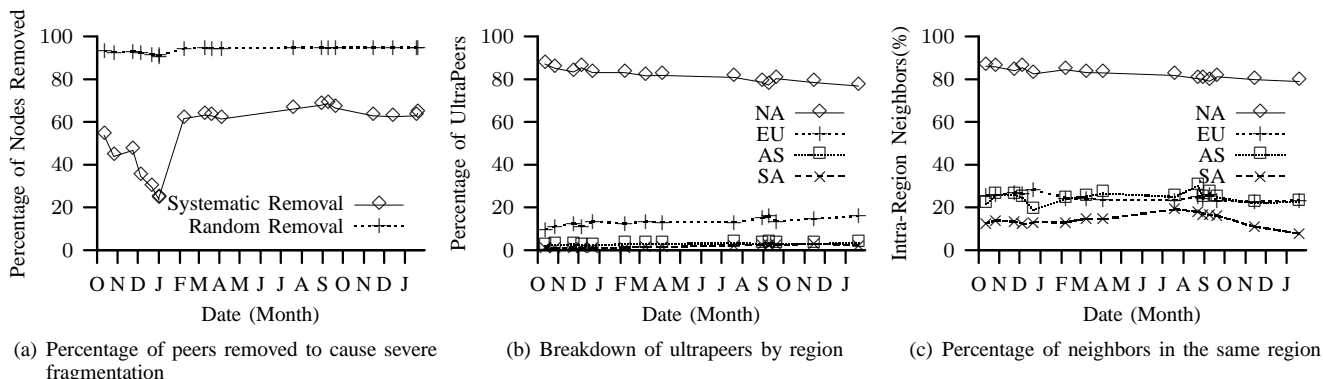(c) Percentage of neighbors in the same region

Fig. 5.   Resiliency and Geographical Properties

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 over time. 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[6]. 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[7].

Intra-Region Bias in Connectivity: For each one of the main four regions, Figure 5(c) 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 5(b)). Figure 5(c) 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 clients attempt 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 are attempting to establish more neighbors, they initiate connections to more peers than are actually needed and select the fastest responders, dropping any extras. This simple mechanism implicitly leads to bias in connectivity within each region. Third, because users in the same region tend to arrive at around the same time of day, their clients tend to be looking for neighbors at the same time and are more likely to find one another. Clearly, one could determine the potential for such a

biased connectivity by examining the source code of various implementations. However, our results quantify the degree of such bias in practice.

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

IV. Conclusions

In this paper, we explored long-term trends in properties of the overlay topology in the popular Gnutella P2P file-sharing system. In particular, we illustrated how the two-tier overlay topology has evolved in order to accommodate dramatic changes in the scale of the user population during the past 15 months. The rapid rate of software updates by participating users enabled developers to effectively modify their software to cope with this moving target and maintain a two-tier overlay with desired properties. We have explored potential correlations between the evolution of overlay properties and the popularity of different versions of major client softwares. Finally, we illustrated the intra-region bias in connectivity among peers.

We plan to continue this study and identify the contribution of various factors to observed properties. Furthermore, we plan to characterize the user-driven dynamics of software updates in further detail. This will enable developers to determine how "tightly" they can control the evolution of the system as it grows over time.

References

[1] slyck.com. http://www.slyck.com, 2005.
[2] G. Bildson. CTO, LimeWire, LLC. Personal correspondence, 2005.
[3] J. Chu, K. Labonte, and B. N. Levine. Availability and Locality Measurements of Peer-to-Peer File Systems. In *ITCom: Scalability and Traffic Control in IP Networks II Conferences*, 2002.
[4] A. Fisk. Dynamic Query Protocol. http://www.the-gdf.org/wiki/index.php?title=Dynamic_Query_Protocol, 2006.
[5] M. Jovanovic, F. Annexstein, and K. Berman. Modeling Peer-to-Peer Network Topologies through "Small-World" Models and Power Laws. In *TELFOR*, 2001.
[6] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy. Transport Layer Identification of P2P Traffic. In *International Measurement Conference*, 2004.
[7] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. *IEEE Internet Computing Journal*, 6(1), 2002.

[6]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 5(b).

[7]We noticed that, 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.

[8] S. Saroiu, P. K. Gummadi, and S. D. Gribble. Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. *Multimedia Systems Journal*, 9(2), 2003.

[9] S. Sen and J. Wang. Analyzing Peer-To-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 12(2), 2004.

[10] A. Singla and C. Rohrs. Ultrapeers: Another Step Towards Gnutella Scalability. Gnutella Developer's Forum, 2002.

[11] D. Stutzbach and R. Rejaie. Capturing Accurate Snapshots of the Gnutella Network. In *Global Internet Symposium*, 2005.

[12] D. Stutzbach and R. Rejaie. Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers. In *Passive and Active Measurement Workshop*, Extended Abstract, 2005.

[13] D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. In *Internet Measurement Conference*, 2005.