

Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems

Technical Report CIS-TR-2005-01

University of Oregon

June 3, 2005

Daniel Stutzbach, Reza Rejaie
University of Oregon
{agthorr,reza}@cs.uoregon.edu

Subhabrata Sen
AT&T Labs—Research
sen@research.att.com

Abstract

During recent years, peer-to-peer (P2P) file-sharing systems have evolved in many ways to accommodate growing numbers of participating peers. In particular, new features were introduced to change the properties of the unstructured overlay topology formed by these peers. Despite its importance, little is known about the characteristics of these topologies and their dynamics in modern file-sharing applications.

This paper presents a detailed characterization of P2P overlay topologies and their dynamics, focusing on the modern Gnutella network. Using our parallel P2P crawler, we capture a complete snapshot of the Gnutella network with more than one million peers in just a few minutes. Leveraging more than 18,000 such recent overlay snapshots, we characterize the graph-related properties using individual overlay snapshots and overlay dynamics using hundreds of back-to-back snapshots. We show that snapshots captured by a slow crawler can lead to erroneous conclusions—such as a power-law degree distribution. Our results reveal that while the Gnutella network has dramatically grown and changed in many ways, it still exhibits the clustering and the short path lengths of a small world network. Furthermore, its overlay topology is highly resilient to random peer departure and even systematic attacks. More interestingly, overlay dynamics lead to an “onion-like” biased connectivity among peers where each peer is more likely connected to peers with higher uptime. Therefore, long-lived peers form a stable core that ensures reachability among peers despite overlay dynamics.

1 Introduction

The Internet has witnessed a rapid growth in the popularity of various Peer-to-Peer (P2P) applications during recent years. In particular, today’s P2P file-sharing applications (*e.g.*, FastTrack, eDonkey, Gnutella) are extremely popular with millions of clients and contribute a significant portion of the total Internet traffic [24, 12, 13]. These applications have changed in many ways to ac-

commodate growing number of participating peers. In these applications, participating peers often form an unstructured overlay which provides connectivity among the peers to search for desired files. More importantly, the overlay topology is continuously reshaped by both user-driven dynamics of peer participation as well as protocol-driven dynamics of neighbor selection. In a nutshell, as participating peers join and leave, they collectively, in a decentralized fashion, form an “unstructured” and dynamically changing overlay topology.

The design and simulation-based evaluation of new search and replication techniques have received much attention in recent years. These studies often make certain assumptions about topological characteristics of P2P networks (*e.g.*, power-law degree distribution) and usually ignore dynamics of overlay topology. However, little is known about the topological characteristics of popular P2P file sharing applications, in particular about overlay dynamics. Note that properties of unstructured overlay topology cannot be easily derived from implemented neighbor selection mechanism at individual peers mainly due to implementation heterogeneity, and dynamic peer participation. Without a solid understanding of topological characteristics in file-sharing applications, the actual performance of the proposed search and replication techniques in practice is unknown, and can not be meaningfully simulated.

Accurately characterizing the overlay topology of a large scale P2P network is challenging[30]. A common approach is to examine properties of snapshots of the overlay captured by a topology crawler. However, capturing accurate snapshots is inherently difficult for two reasons: (*i*) the dynamic nature of overlay topology, and (*ii*) a non-negligible fraction of discovered peers in each snapshots are not directly reachable by the crawler. Furthermore, the accuracy of captured snapshots is difficult to verify due to the lack of any accurate reference snapshot.

Previous studies that captured P2P overlay topologies with a crawler either deployed slow crawlers which inevitably lead to distorted (*i.e.*, stretched) snapshots of the

overlay [21], or partially crawled the overlay [22, 17] which is likely to capture biased (and non-representative) snapshots. These studies have not examined the accuracy of their captured snapshots and only conducted limited analysis on the overlay topology. More importantly, these few studies (except [17]) are considered outdated (more than three years old) since P2P filesharing applications have significantly increased in size, and incorporated several new topological features, in particular two-tier architecture and high degree connectivity. An interesting recent study [17] presented high level characterization of the two-tier Kazaa overlay topology. However, they did not present detailed graph-related properties of the overlay. Finally, to our knowledge, dynamics of unstructured P2P overlay topologies have not been studied in any detail.

This paper presents detailed characterization of both graph-related properties as well as dynamics of unstructured overlay topology based on recent large-scale and accurate measurements of the Gnutella network. We have recently developed a set of measurement techniques into a parallel P2P crawler, called *Cruiser*[28]. Cruiser can accurately capture a complete snapshot of Gnutella network with more than one million peers in just a few minutes. This speed is several orders of magnitude faster than any previously reported P2P crawler and thus its captured snapshots are significantly more accurate. Capturing snapshots rapidly also allows us to examine the dynamics of the overlay over much shorter timescales, which was not feasible in previous studies.

1.1 Contributions

Using the Cruiser, we have captured more than 18,000 snapshots of the Gnutella network during the past year. We use these snapshots to characterize Gnutella topology at two levels:

- *Graph-related Properties of Individual Snapshots:* We treat individual snapshots of the overlay as graphs and apply different forms of graph analysis to examine their properties.
- *Dynamics of the Overlay:* We present new methodologies to demonstrate the dynamics of the overlay and its evolution over different timescales.

We examine underlying causes of the observed properties and dynamics of the overlay topology, discuss their implications on design of search and replication techniques, and show how they can be leveraged to properly simulate overlay topologies. To the extent possible, we conduct our analysis in a generic (*i.e.*, Gnutella-independent) fashion to ensure the applicability to other P2P systems. Our main findings can be summarized as follows

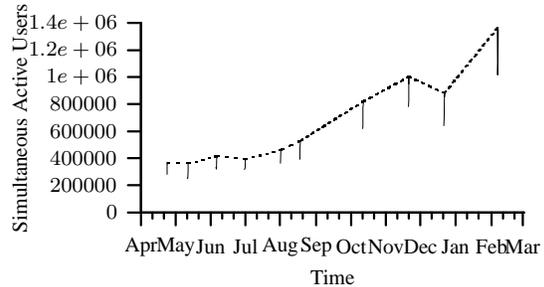


Figure 1: Change in network size over months. Vertical bars show variation within a single day.

- In contrast to earlier studies [6, 21, 19], we find that the overall node degree does not exhibit a power-law distribution. We show how power-law degree distributions could be a result of measurement artifacts.
- While the Gnutella network has dramatically grown and changed in many ways, it still exhibits the clustering and the short path lengths of a small world network. Furthermore, its overlay topology is highly resilient to random peer departure, and even systematic attacks.
- Long-lived ultrapeers form a stable and densely connected *core overlay* that serves as a backbone for routing messages throughout the top-level overlay. Therefore, this core overlay provides stable and efficient connectivity among participating peers despite the high degree of dynamics in peer participations.
- Also interestingly, the longer a peer remains in the overlay, the more it becomes clustered with other long-lived peers with same (or longer) uptime¹. In other words, connectivity within the core overlay exhibits an “onion-like” bias where most long-lived peers form a very connected core, and a group of peers with shorter uptime form a layer with a relatively biased connectivity to each other and to peers with higher uptime (*i.e.*, internal layers).

We shall discuss these results and their implications more later in the paper.

1.2 Why Examine Gnutella?

eDonkey, FastTrack, and Gnutella are the three most popular P2P file-sharing applications, according to Slyck.com [24], a website which tracks the number of users for different P2P applications. We elected to first

¹Throughout this paper, by “uptime” we mean the time that has elapsed since the peer has arrived.

focus on the Gnutella network based on a number of considerations. First, a variety of evidence indicates that the Gnutella network has a large and growing population of active users and generates considerable traffic volumes. Figure 1 depicts the average size of the Gnutella network over an eleven month period ending February 2005, indicating that network size has more than tripled (from 350K to 1.3 million peers) during our measurement period. We also observed time-of-day effects in the size of captured snapshots which is a good indication of active user participations in Gnutella network. While the total number of peers can change about 25% between day and night, a large number of peers (around 800,000) are always available. Also, examination of Internet2 measurement logs² reveal that the estimated Gnutella traffic measured on that network is considerable and growing. For example, for the 6 week period 10/11/04 – 11/21/04, the Gnutella traffic on Internet2 was estimated to be 79.69 Tbytes, up from 21.52 Tbytes for a 6 week period (02/02/04 – 03/14/04) earlier that year.

Second, Gnutella which was the first decentralized P2P system has evolved significantly since its inception in 2000. While it is among the most studied P2P networks in the literature, prior studies are at least 2–3 years old, and mostly considered the earlier flat-network incarnation. A detailed measurement study of the modern 2-tier Gnutella network is therefore timely and allows us to compare and contrast the behavior today from the earlier measurement studies, and to gain insights into the behavior and impact of the two-tier, unstructured overlay topologies which have been adopted by most modern P2P systems.

Third, our choice was also influenced by the fact that Gnutella is the most popular P2P file-sharing network with an open and well-documented protocol specification. This eliminates (or at least significantly reduces) any “incompatibility” error in our measurement that could potentially occur in other proprietary P2P applications that have been reverse-engineered, such as FastTrack/Kazaa and eDonkey.

The rest of this paper is organized as follows: Section 2 describes the fundamental challenges in capturing accurate snapshots, provides an outline of the modern Gnutella network. We present a brief overview of our crawler in Section 2.1. Section 4 presents a detailed characterization of graph-related properties of individual snapshots as well as the implications of our findings. In Section 5, we examine overlay dynamics, their underlying causes, and their implications on design and evaluation of P2P applications. Section 6 presents an overview of related work and Section 7 concludes the paper.

²<http://netflow.internet2.edu/weekly/>

2 Background

To accurately characterize P2P overlay topology, we need to capture *complete* and *accurate* snapshots. By “snapshot”, we refer to a graph that presents all participating peers (as nodes) and the connections (as edges) at a single instance in time. The most reliable and thus common approach to capture a snapshot is to crawl the overlay. Given information about a handful of initial peers, the crawler progressively contacts participating peers and collects information about their neighbors. In practice, capturing accurate snapshots is challenging for two reasons:

(i) The Dynamic Nature of Overlay: Crawlers are usually slow and require a relatively long time to capture a complete snapshot. Because of the dynamic nature of peer participation and neighbor selection, the longer a crawl takes, the more changes in participating peers and their connections occur, and the more *distorted* (or stretched) the captured snapshot becomes. More specifically, any connection that is established or closed during a crawl (*i.e.*, *changing connections*) is likely to be reported only by one end of the connection. We note that there is no reliable way to accurately resolve the status of changing peers or changing connections. In a nutshell, any captured snapshot by a crawler will be distorted, where the degree of such distortion is a function of the crawl duration relative to the rate of change in the overlay.

(ii) Unreachable Peers: A significant portion of discovered peers in each snapshot are not directly reachable since they have departed, reside behind a firewall, or overloaded []. Therefore, information about the edges of the overlay that are connected between these unreachable peers might be missing from the captured snapshots.

We argue that sampling a snapshot of the unstructured networks through partial crawls [22] or passive monitoring [23] is not a reliable technique for an initial characterization of the overlay topology for the following reasons: (i) in the absence of adequate knowledge about the properties and dynamics of the overlay topology, it is difficult to collect unbiased samples. For example, partial crawling of the network can easily result in a snapshot that is biased towards peers with higher degree; (ii) some graph-level characteristics of the overlay topology, such as the mean shortest path between peers (which we discuss in Subsection 4.2) cannot be accurately derived from partial snapshot. Because of these reasons, we attempt to capture as complete snapshots as we can and use them for our characterizations.

To describe our measurement methodology for addressing the above challenges, we provide a brief description of modern Gnutella as an example of a two-tier P2P file-sharing application.

2.1 Modern Gnutella

In the original Gnutella protocol, participating peers form a flat overlay and use TTL-scoped flooding of search queries to other peers. This approach has limited scalability. To improve the scalability of the Gnutella protocol, most modern Gnutella clients adopt a new overlay structure along with a new query distribution mechanism as follows:

(i) *Two-tier Overlay*: A new generation of popular file-sharing applications have adopted a *two-tier* overlay architecture to improve their scalability: a subset of peers, called *ultrapeers*, form a *top-level* overlay while other participating peers, called *leaf peers*, are connected to the top-level overlay through one or multiple ultrapeers (Figure 2). FastTrack (or Kazaa), Gnutella, and eDonkey all use some variation of this model. Those peers that do not implement the ultrapeer feature can only reside in the top-level overlay and do not accept any leaves. We refer to these peers as *legacy* peers. When a leaf connects to an ultrapeer, it uploads a set of hashes of its filename keywords to that ultrapeer. This allows the ultrapeer to only forward messages to the leaves who might have matching files. Leaf peers never forward messages. This approach reduces the number of forwarded messages towards leaf peers which in turn increases the scalability of the network by a constant factor.

(ii) *Dynamic Query*: The Gnutella developer community has adopted a new scheme for query distribution called *Dynamic Querying* [8]. The goal in this scheme is to only gather enough results to satisfy the user. Rather than forwarding a query to all neighbors, ultrapeers manage the queries for their leaves. Toward this end, an ultrapeer begins by forwarding a query down a few top-level connections with a low TTL. The receiving peer floods the query to its neighbors. The ultrapeer then waits for the results, and uses the ratio of the number of results to the estimated number of visited peers to determine how rare matches are. If matches are rare (*i.e.*, there are few or no responses), the query is sent down more connections with a relatively high TTL. If matches are more common but not sufficient, the query is sent down a few more connections with a low TTL. This process is repeated until the desired number of results (typically between 50 to 200 results) are collected or the ultrapeer gives up. Each ultrapeer estimates the number of searched ultrapeers through each neighbor based on the following equation: $Searched_Ultrapeers = \sum_{i=0}^{TTL-1} (d-1)^i$ where d denotes the connection degree of the neighbor. This equation assumes that all peers have the same connection degree.

At the time Dynamic Querying was introduced, the target number of neighbors for each ultrapeer was increased to allow more fine-grained control with Dynamic Querying since ultrapeers then have more neighbors to

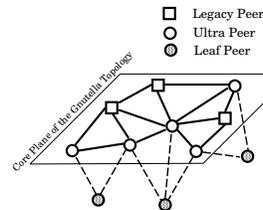


Figure 2: Two-tier Topology of Modern Gnutella

choose from.

3 Cruiser: A Parallel Crawler

To clarify the higher accuracy of our measurement compared to previous characterization studies of overlay topologies, we present an overview of our data collection and pre-processing steps in this section. We have developed a set of measurement techniques in a parallel Gnutella crawler, called *Cruiser* [28]. While the basic crawling strategy by Cruiser is similar to other crawlers, it improves the accuracy of captured snapshots by significantly increasing the crawling speed (*i.e.*, reducing crawl duration) primarily by using the following techniques: First, Cruiser employs a master-slave architecture in order to achieve a high degree of concurrency and to effectively utilize available resources on multiple desktop PCs³. A master process coordinates among multiple slave processes that act as virtually independent crawlers and crawl disjoint portions of the network in parallel. Each slave crawler opens hundreds of parallel connections, contributing an speed-up of nearly three orders of magnitude.

Second, Cruiser leverages the two-tier structure of the modern Gnutella network by only crawling the top-level peers (*i.e.*, ultrapeers and legacy peers). Since each leaf must be connected to an ultrapeer, this approach enables us to capture all the nodes and links of the overlay by contacting a relatively small fraction of all peers. Overall, this strategy leads to around 85% reduction in the duration of a crawl without any loss of information.

These techniques collectively result in a significant increase in crawling speed. *Cruiser can capture the Gnutella network with one million peers in around 7 minutes using six off-the-shelf 1 GHz GNU/Linux boxes in our lab. Cruiser’s crawling speed is about 140K peers/minute (by directly contacting 22K peers/minute), This is orders of magnitude faster than previously reported crawlers (i.e., 2 hours for 30K peers (250/minute) in [21], and 2 minutes for 5K peer (2.5K/minute) in [22]).* It is worth clarifying that while our crawling strategy is aggressive and our crawler requires considerable local resources, its behavior is not intrusive since each top-level peer is contacted only once per crawl.

³Using a master-slave architecture also allows us to deploy Cruiser in a distributed fashion if Cruiser’s access link becomes a bottleneck.

Post-Processing: Once information is collected from all reachable peers, we perform some post-processing to remove any obvious inconsistencies that might have been introduced due to changes in the topology during the crawling period. In summary, we might over-count edges by about 1% and ultrapeers by about 0.5%.

Unreachable Peers: We have carefully examined effect of unreachable peers on accuracy of captured snapshots. Previous studies assumed that these unreachable peers departed the network or are legacy peers that reside behind a firewall (*i.e.*, NATed), and simply excluded these large group of unreachable peers from their snapshot. It is important to determine what portion of unreachable peers are departed or NATed because each group introduces a different error on the snapshot. However, there is no reliable test to distinguish between departed and firewalled peers because firewalls can time out or refuse connections depending on their configuration.

We have conducted further investigations to learn more about this large group of unreachable peers in order to minimize (or at least accurately quantify) the resulting error in the captured snapshot [30]. *In summary, our investigation revealed that in each crawl, 30%-38% of discovered peers are unreachable. Among this group, 2%-3% are departed peers, and a majority of the 15%-24% of top-level peers that timeout are firewalled. The remaining unreachable peers are either firewalled or overwhelmed ultrapeers.*

Quantifying Snapshot Accuracy: We rigorously examined the effect of crawling speed and duration on two dimensions of snapshot accuracy: completeness and distortion. Our evaluations revealed that (i) Cruiser captures all ultrapeers and pair-wise connections among them, and the majority of connections to leaves; (ii) Both node distortion and edge distortion in captured snapshots linearly increases with crawling speed; and (iii) there is a fundamental tradeoff between completeness and distortion of captured snapshots. There exist a sweet-spot for crawl duration beyond which crawling has diminishing return.

Data Set: We have captured more than 18,000 snapshots of the Gnutella network during the past eleven months (Apr. 2004–Feb. 2005) with Cruiser. In particular, we collected back-to-back snapshots for several one-week intervals as well as randomly distributed snapshots during various times of the day to ensure that captured snapshots are representative.

4 Overlay Graph Properties

The two-tier overlay topology in modern Gnutella (as well as other unstructured P2P networks) consists of ultrapeers that form a “spaghetti-like” top-level overlay and a large group of leaf peers that are connected to the top-level at multiple points. We treat individual snap-

Implementation Percentage	LimeWire	BearShare	Other
	74%–77%	19%–20%	4%–6%

Table 2: Distribution of Implementation

shots of the overlay as graphs and apply different forms of graph analysis to examine their properties. We pay special attention to the top-level overlay since it is the core component of the topology. Throughout our analysis, we compare our findings with similar results reported in previous studies. However, it is important to note that we are unable to determine whether the reported differences (or similarities) are due to changes in the Gnutella network or the accuracy of captured snapshots in previous studies.

Table 1 presents summary information of four sample snapshots after post-processing. The results in this section are primarily from the snapshots in Table 1. However, we have examined many other snapshots and observed similar trends and behaviors. Therefore, we believe the presented results are representative. Presenting different angles of the same subset of snapshots allows us to conduct cross comparison and relate various findings.

In this section, we explore the node degree distribution in Subsection 4.1, the reachability and pairwise distance properties of the overlay in Subsection 4.2, small world characteristics in Subsection 4.3, and the resilience of the overlay in Subsection 4.4.

Implementation Heterogeneity: The open nature of the Gnutella protocol has led to several known (and possibly many unknown) implementations. It is important to determine the distribution of different implementations (and configurations) among participating peers since it could directly affect the overall properties of the overlay topology. This will help us explain some of the observed properties of the overlay. Table 2 presents the distribution of different implementations across discovered ultrapeers in four snapshots in Table 1. This table shows that a clear majority of contacted ultrapeers use the LimeWire implementation. We also discovered that a majority of LimeWire peers (around 94%) use the most recent version of the software available at the time of the crawl (3.8.x). These results reveal that while heterogeneity exists, a clear majority of Gnutella users run a recent version of LimeWire.

We are particularly interested in the number of connections that are used by each implementation since it directly affects the degree distribution of the overall topology⁴. Our measurements reveal that LimeWire and BearShare ultrapeers implementation prefer to serve 30 and 45 leaves, respectively, whereas both try to maintain

⁴This information can be obtained from available LimeWire source code. However, not all implementations are open, and users can always change the source code of open implementations. Thus, we need to collect this information from running ultrapeers in action.

Crawl Time	Total Nodes	Leaves	Top-level	Unreachable	Top-Level Edges
09/27/04	725,120	614,912	110,208	35,796	1,212,772
10/11/04	779,535	662,568	116,967	41,192	1,244,219
10/18/04	806,948	686,719	120,229	36,035	1,331,745
02/02/05	1,031,471	873,130	158,345	39,283	1,964,121

Table 1: Sample Crawl Statistics

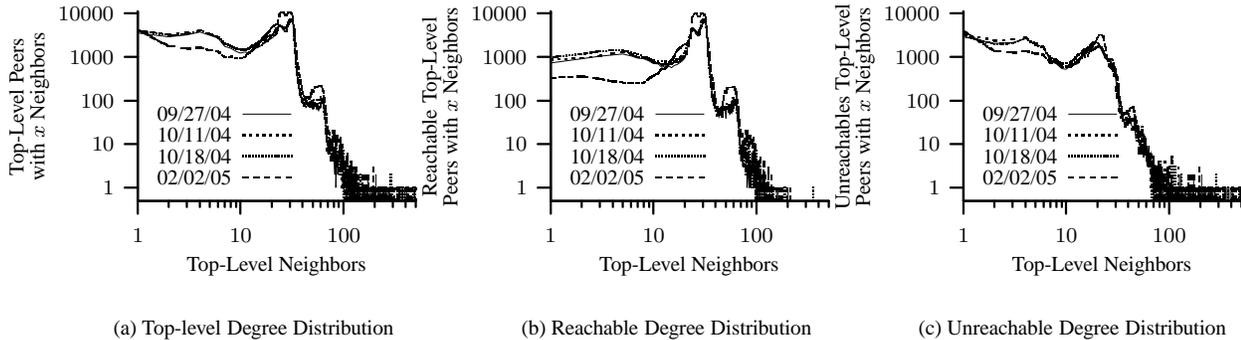


Figure 3: Different angles of the top-level degree distribution in Gnutella topology

around 30 neighbors in the top-level overlay.

4.1 Node Degree Distributions

The introduction of the two-tier architecture in the overlay topology along with the distinction between ultra-peers, leaves, and legacy peers in the modern Gnutella protocol demands a closer examination of the different degree distribution among different group of peers.

Node Degree in the Top-Level Overlay: Previous studies reported that the distribution of node degree in the Gnutella network exhibited a power-law distribution [21, 1, 6] and later changed to a two-segment power-law distribution [19, 21]. To verify this property for the modern Gnutella network, Figure 3(a) depicts the distribution of node degree among all peers (both unreachable and reachable) in the top-level overlay for the four sample snapshots presented in Table 1. This distribution has a spike around 30 and does not follow a power-law⁵. A key question is *to what extent this difference in degree distribution is due to the change in the overlay structure versus error in captured snapshots by earlier studies*. To examine this question, we captured a distorted snapshot by a slow crawler⁶ which is similar to the 50-connection crawler used in an earlier study [21]. Figure 4(a) depicts the degree distribution based on this distorted snapshot, which is significantly more similar to a two-piece power-law distribution⁷. If we further slow down the

⁵The degree distribution for all the presented results is limited to 500 which includes all but a handful of peers with larger degree that are discussed later.

⁶To reduce the crawling speed, we simply limited the degree of concurrency (*i.e.*, number of parallel crawler) to 60 in Cruiser.

⁷To properly compare these snapshots with different sizes, Y-axis

crawling speed, the resulting snapshots contains a higher degree of edge distortion, and the derived degree distribution looks more similar to a single-piece power-law distribution, the result reported by earlier studies [1, 6]. *To a slow crawler, peers with long uptimes appear as high degree because many short-lived peers report them as neighbors. However, this is a mischaracterization since these short-lived peers are not all present at the same time. More importantly, this finding demonstrates that using distorted snapshots that are captured by slow crawlers could easily lead to incorrect characterization of P2P overlays.*

We further examined the derived degree distribution from accurate snapshots captured by Cruiser as follows. Because we were unable to contact every top-level peer, this distribution is biased slightly low since it does not include all edges. To address this problem, we split the data into Figures 3(b) and 3(c) that depict the neighbor degree distribution for reachable and unreachable peers, respectively. The data in Figure 3(b) is unbiased since we contacted each peer successfully, *i.e.*, we discovered every edge connected to these peers. The spike around a degree of 30 is more pronounced in this figure. Figure 3(c) presents the observed degree distribution for unreachable top-level peers (*i.e.*, overloaded or NATed). This distribution is more strongly biased low since we cannot observe the connections between pairs of these peers. In this data, a much greater fraction of peers have an observed degree below 30. Many of these peers probably have a true degree closer to 30, with the true distribution likely similar to that in Figure 3(b).

The degree distribution among contacted top-level in Figure 4(a) was normalized by number of peers in the snapshot

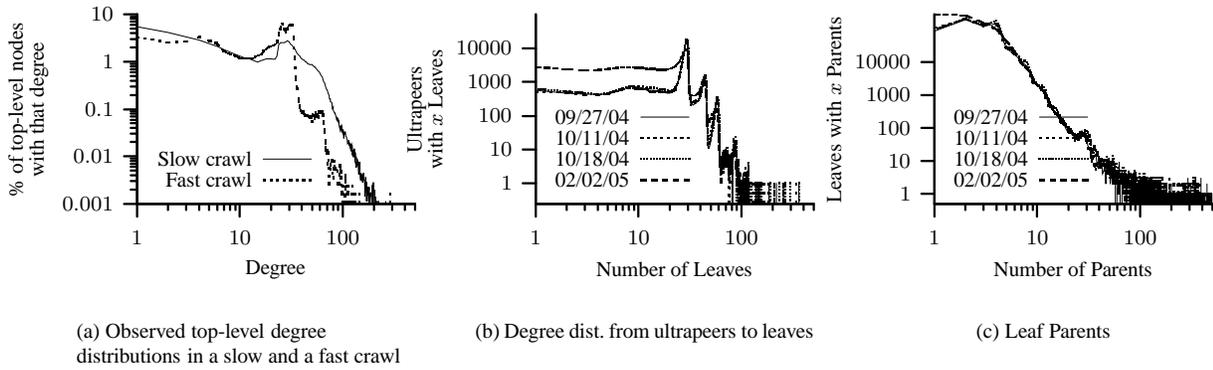


Figure 4: Different angles on degree distribution in Gnutella

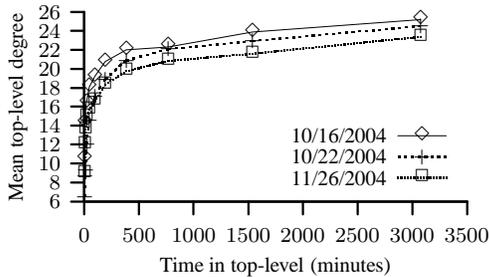


Figure 5: Mean degree as a function of uptime. Standard deviation is large (7–13).

peers has two distinct segments around a spike in degree of 30, resulting from LimeWire and BearShare’s behavior of attempting to maintain 30 neighbors. The peers with higher degree represent other implementations that try to maintain a higher node degree, or the rare user who has modified their client software. The peers with lower degree are peers which have not yet established 30 connections. In other words, the observed degree for these peers is temporary. They are in a state of flux, working on opening more connections to increase their degree. To verify this hypothesis, we plot the mean degree of peers as a function of their uptime in Figure 5. The standard deviation for these measurements is quite large (around 7 – 13), indicating high variability. When peers first arrive, they quickly establish several connections. However, since node churn is high, they are constantly losing connections and having to make new ones. As time passes, long-lived peers gradually accumulate stable connections to other long-lived peers. We further explore this issue in Section 5 when we examine overlay dynamics.

Node Degree For Leaves: To characterize properties of the semi-structure topology, we have examined the degree distribution between the top-level overlay and leaves, and vice versa. Figure 4(b) presents the degree distribution of connections from ultrapeers to leaf peers. Distinct spikes at 30, 45 and 75 degree are visible. The

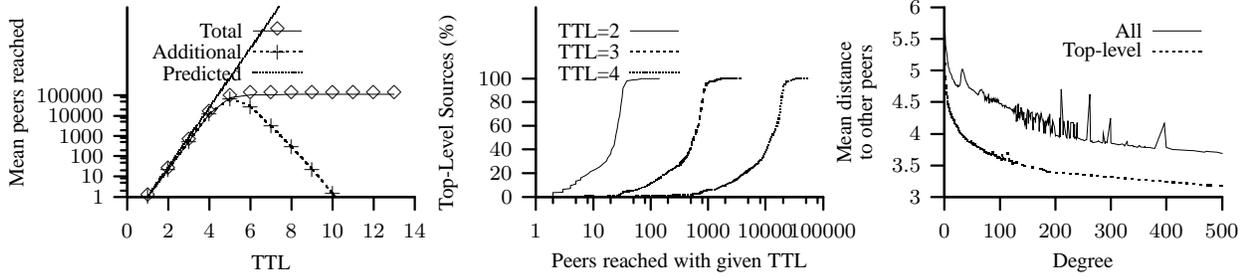
first two spikes are due to the corresponding parameters used in LimeWire and BearShare implementations, respectively. The third spike is due to a less common implementation. This figure shows that a significant minority of ultrapeers are connected to less than 30 leaf peers, which indicates availability of the system to accommodate more leaf peers.

In Figure 4(c), we present the degree of connectivity for leaf peers. This result reveals that most of leaf peers connect to three ultrapeers or less (the behavior of LimeWire), a small fraction of leaves connect to several ultrapeers, and a few leaves (< 0.02%) connect to an extremely large number of ultrapeers (100–3000).

Implications of High Degree Peers: We observed a few outlier peers with an unusually high degree of connectivity in all degree distributions in this subsection. The main incentive for these peers is for them to reduce their mean distance to other peers. To quantify the benefit of this approach, Figure 6(c) presents the mean distance to other peers as a function of degree, averaged across peers with the same degree. We show this for both the top-level overlay and across all peers. This figure shows that the mean path to participating peers exponentially decreases with degree. In other words, there are steeply diminishing returns from increasing degree as a way of decreasing distance to other peers.

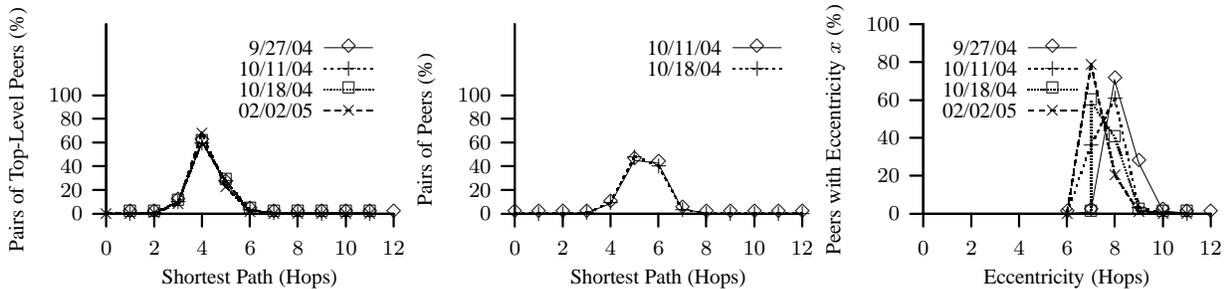
Turning our attention to the effects of high-degree peers on the overlay, for scoped flood-based querying the traffic generated by these nodes is proportional to their degree. Note that high-degree ultrapeers may not be able to, or may not choose to, route all of the traffic between their neighbors. Thus, they may not actually provide as much connectivity as they appear to, affecting the performance of the overlay.

During our analysis, we discovered around 20 ultrapeers (all on the same /24 subnet) with an extremely high degree (between 2500 to 3500) in our snapshots. In fact, these were the only nodes with a degree greater than 1000. These high-degree peers are widely “visible” throughout the overlay, and thus receive a significant por-



(a) Mean Top-Level Peers Searched by TTL from the 9/27/2004 snapshot (b) Cumulative Top-Level Peers Searched CDF (c) Correlation between ultrapeer's degree and its mean distance from other ultrapeers from the 10/18/04 snapshot

Figure 6: reachability, diameter, and shortest path in Gnutella topology



(a) Ultrapeer-to-ultrapeer shortest paths (b) Distribution of path lengths across all pairs of peers (c) Distribution of Eccentricity in the Top-level Overlay

Figure 7: Different angles on path lengths

tion of exchanged queries among other peers. We directly connected to these high degree peers and found they do not actively participate in the overlay (*i.e.*, do not forward any traffic)⁸. We removed these inactive high degree peers from our snapshots when considering path lengths since their presence would artificially improve the connectivity of the overlay⁹.

4.2 Reachability

Node degree distribution of overlay topologies does not reveal reachability among nodes in the overlay graph. Therefore, we examine two equally important properties of overlay topologies that represent the reachability of queries throughout the overlay as follows: (i) reachability of flood-based query, and (ii) the pairwise distance between arbitrary pairs (*i.e.*, or diameter of the overlay). **Reachability of Flood-Based Query:** Figure 6(a) depicts the *mean* number of newly visited peers and its cu-

mulative value as a function of TTL, averaged across top-level peers in a single snapshot. The shape of this figure is similar to the result that was reported by Lv, et al. (Figure 3 in [19]) which was captured in October 2000, with a significantly smaller number of peers (less than 5000). Both results indicate that the number of newly visited peers exponentially grows with increasing TTL up to a certain threshold and has diminishing returns afterwards. This illustrates that the dramatic growth of network size has been effectively balanced by the introduction of ultrapeers and an increase in node degree. Thus, while the network has changed in many ways, the percentage (but not absolute number) of newly reached peers per TTL has remained relatively stable. Figure 6(a) also shows the number of newly visited peers predicted by the Dynamic Querying formula (assuming a node degree of 30), which we presented in Section 2.1. This result indicates that the formula closely predicts the number of newly visited peers for TTL values less than 5.

Figure 6(b) shows a different angle of reachability for the same snapshot by presenting the CDF of the number of visited peers from top-level peers for different TTL values. This figure clearly shows the distribution of reachability for flood-based queries among participating peers. We use a logarithmic x -scale to magnify the left part of the figure for lower TTL values. This figure shows

⁸To our surprise, it appears that these peers monitor exchanged messages among other participating peers. They could be trying to locate copyright infringement among Gnutella users or collecting ratings information to measure which songs consumers might like to buy.

⁹These high degree peers have a more important effect on a power-law graph than current Gnutella topology. However, none of the previous studies have verified active participation of these nodes in the overlay.

two interesting points: First, the total number of visited peers with a TTL value of n is almost always an order of magnitude higher than with a TTL value of $(n - 1)$. In other words, TTL is the primary determinant of the mean number of newly visited peers independent of a peer’s location. Second, the distribution of newly visited peers for each TTL is not uniform among all peers. As TTL increases, this distribution becomes more skewed (considering the logarithmic scale for x axis). This is a direct effect of node degree. More specifically, if a peer or one of its neighbors have a very high degree, its flood-based query reaches a proportionally larger number of peers.

Pair-wise Distance: Figure 7(a) shows the distribution of shortest-path lengths in terms of overlay hops among all pairs of top-level peers from three snapshots. Ripseau et al. [21] presented a similar distribution for the shortest-path length based on snapshots that were collected between November 2000 and June 2001 with 30,000 peers. Comparison between these results reveal two differences: (i) the pairwise path between peers over the Gnutella topology have become *significantly more homogeneous in length*, with shorter mean value over the past two years. More specifically, the old snapshot shows 40% and 50% of all paths having a length of 4 and 5 hops whereas our result shows a surprising 60% of all paths having a length of 4. (ii) the results from our snapshots are nearly identical; whereas in [21], there is considerable variance from one crawl to another. In summary, *the path lengths have become both more homogeneous and more stable*.

Effect of Two-Tier Topology: To examine the effect of the two-tier overlay topology on path length, we also plot the path length between all peers (including leaves) in 7(b). If each leaf had only one ultrapeer, the distribution of path length between leaves would look just like the top-level path lengths (Figure 7(a)), but right-shifted by two. However, since each leaf peer has multiple parents, the path length distribution between leaves (and thus for all peers) has a subtle relationship with Figure 7(a). Comparing Figures 7(a) and 7(b) shows us the cost introduced by using a two-tier overlay. In the top-level, most paths are of length 4. Among leaves, we see that around 50% of paths are of length 5 and the other 50% are of length 6. Thus, getting to and from the top-level overlay introduces an increase of 1 to 2 overlay hops.

Eccentricity: The longest observed path in these three snapshots was 12 hops, however the vast majority (99.5%) of paths have a length of 5 hops or less. To further explore the longest paths in the topology, we examined the distribution of eccentricity in the top-level overlay. Given the function $P(i, j)$ that returns the shortest path distance between nodes i and j , the eccentricity, E_i of node i is defined as follows: $E_i = \max(P(i, j) \forall j)$.

Graph	L_{actual}	L_{random}	C_{actual}	C_{random}
New Gnutella.	4.17–4.23	3.75	0.018	0.00038
Old Gnutella	3.30–4.42	3.66	0.02	0.002
Movie Actors	3.65	2.99	0.79	0.00027
Power Grid	18.7	12.4	0.08	0.005
C. Elegans	2.65	2.25	0.28	0.05

Table 3: Small World Characteristics

Figure 7(c) shows the distribution of eccentricity in four topology snapshots. This figure shows that the distribution of eccentricity is homogeneous and low which is an indication that the overlay graph is a relatively balanced and well-connected mesh, rather than a chain of multiple groups of peers.

4.3 Small World

Recent studies have shown that many biological and man-made graphs (*e.g.*, collaborations among actors, the electrical grid, and the WWW graph) exhibit “small world” properties. In these graphs, the mean pairwise distance between nodes is small and nodes are highly clustered compared to random graphs with the same number of vertices and edges. A study by Jovanovic et al. [11] in November–December 2000 concluded that the Gnutella network exhibits small world properties as well. Our goal is to verify to what extent recent topologies of the Gnutella network still exhibit small world properties despite growth in network size, increase in node degree and changes in network structure. The clustering coefficient of a graph, C_{actual} , represents how frequently each node’s neighbors are also neighbors, and is defined as follows [32]: $C(i) = \frac{D(i)}{D_{max}(i)}$, $C_{actual} = \frac{\sum_i C(i)}{|V|}$. $D(i)$, $D_{max}(i)$ and $|V|$ denote the number of edges between neighbors of node i , the maximum possible edges between neighbors of node i , and the number of vertices in the graph, respectively. For example, if node A has 3 neighbors, they could have at most 3 edges between them, so $D_{max}(A) = 3$. If only two of them are connected together, that’s one edge and we have $D(A) = 1$ and $C(A) = \frac{1}{3}$. $C(i)$ is not defined for nodes with fewer than 2 neighbors. Thus, we simply exclude these nodes from the computation of C_{actual} . Table 3 presents ranges for the clustering coefficient and mean path length for Gnutella snapshots in Table 1 as well as the mean values from three random graphs with the same number of vertices and edges (*i.e.*, C_{random} and L_{random}). Because computing the true mean path lengths (L_{random}) is computationally expensive for large graphs, we used the mean of 500 sample paths selected uniformly at random. We also include the same information presented by Jovanovic et al. [11] and three classic small world graphs [32].

A graph is loosely identified as a small world when its mean path length is close to random graphs with the same

number of edge and vertices, but its clustering coefficient is orders of magnitude larger than the corresponding random graph (*i.e.*, L_{actual} and L_{random} are close, but C_{actual} is orders of magnitude larger than C_{random}). All three classic small world graphs in the table exhibit variants of these conditions. Snapshots of modern Gnutella clearly satisfy these conditions which means that modern Gnutella still exhibits small world properties. Comparing the clustering coefficient between modern Gnutella and old Gnutella shows that modern Gnutella has less clustering. A plausible explanation is the increased size which provides the opportunity for more diverse connectivity to other peers. A high clustering coefficient implies a larger portion of redundant messages in flood-based querying. The observed clustering could be a result of factors like peer bootstrapping, peer discovery mechanism and overlay dynamics, and further analysis is needed to better understand the underlying causes.

4.4 Resilience

We also examine the resilience in different snapshots of the Gnutella overlay topology using two different types of node removal: random and pathologically removing the highest-degree nodes first. An early study [22] conducted the same analysis on Gnutella based on a partial topology snapshot, found that the overlay is resilient to random departures, but under pathological node removal it quickly becomes very fragmented (after removing just 4% of nodes).

Figure 8 depicts the fraction of remaining nodes in the topology which remain still connected in both the random and pathological node removal. *This figure clearly shows that not only the Gnutella overlay is extremely robust to random peer removals, but it also exhibits high resilience to pathological node removal.* Even after removing 85% of peers randomly, 90% of the remaining nodes are still connected. For the pathological case, after removing the 50% of peers with the highest-degree, 75% of the remaining nodes remain connected. There are two possible factors contributing to this difference with earlier results [22] (*i*) the higher node degree of most nodes in modern Gnutella, and (*ii*) a non-negligible number of missing nodes and edges in their partial snapshot. Our result implies that complex overlay construction algorithms like (*e.g.*, [33]) are not always a necessary prerequisite for ensuring resilience in unstructured overlay.

5 Overlay Dynamics

In Section 4, we only characterized the graph-related properties of individual snapshots of the overlay topology. However, in practice the overlay topology is inherently dynamic since connections (*i.e.*, edges) of the over-

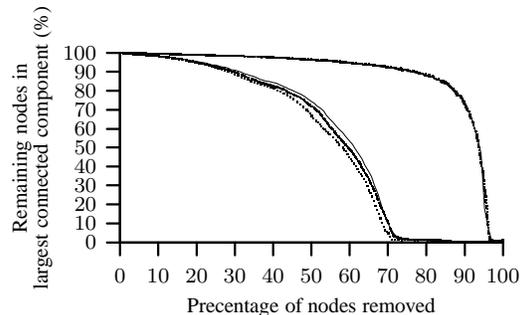


Figure 8: Fraction of remaining nodes in the largest connected component as a function of the percentage of original nodes removed for the 9/27, 10/11, and 10/18 snapshots. The top (overlapped lines) and the bottom three lines present random and pathological node removal scenarios, respectively.

lay are constantly changing with time. These dynamics can significantly affect the main functionality of the overlay which is to provide connectivity and efficiently route the messages (*e.g.*, queries, responses) among participating peers. Characterizing overlay dynamics enable us to examine their impact on performance of P2P applications. For example, a query or response message can be routed differently or even dropped as a result of changes in the edges of the overlay. To our knowledge, aggregate dynamics of unstructured P2P overlay have not been studied. There are two basic causes for observed dynamics in the overlay topology as follows:

Dynamics of Peer Participation: When a peer joins (or leaves) the network, it establishes (or tears down) its connections to other participating peers in the overlay. Therefore, these changes in overlay edges are *user-driven*¹⁰ *Dynamics of Neighbor Selection:* Two existing peers in the overlay may establish a new (or tear down an existing) connection between them. Such a change in edges is not triggered by users and thus considered *protocol-driven*.

Note that the user-driven dynamics of peer participation are likely to exhibit similar heavy-tail distribution in different P2P applications[29, 27]. Therefore, characterization of user-driven dynamics in the overlay provides a useful insight for design of other Gnutella-like unstructured P2P overlays.

In this section, we characterize the dynamics of the Gnutella network. In our analysis, we primarily focus on the top-level overlay since leaf nodes do not play any role in forwarding traffic. Our main goal is to determine whether observed dynamics (*i.e.*, the rate of change in

¹⁰Note that Gnutella does not run as a daemon. Therefore, peer arrival/departure is a reliable indication of user action. We are mindful that dynamic IP addresses could force some peers to leave and rejoin the network with a new address. However, this does not affect our analysis since we examine the effect of each departure/arrival event on the overlay dynamics.

the edges of the overlay) is different at various regions of the overlay. More specifically, we want to investigate (i) *whether a subset of participating peers form a relatively stable core for the overlay*, (ii) *what properties (such as size, diameter, degree of connectivity or clustering) this stable core exhibits*, and (iii) *what the underlying factors in the formation and observed properties of such a stable core are*.

Methodology: One key issue is to define a core region for the “spaghetti-like” overlay. We use the following methodology to identify and characterize any potentially stable core for the overlay. Intuitively, if the overlay has a stable core, it must contain the long-lived peers of the overlay. Therefore, to identify the stable core of the overlay at any point of time, we select the subset of participating peers who have been part of the overlay for at least τ minutes *i.e.*, all peers whose uptime is longer than a threshold τ . We call this subset of peers the *stable peers*, or $SP(\tau)$, and only focus on this subset in our analysis. However, by changing τ , we can control the minimum uptime of selected peers and thus the relative stability and size of $SP(\tau)$.

To conduct this analysis, we use several slices of our dataset where each slice contains hundreds of back-to-back topology snapshots that we have captured over contiguous 48 hours periods. Let’s consider the last captured snapshot over each 48 hour period as a reference snapshot. Any peer in the reference snapshot must have joined the overlay either before or during our measurement period. By looking back through the snapshots, we can determine (with the accuracy of few minutes) the arrival time of all peers that joined during the measurement period. For those peers that were present for the entire measurement period, we can only conclude that their uptime is at least 48 hours. Having this information we can annotate all peers in the reference snapshot with their uptime information. Figure 9(a) depicts the CCDF of uptime among existing peers in the reference snapshot of several slices of data (Figure 9(b) presents the initial part of the same graph). In essence, this figure presents the distribution of uptime among participating peers in steady state, implying that the size of $SP(\tau)$ exponentially decreases with τ . This is more visible over longer time scales. Furthermore, this also implies that the number of connections within $SC(\tau)$ dramatically decreases with τ .

Internal Connectivity Within Stable Core: To study different angles of connectivity among ultrapeers within $SP(\tau)$, we focus only on the connections of the overlay where both end points are inside $SP(\tau)$, *i.e.*, we remove all edges to peers outside $SP(\tau)$. We call this the stable core overlay or $SC(\tau)$. The first question is: *whether the $SC(\tau)$ is fully connected?* Figure 10(a) depicts the portion of ultrapeers within $SC(\tau)$ that are connected, as a

function of τ . This figure clearly demonstrates that while the portion of connected peers slightly decreases with τ over long times scales, a significant majority (86%–94%) of peers within $SP(\tau)$ remain fully connected. The minor drop in the percentage of connected peers is due to exponential decrease in number of peers within $SC(\tau)$ which in turn reduces the number of edges among peers and thus affects the opportunity for pairwise connectivity. The second question is: *how clustered and dense is the connected portion of the core overlay?* Figure 10(b) shows the diameter and characteristic (mean) path length among fully connected peers in the stable core overlay. Interestingly, both the mean path length and the diameter of the stable core overlay remain relatively stable as τ increases, despite the dramatic drop in number of edges. Furthermore, the mean path length for the stable core overlay, even when it has a very small population (only 10% of top-level peers for $\tau=45h$), is around 5 hops, very close to the mean path length for the entire top-level overlay (4.17–4.23 from the first row of Table 3). Finally, Figure 10(c) depicts the evolution of the clustering coefficient for the stable core overlay as τ increases, along with the clustering coefficient for the entire top-level overlay in the reference snapshot. This figure shows two important points: (i) peers within the stable core overlay are more clustered together than the entire top-level overlay on average, and, more importantly, (ii) connectivity among peers within the stable core overlay becomes increasingly more clustered with τ . This latter point implies that *the longer a peer remains in the overlay, the more likely it establishes connections to peers with equal or higher uptimes*, *i.e.*, *the more biased its connectivity becomes toward peers with higher uptime*. Since connections for all participating peer exhibit the same behavior, connectivity of the overlay exhibits a biased “onion-like” layering where peers with similar uptime (form a layer) have a tendency to connect to peers with the same or higher uptime (internal layers of the onion). However, this tendency becomes stronger as uptime increases. Since the size of $SN(\tau)$ decreases with τ , this means that internal layers are smaller and more clustered.

External Connectivity to/from Stable Core: To quantify the connectivity between $SC(\tau)$ and the rest of the overlay we examined whether peers within $SC(\tau)$ have a higher tendency to connect to each other rather than peers outside the core. To quantify any potential tendency, we calculate the ratio of internal edges to the total number of edges and compare that with the same ratio for a randomly generated graph with the same number of nodes, same degree distribution among nodes, and same number of edges. For a fair comparison, we present the notion of a *half edge* for a graph as follows: we cut the edge E_{ij} between two nodes i and j , and define $HalfEdge(i, j)$

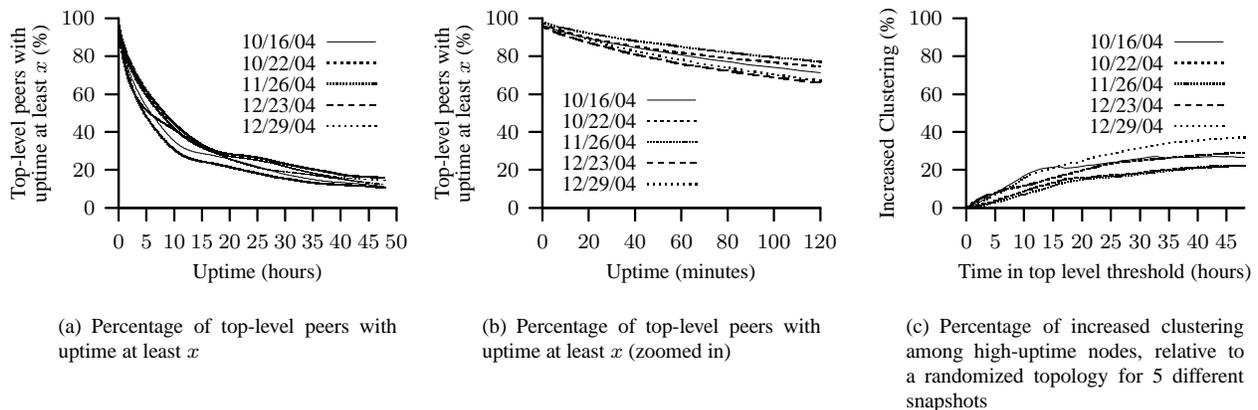


Figure 9: Size of the stable core and its external connectivity for different τ

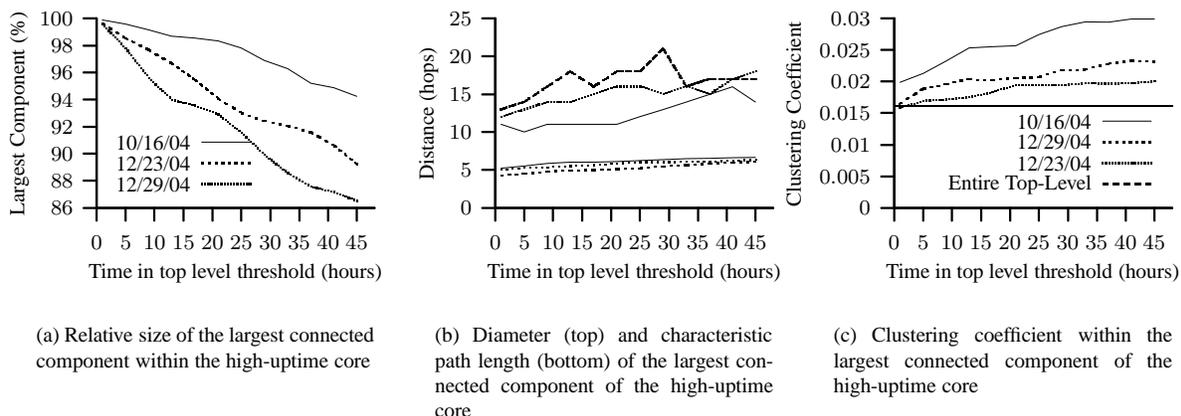


Figure 10: Different angles of connectivity with the stable core

as the half of E_{ij} that is connected to node i . Then, the ratio of internal to total half-edges can be easily calculated as follows:

$$R = \frac{\sum_{i \in SC} \sum_{j \in SC} \text{HalfEdge}(i,j)}{\sum_{i \in SC} \sum_{\text{all } j} \text{HalfEdge}(i,j)}$$

Figure 9(c) depicts $(R_g - R_r)/R_r$ as a function of τ where R_g and R_r denote the value of R for several snapshots and their corresponding randomly generated graphs, respectively. This figure demonstrates that the longer a peer remains in the network, its connectivity becomes more biased towards peers with the same or higher uptime. This is another evidence that peers exhibit an onion-like biased connectivity and the degree of such biased increases with uptime.

Implications of Stable and Layered Core Overlay: The connectivity of the core overlay implies that all peers within the core do not depend on peers outside the core for reachability. In other words, the core overlay provides a stable and efficient backbone for the entire top-level overlay that ensures connectivity among all participating peers despite the high rate of dynamics among

peers outside the core.

5.1 Examining Underlying Causes

A key question is *how this interesting onion-like layered connectivity is formed in the overlay in an unintentional and uncoordinated fashion?*. To address this issue, we quantify the contribution of user-driven and protocol-driven dynamics in changes of the edges of the overlay. We can distinguish protocol-driven vs user-driven changes in edges between two snapshots of the overlay as follows: if at least one of the endpoints for a changing edge has arrived (or departed) the network between the snapshots, that change is user-driven. Otherwise, a changing edge is considered as protocol-driven. To answer the above question, we examine a 48-hour slice of back-to-back snapshots from 10/14/2004 to 10/16/2004, using the first snapshot as a reference. Given a slice, we can detect new or missing edges in any snapshot compared to the reference snapshot, for peers in both snapshots. Let δ_{p-} and δ_{u-} (δ_{p+} and δ_{u+}) denote the normal-

ized ratio of missing (and new) edges in a snapshot due to protocol-driven (p) and user-driven (u) causes, normalized by the number of edges in the reference snapshot. Figure 11(a) and 11(b) depict $\delta_- = \delta_{p-} + \delta_{u-}$ and $\delta_+ = \delta_{p+} + \delta_{u+}$ for back-to-back snapshots for the slice under investigation. Each figure also depicts the breakdown of changes in edges into two groups: protocol-driven and user-driven changes¹¹. The left graph (δ_-) shows that around 10% and 30% of edges in the overlay are removed due to protocol-driven and user-driven factors during the first 100 minutes, respectively. After this period, almost all removed edges are due to departing peers. Similarly, from the right graph, many edges are added during the first 100 minutes due to both protocol-driven factors and the arrival of new peers. After this period, almost all new edges involve a newly arriving peer. These results shows two important points: First, each peer may establish and tear down many connections to other peers during the initial 100 minutes of its uptime. But peers with higher uptime (*i.e.*, peers inside $SC(\tau)$ for $\tau \geq 100$ min), maintain their connections to their remaining long-lived neighbors, and only add (or drop) connections to arriving (or departing) peers. This behavior appears to explain the formation of the biased onion-like layering in connectivity within the overlay. Second, user-driven dynamics are the dominant factor in long-term changes of the overlay. Since dynamics of peer participations exhibit similar dynamics in different P2P systems, other Gnutella-like overlays are likely to show similar behavior. We plan to conduct further investigation to better understand the underlying dynamics that contribute to this behavior.

6 Related Work

As we listed throughout the paper, there is a handful of prior studies on characterizing peer-to-peer overlay

¹¹Note that δ_p and δ_u are by definition cumulative.

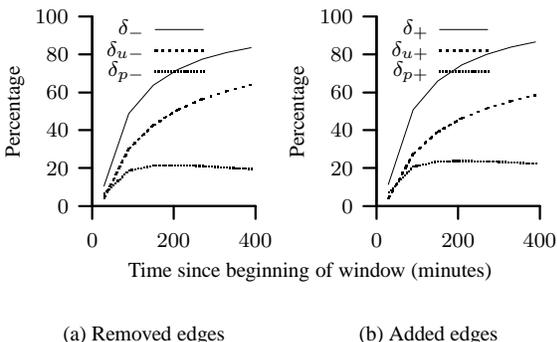


Figure 11: Contribution of user- and protocol-driven dynamics in variations of edges in the overlay

topologies in file-sharing applications [21, 1, 19, 21, 11]. These studies are more than three year old, did not verify the accuracy of their captured snapshots, conducted only limited analysis. A recent study [17] have used both passive measurement and active probing of 900 super nodes to study behavior of the Kaaza overlay. They have mostly focused on the number of observed connections (within the top-level overlay and from top-level overlay and leaf nodes) and their evolution with time. However they have not examined detailed graph-related properties of the overlay, or collective dynamics of the "entire overlay topology", both of which are the focus of the current work.

There has been a wealth of measurement research on other properties of peer-to-peer systems. These studies cover several topics: (i) file characteristics [5, 16, 2, 18], (ii) transfer characteristics [9, 16], (iii) peer characteristics [23, 22], (iv) query characteristics [25, 2, 15, 3], and (v) comparisons of different implementations [14, 10]. Since they explore different aspects of peer-to-peer networks, these studies complement our work. There has also been several modeling and simulation-based studies on improvement of search in Gnutella-like P2P networks [4, 35, 34, 26]. Our characterization can be directly used by these studies as a reference for comparison of suggested topology models, and our crawled snapshot can be used as realistic topologies for conducting simulation-based study on proposed search mechanisms on a network much larger than those captured by previous work.

Finally, the research studies on characterization of the Internet topology (*e.g.*, [7]) and network topology generators (*e.g.*, [31]) are closely related to our work. However, these studies focus on the Internet topology rather than an overlay topology. We plan to conduct further characterization of Gnutella topology by applying some of the suggested graph analysis in these studies to the Gnutella overlay topology.

7 Conclusions

In this paper, using Gnutella, we presented the first detailed characterization of an unstructured two-tier overlay topology that is typical of modern popular P2P systems, based on measurements involving fast large-scale crawls, gathered over a 11-month period. We described fundamental challenges in capturing accurate snapshots, and demonstrated that inaccurate snapshots can lead to erroneous conclusions—such as a power-law degree distribution. We characterized the graph-related properties of individual snapshots, the dynamics across time of the overlay topology, and investigated their underlying causes and implications. Our main findings are summarized in Section 1.1.

This study developed essential insights into the behav-

ior of overlay topologies which are necessary to improve the design and evaluation of file-sharing applications. The existence of a stable well-connected core of long-lived peers suggests that there may be benefits in terms of increasing search resilience in the face of the overlay dynamics, by biasing/directing the search towards longer lived peers and therefore towards this core. It may also be useful to cache indexes or content at long lived peers in order to reduce load on the stable core (or backbone of long-lived peers) specially if biased forwarding of queries are adopted. For example the idea of one-hop replication [20] that is intended for power-law topologies can be changed to probabilistic one-hop replication biased towards peers with longer uptime.

We are continuing this work along number of directions. We are actively monitoring the Gnutella network and plan to further examine the dynamics of peer participation over short time scales, explore any longer term trends in the topology, and observe variations in several key properties (*e.g.*, small-world coefficient, degree distribution, and mean pairwise distance) with time. We are applying our techniques to develop characterizations of the eDonkey/Overnet and BitTorrent P2P networks in ongoing work.

References

- [1] L. A. Adamic, R. M. Lukose, B. Huberman, and A. R. Puniyani. Search in Power-Law Networks. *Physical Review E*, 64(46135), 2001.
- [2] E. Adar and B. A. Huberman. Free riding on gnutella. *First Monday*, 5(10), Oct. 2000.
- [3] F. S. Annexstein, K. A. Berman, and M. A. Jovanovic. Latency effects on reachability in large-scale peer-to-peer networks. In *Symposium on Parallel Algorithms and Architectures*, pages 84–92, 2001.
- [4] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making Gnutella-like P2P Systems Scalable. In *SIGCOMM*, 2003.
- [5] 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*, July 2002.
- [6] clip2.com. Gnutella: To the Bandwidth Barrier and Beyond, Nov. 2000.
- [7] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *SIGCOMM*, 1999.
- [8] A. Fisk. Gnutella Dynamic Query Protocol v0.1. Gnutella Developer’s Forum, May 2003.
- [9] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In *SOSP*, 2003.
- [10] Q. He and M. Ammar. Congestion Control and Message Loss in Gnutella Networks. In *Multimedia Computing and Networking*, Jan. 2004.
- [11] M. Jovanovic, F. Annexstein, and K. Berman. Modeling Peer-to-Peer Network Topologies through “Small-World” Models and Power Laws. In *TELFOR*, Nov. 2001.
- [12] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos. Is P2P dying or just hiding? In *Globecom*, Nov. 2004.
- [13] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy. Transport Layer Identification of P2P Traffic. In *International Measurement Conference*, Oct. 2004.
- [14] P. Karbhari, M. Ammar, A. Dhamdhere, H. Raj, G. Riley, and E. Zegura. Bootstrapping in Gnutella: A Measurement Study. In *PAM*, Apr. 2004.
- [15] A. Klemm, C. Lindemann, M. Vernon, and O. P. Waldhorst. Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems. In *Internet Measurement Conference*, Oct. 2004.
- [16] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the Kazaa Network. In *WIAPP*, 2003.
- [17] J. Liang, R. Kumar, and K. W. Ross. The KaZaA Overlay: A Measurement Study. *Computer Networks Journal (Elsevier)*, 2005.
- [18] J. Liang, R. Kumar, Y. Xi, and K. W. Ross. Pollution in P2P File Sharing Systems. In *INFOCOM*, Mar. 2005.
- [19] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. In *International Conference on Supercomputing*, 2002.
- [20] Q. Lv, S. Ratnasamy, and S. Shenker. Can heterogeneity make Gnutella scalable? In *IPTPS*, 2002.
- [21] 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.
- [22] S. Saroiu, P. K. Gummadi, and S. D. Gribble. Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. *Multimedia Systems Journal*, 8(5), Nov. 2002.
- [23] S. Sen and J. Wang. Analyzing Peer-To-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 12(2):219–232, Apr. 2004.
- [24] slyck.com. slyck.com. <http://www.slyck.com>, 2005.
- [25] K. Sripanidkulchai. The popularity of Gnutella queries and its implications on scalability. <http://www-2.cs.cmu.edu/kunwadee/research/p2p/paper.html>, Jan. 2001.
- [26] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems. In *INFOCOM*, 2003.
- [27] K. Sripanidkulchai, B. Maggs, and H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In *Internet Measurement Conference*, Oct. 2004.
- [28] D. Stutzbach and R. Rejaie. Capturing Accurate Snapshots of the Gnutella Network. In *Global Internet Symposium*, pages 127–132, Mar. 2005.
- [29] D. Stutzbach and R. Rejaie. Characterizing Churn in Peer-to-Peer Networks. Technical Report 2005-03, University of Oregon, May 2005.
- [30] D. Stutzbach and R. Rejaie. Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers. In *Passive and Active Measurement Workshop*, Extended Abstract, pages 353–357, Mar. 2005.
- [31] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network Topology Generators: Degree-Based vs. Structural. In *SIGCOMM*, 2002.
- [32] D. J. Watts. Six Degrees. In *The Essence of a Connected Edge*. ACM Press, 2003.
- [33] R. H. Wouhaybi and A. T. Campbell. Phenix: Supporting Resilient Low-Diameter Peer-to-Peer Topologies. In *INFOCOM*, 2004.
- [34] B. Yang and H. Garcia-Molina. Designing a Super-Peer Network. In *International Conference on Data Engineering*, Mar. 2003.
- [35] B. Yang, P. Vinograd, and H. Garcia-Molina. Evaluating GUESS and Non-Forwarding Peer-to-Peer Search. In *IEEE International Conference on Distributed Systems*, 2004.