Characterizing Today's Gnutella Topology

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

ABSTRACT

Gnutella represents a popular class of peer-to-peer (P2P) networks that are known as unstructured P2P networks, and has served as a real-world testbed for measurement-based characterization of these networks. Due to its open nature, the Gnutella protocol has undergone a series of gradual improvements over the years. To address the well known scalability problem with flood-based search techniques in the face of growing user population, two key features were introduced to the protocol: *(i)* a semi-structured topology, and *(ii)* a dynamic querying mechanism. Despite its importance, to our knowledge, no characterization of the Gnutella topology has been conducted during the last few years.

In this paper, we present a detailed characterization of the Gnutella overlay topology based on a recent measurements. We present a set of techniques *(i)* to efficiently capture accurate snapshots of the Gnutella network, and, more importantly, *(ii)* to properly quantify the accuracy of the captured snapshots. Using a new crawler that incorporates these techniques, we characterize different properties of today's Gnutella topology, examine their underlying causes, and investigate their implications. Our characterizations not only shed light on the current status of the Gnutella network but more importantly provide a better understanding of several fundamental challenges in the design of unstructured overlays.

Keywords

Peer-to-peer, Gnutella, topology, search

1. INTRODUCTION

The Internet has witnessed the explosive growth in popularity of Peer-to-Peer (P2P) networks which in turn has led to an astounding increase in network usage by these applications, in particular for file-sharing. P2P systems present an alternative communication paradigm that enables a distributed group of participating peers to collaborate and share their resources (*e.g.*, files). In particular, unstructured P2P networks are extremely popular on the Internet. Gnutella is the most well known and most popular non-proprietary example in this class (thus this class is often called Gnutella-like P2P networks). In these systems there is neither direct control over the formation of the P2P overlay topology nor over file placement among participating peers. The overlay topology is formed by peers joining the network based on some loosely defined (and possibly different) set of rules, and may change their connections to the network in response to departing neighbors. Therefore, the overlay topology in these systems is dynamically changing (*i.e.*, a moving target). The simplest query method to search for a resource file in unstructured P2P networks is propagating the query to all neighbors within a certain distance (*i.e.*, flooding) [1]. This search mechanism is known to be unscalable since it generates heavy network load across participating peers.

Recently, there has been growing interest in measurement-based characterization of unstructured P2P networks primarily due to their extreme popularity over the Internet, and their resiliency to the dynamics of peer participation. Deriving such characterizations provides unique insights into the behavior of these decentralized systems in a real setting (*i.e.*, realistic group size, degree of heterogeneity, workload, network and peer dynamics) which is very hard (if not impossible) to obtain through simulation or modeling. These characterizations deepen our understanding about the performance, dynamics, design anomalies and limitations of P2P systems in practice that are necessary to improve their design. Most of these empirical studies have focused on the Gnutella network because it provides a real-world testbed for the characterization of P2P networks with the following collection of unique properties: *(i)* heterogeneous peers with a variety of implementations of the protocol, *(ii)* geographically distributed, *(iii)* a large scale network with hundreds of thousands of concurrent peers, and *(iv)* an open protocol with several mature implementations. The open nature of the protocol has enabled developers and researchers to incorporate new ideas and discover new challenges. In a nutshell, the Gnutella network has significantly inspired and influenced both research and development of P2P networks.

There are three key aspects of unstructured P2P networks that can be characterized through measurement:

query workload [2], file distribution (or replication) [3] and overlay network topology [4]. While these three issues are equally important, to our surprise, little attention has been given to characterization of the Gnutella network topology. We are only aware of two previous studies on this issue. Ripeanu et al. [4] mapped the Gnutella network, and Saroiu et al. [5] briefly examined the resiliency of the Gnutella overlay topology in the face of attack. These studies are clearly inadequate to characterize today's Gnutella overlay for two important reasons: First, these two studies are outdated (almost three years old), and have only conducted a limited analysis on the Gnutella overlay. During the past couple of years, the Gnutella network has grown by more than an order of magnitude, and the protocol has undergone major changes. In particular, to improve the scalability of the Gnutella protocol, the notion of *Ultrapeers* (or super-peer) was introduced [6] in order to add a *semi-structure* to the Gnutella network. There has not been any study to characterize the Gnutella overlay since this semi-structured was incorporated into the overlay. Second, measurement-based characterization of large scale P2P networks is inherently difficult. A common approach in these studies is to examine properties of snapshots of the system captured using a crawler. However, capturing accurate snapshots of these systems is hard for two reasons: *(i)* the dynamics nature of peer participation (*i.e.*, churn), and *(ii)* a significant portion of peers are unreachable. Previous studies either deployed slow crawlers which lead to distorted (*i.e.*, stretched) snapshots of the system [4], or partially crawled the network [5] which could result in biased (and non-representative) snapshots. To our knowledge, none of the previous measurement-based studies have quantified the accuracy of their captured snapshots.

In this paper, we present a detailed characterization of the Gnutella overlay topology based on recent measurements. This study makes two important contributions: First, we present a set of techniques: *(i)* to efficiently capture accurate snapshots of the Gnutella network, and more importantly, *(ii)* to properly quantify the accuracy of the captured snapshots. To achieve these goals, we have developed a new Gnutella crawler, called *Cruiser*. Cruiser can effectively leverage the semi-structure in the Gnutella network to reduce the duration of each crawl by an order of magnitude compared to previous crawlers. Furthermore, it uses the new handshaking mechanism in Gnutella to obtain fresh information from each peer. Therefore, Cruiser can capture significantly more accurate snapshots of the network. Having more accurate snapshots allows us to examine the dynamics of the overlay in more detail, and more importantly, to quantify the accuracy of captured snapshots and specify an upper bound for potential error in our characterization.

Second, using our crawler, we have collected more than 10,000 snapshots of the Gnutella network during the past 6 months. We use this dataset to characterize different properties of today's Gnutella topology, examine their underlying causes, and their implications on the Gnutella network. Our characterizations not only shed light on the current status of the Gnutella network but more importantly raises several interesting issues and problems that are relevant to the popular class of Gnutella-like unstructured P2P networks. In particular, our characterization reveals the following:

- The overall node degree does not exhibit a power-law distribution, differing from previous studies [1, 4, 7].
- A non-negligible portion of ultrapeers are cannot accept incoming connections.
- The size of the Gnutella network has dramatically grown over the past couple of years. Despite this increase, the diameter of the topology remains low. More importantly the distribution of pair-wise path lengths has become more homogeneous with lower mean value. These desired properties have been maintained by the introduction of semi-structure to the topology, and increasing the degree of peers in the top-level overlay.
- The overall topology has become denser and exhibits clear small-world properties.
- Despite variations in the total number of peers with time of day, a large number of peers are available at any time, and the semi-structure remains balanced. (*i.e.*, the ratio between leaf to ultrapeers remains relatively constant).

The rest of this paper is organized as follows. In Section 2, we present our measurement methodology, and describe how we capture and postprocess snapshots of the Gnutella network, and examine their accuracy. A detailed characterization of the Gnutella overlay is presented in Section 3. Section 5 presents a summary of related work. Finally, Section 6 concludes the paper and presents our future plans.

2. MEASUREMENT METHODOLOGY

Our goal is to capture accurate snapshots of the Gnutella network. In practice, two factors reduce the accuracy of captured snapshots: *(i) the Length of a Crawl*: Because of the dynamic nature of peer participation, the longer a crawl takes, the more distorted the captured snapshot becomes. *(ii) Unreachable Peers*: A significant portion of discovered peers in each snapshot are not directly reachable. Therefore, information about edges of the overlay that are connected between these unreachable peers might be missing from the captured snapshots. To address these issues, we provide a brief description of modern Gnutella, and an overview of Cruiser. Then, we examine the problem of unreachable peers.

2.1 Modern Gnutella

Similar to many unstructured P2P networks, each Gnutella peer joins the network by establishing separate TCP connections to several existing peers. In the original Gnutella protocol, participating peers form a flat overlay in a rather ad-hoc fashion 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 mechanism as follows:

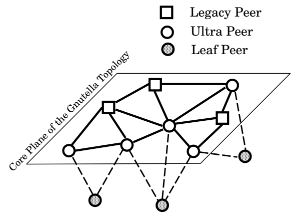


Figure 1: Semi-Structured Topology of Modern Gnutella

(i) Semi-structured Overlay: modern Gnutella clients implement a two-tiered overlay structure by dividing peers into two groups: *ultrapeers* (or super-peers) and *leaf* peers. As shown in Figure 1, each ultrapeer neighbors with several other ultrapeers within the top-level overlay. The majority of the peers are leaves that are connected to the overlay through a few ultrapeers. High-bandwidth, un-firewalled leaf peers become ultrapeers on demand in order to maintain a proper ultrapeer-to-leaf ratio in the overlay. 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. We also refer to the legacy peers and ultrapeers collectively as the *top-level* 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] to only gather enough results to satisfy the user. It is similar in principle to an expanding ring search. Rather than simply 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 many 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: $Searched\_Ultrapeers = \sum_{i=0}^{\mathrm{TTL}-1} (\mathrm{d}-1)^i$ where d denotes the connection degree of the neighbor. This equation simply assumes that all peers have the same connection degree. Finally, modern Gnutella clients implement a special handshaking feature [9] that enables the crawler to quickly query a peer for a fresh list of its current neighbors. Cruiser uses this feature as we describe in the next subsection.

2.2 Gnutella Crawler

We have designed and developed a new Gnutella crawler, called *Cruiser*. Cruiser begins with an initial set of known ultrapeers. Then, it progressively contacts known ultrapeers to obtain several pieces of information including: *(i)* Peer type (ultrapeer, leaf, or legacy), *(ii)* Implementation and version, *(iii)* A list of the peer's neighbors, and *(iv)* A list of an ultrapeer's leaf nodes. Each newly discovered neighbor is added to the queue of new peers to be contacted. While the basic crawling strategy by Cruiser is similar to other crawlers, Cruiser significantly improves the accuracy of captured snapshots by incorporating the following techniques: First, Cruiser uses the handshaking mechanism in modern Gnutella [9] to quickly obtain a fresh list of current neighbors from each peer. Previous crawlers relied on other features of the Gnutella protocol, namely Ping-Pong messages, to retrieve this information. These techniques were less efficient and potentially less reliable. 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. Furthermore, the high degree of peer connectivity within the top level overlay substantially increases the rate of discovery for new ultrapeers. Overall, this strategy leads to a major reduction in the duration of a crawl without loss of information. Third, Cruiser employs a master-slave architecture in order to achieve a high degree of concurrency and to effectively utilize available resources on multiple desktop PCs. A master process coordinates among multiple slave processes that act as virtually independent crawlers and crawl the network in parallel. To further improve the degree of concurrency, each slave process uses asynchronous communications to maintain hundreds of open connections in parallel. Fourth, Cruiser implements an adaptive load management mechanism to ensure that slaves processes remain busy but do not become overwhelmed. This is important for the steady progress of the crawl especially when different slave nodes have heterogeneous processing capabilities. Toward this end, Cruiser enables each slave process to adjust its own load (*i.e.*, number of open connections) using an AIMD algorithm similar to TCP's congestion control mechanism.

These techniques result in a significant increase in crawling speed. Cruiser can capture a snapshot of the Gnutella network with 300–400K peers *in less than 4 minutes* using 8 off-the-shelf 1GHZ Linux boxes in our lab. This is several orders of magnitude faster than previously reported crawlers (*i.e.*, 2 hours for 30K peers in [4], and 2 minutes for 5K peer in [5]). It is worth clarifying that while our crawling strategy is aggressive and our crawler requires a considerable amount of local resources, its behavior is not intrusive since each top-level peer is contacted only once per crawl.

Post-Processing: Once required information is collected from all reachable peers, we perform some post-processing to remove any obvious inconsistencies that might have been introduced due to the dynamic changes in the topology during the crawling period. Toward this end, we enforce the following rules on each snapshot:

1. All links are bidirectional.

2. Any peer that is the neighbor of an ultrapeer or legacy peer must also be an ultrapeer or legacy peer.

3. Any parent of a leaf peer must be an ultrapeer.

The first rule is violated, when only one end node declares a connection, by about 1% of ultrapeers in a snapshot . This occurs when a connection is established or closed during a crawl. We properly adjust the snapshot by considering all these connection to be bidirectional, *i.e.*, include an edge between those nodes. Also a very small portion (<0.5%) of detected peers violate rules 2 and 3. These peers were a leaf when we contacted them but became an ultrapeer during the crawl. Therefore, we consider them as ultrapeers in the captured snapshot.

2.3 Unreachable Peers

In any arbitrary crawl, a significant portion (almost 30%–38%) of discovered top-level peers in each crawl are not directly reachable by a crawler. This means that the TCP connection to these peers either timed out (15%–24%)[1], closed prematurely (6%–10%) or was refused (5%–7%) by the contacted peers. 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. Note that there is no reliable test to distinguish between departed and firewalled peers because firewalls can time out or refuse connections depending on their configuration and congested peers can timeout or drop connections. In theory, it should not be possible for a firewalled peer to become an ultrapeer. Any firewalled peer in the top-level overlay should be a legacy peer. However, we cannot rule out buggy code permitting firewalled leaves to become ultrapeers in some circumstances.

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 on captured snapshot as follows: First, we devised the following simple technique to identify the ratio of departed peers in each snapshot. We performed back-to-back crawls to capture two snapshots. Then, the unreachable peers in the first snapshot that were missing from the second snapshot, are considered "departed peers" during the first snapshot. This approach reveals that departed peer constitute only 2–3% of unreachable peers in each snapshot.

Second, detailed examination of the remaining unreachable peers led to a surprising discovery: some of the peers that refuse connections are actually overwhelmed ultrapeers that sporadically accept TCP connections and can be contacted after several attempts. This suggests that the application is not able to call *accept()* sufficiently fast which leads to a TCP listen buffer overflow. We also noticed that connections to most of these overwhelmed ultrapeers exhibit long RTT (> 1 second) and hardly any packet loss. This indicates that their CPU is the bottleneck, likely due to other applications running on these systems. Despite this finding, we did not incorporate the multiple attempt strategy into the crawler for two reasons: *(i)* it only marginally increases the number of reachable peers at the cost of significant increase in the duration of each crawl which in turn increases distortion in captured snapshots, and *(ii)* it is intrusive and may exacerbate the existing problem. Therefore, the results presented in this paper are obtained with only one attempt to connect to each peer.

Third, we examined those unreachable peers that experienced application-level timeout. Since overwhelmed ultrapeers are unlikely to exhibit this behavior, we hypothesized that these group of peers are firewalled. To establish if these nodes were firewalled or merely congested, we randomly selected 1000 peers (about 3%) that timed out, and attempted to contact them every 5 minutes[2]. Interestingly, more than 92% of these peers were not reachable at all after several hours of trying. This implies that timeout is a good indicator for firewalled peers. *In summary, our investigation revealed that in each crawl, 2%–3% of unreachable peers are departed, 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.*

The unreachable peers can introduce the following errors in the captured snapshots of the Gnutella topology:

1. Connections between two unreachable ultrapeers, or unreachable ultrapeers and firewalled legacy peers are missing from a snapshot.

2. Connections between unreachable ultrapeers and their leaves are missing from a snapshot.

3. Some leaves that are only connected to unreachable ultrapeers are not being discovered at all.

One key issues is to quantify the resulting error due to unreachable peers on the captured topology. Note that in order to miss a connection in the overlay, both end nodes must be unreachable. Let us pessimistically assume that all of the unreachable nodes are overwhelmed ultrapeers. If all peers have roughly the same degree, and unreachable nodes do not have a strong bias towards being connected to other unreachable nodes, then the probability that both end nodes of a connection would be unreachable is approximately $1-(1-p)^2$, where p is the fraction of unreachable nodes. As our results show, the degree of connectivity among top-level peers is indeed fairly homogeneous which implies that at most only 9%–15% of edges in the overlay could be missing from our captured snapshot. Since firewalled legacy peers can not be connected together (*i.e.*, can not be located at both end of a missing edge) and they constitute more than half of the unreachable peers (as we discussed above), the actual portion of missing edges is considerably smaller.

2.4 Quantifying Snapshot Accuracy

We examine two dimensions of snapshot accuracy: *(i) Completeness* presents the portion of participating peers that were captured, and *(ii) Freshness* is an indication of

[1] We examined the effect of different timeout values and set it to the value that results in minimum number of false positive timeouts without significantly increasing the duration of a crawl.

[2] Note that each attempt translates into several attempts by TCP to establish a connection by sending SYN packets.

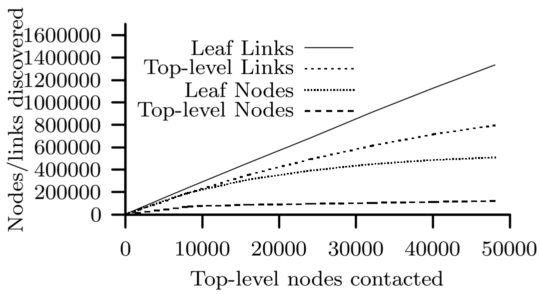


Figure 2: Cumulative information per contacted ultrapeer

the introduced distortion in a snapshot due to the duration of a crawl. As we discussed earlier, there is a clear tradeoff between these two dimensions, *i.e.*, improving completeness through multiple attempts to contact each peer increases the length of each crawl and thus reduces the freshness of captured snapshot.

Completeness of Snapshots: To examine the completeness of captured snapshots by Cruiser, we kept track of the following variables during each crawl: the number of discovered top-level peers, the number of leaves, the number of top-level links, and the number of links to leaves. Figure 2 shows the cumulative value of these four variables as a function of the number of contacted peers in a sample crawl. This figure shows that the number of discovered top-level peers and leaves clearly curve off which indicates that Cruiser has captured a majority of the participating peers. The number of top-level links only somewhat curves off, due to unreachable top-level peers. Finally, links to leaves linearly increases with the number of visited top-level peers because each ultrapeer provides a unique set of links between itself and its leaves.

Impact of Crawling Duration: To examine the impact of crawl duration on the accuracy of captured snapshots, we modified Cruiser to stop the crawl after a specified period. Then, we performed two back-to-back crawls and repeated this process for different durations. We define δ_+ and δ_- as the number of new and missing peers in the second snapshot compare to the first one. Figures 3 presents the relative value $\delta = \delta_+ + \delta_-$ (normalized by the total number of peers in the first crawl) as well as the total number of discovered peers as a function of the crawl duration for all participating peers (both top-level and leaves). During short crawls (left side of the graph), δ is high because the captured snapshot is incomplete, and each crawl captures a different subset. As the duration of crawl increases, δ decreases which indicates that the captured snapshot becomes more complete. Increasing the crawl length beyond four minutes does not decrease δ any further, and achieves marginal increase in number of discovered peers. This figure reveals a few important points. First, there exists a "sweet spot" for crawl duration beyond which crawling has diminishing returns if the goal is simply to capture the population. Second, for sufficiently long crawls, Cruiser can capture a relatively un-stretched snapshot. Third, the change of $\delta = 8$ is an upper-bound on the distortion due to the passage of time as Cruiser runs. The relatively flat delta on the right suggest that around 4% of the network is unstable and turns over quickly, while the remainder of the network is fairly stable during the crawls.

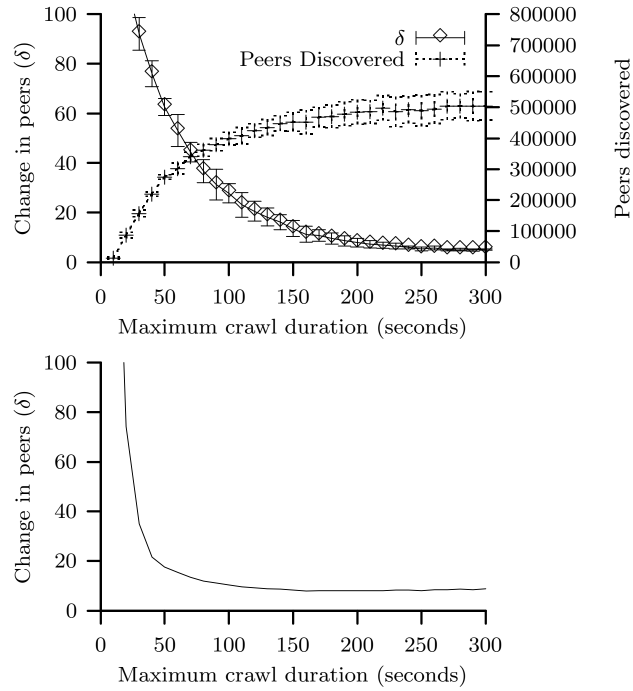


Figure 3: Error as a function of maximum crawl duration, generated by running two crawls back-to-back for each x-value and computing the δ. Averaged over 8 runs with standard deviation shown.

Sampling vs Complete Snapshots: We argue that sampling a snapshot of unstructured networks is not an appropriate technique for an initial characterization for the following reasons: *(i)* in the absence of adequate knowledge about the 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 whereas slow crawling can easily lead to a snapshot that is biased towards peers with short uptime. *(ii)* More importantly, some "graph-level" characteristics of the overlay topology, such as the mean shortest path between peers (which we discuss in Subsection 3.3), require the entire snapshot and cannot be derived from samples. Because of these reasons, we collect complete snapshots and use them for our characterizations.

3. CHARACTERIZATION OF GNUTELLA

In this section, we present the following characterizations of the modern Gnutella topology: *(i)* implementation heterogeneity, *(ii)* several angles of distribution of node degrees, *(iii)* reachability, diameter and density of the overlay, *(iv)* small-world properties, and *(v)* variations of different properties of the topology with time. To characterize properties of the Gnutella overlay topology, we treat the overlay as a graph and apply different forms of graph analysis to examine its properties. Note that the top-level overlay can properly represent a common Gnutella-like unstructured P2P overlay. Therefore, we primarily focus on the properties of the top-level overlay.

Data Set: We have captured more than 10,000 snapshots

Crawl Time	Total Nodes	Leaves	Top-level	Top-Level Edges	Unreachable
09/27/04	725120	614912	110208	2425544	35796
10/11/04	779535	662568	116967	2488438	41192
10/18/04	806948	686719	120229	2663490	36035

Table 1: Sample Crawl Statistics

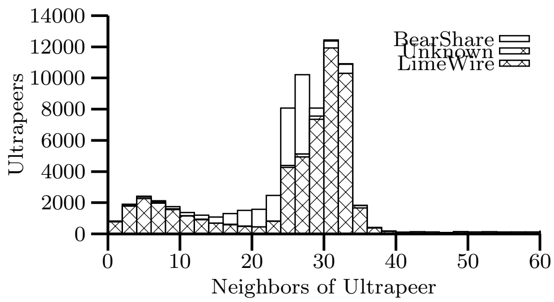


Figure 4: Ultrapeer Degree by Version

Implementation	Perc.	Revised Perc.
LimeWire	64%	74%
BearShare	18%	19%
Unknown	17%	6%
Other	< 1%	< 1%

Table 2: Distribution of Implementation

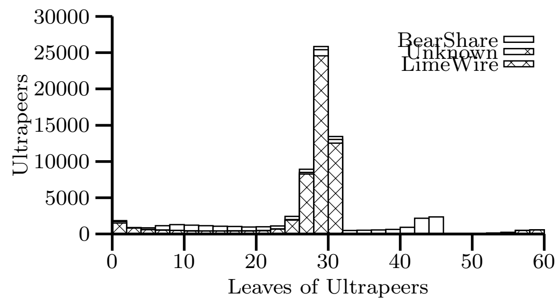


Figure 5: Ultrapeer Leaves by Version

of the Gnutella network during the past six months (Apr.–Oct. 2004) with Cruiser. We have crawled the network in different patterns in order to ensure that captured snapshots are representative. 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. Table 1 presents summary information of three sample snapshots after post-processing, including date, total number of discovered peers, their breakdown between top-level and leaves, top-level edges and unreachable peers.

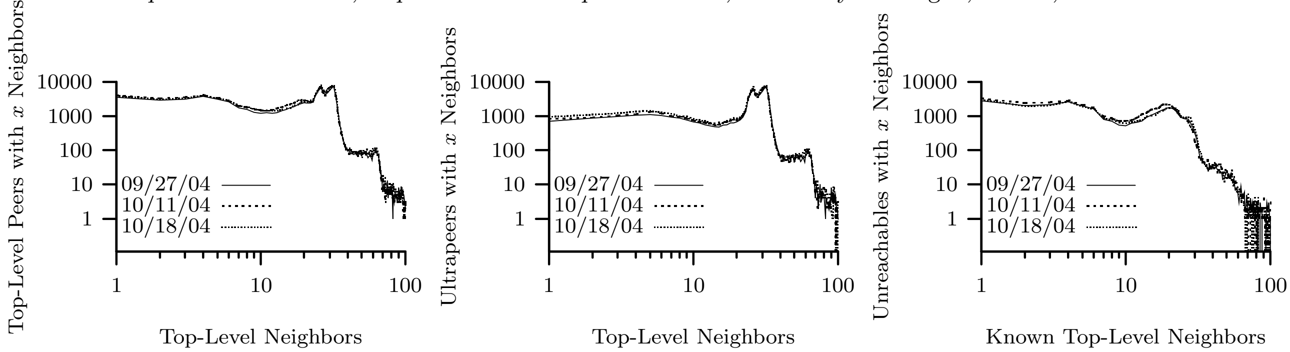
The presented results in this section are primarily from the snapshots in Table 1. However, we have examined many other snapshots and observed similar trends and behaviors. Therefore, the presented results are representative. Presenting different angles of the same subset of snapshots, allows us to conduct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

3.1 Implementation Heterogeneity

The open nature of the Gnutella protocol has led to several known (and possibly many unknown) implementations where each implementation periodically releases an improved version. Although implementations can interoperate, the extensibility of the protocol allows developers to introduce new features (or use different parameters) in their implementations. This has naturally led to heterogeneity among coexisting implementations over the network. 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 Gnutella 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 a single crawl (conducted on 4/30/2004). This table shows that a clear majority of contacted ultrapeers use the LimeWire implementation. The Unknown implementation represents those peers that did not identify their software during the crawl. We later contacted about half of these unknown hosts using HTTP, and verified that almost 2/3 of them are running LimeWire. The third column of Table 2 shows the revised the distribution of implementations after this adjustment. We further examined the distribution of different version among LimeWire ultrapeers and discovered that a majority of them (around 94%) use the most recent version of the software available at the time of the crawl (3.8.x). Interestingly, almost half of these ultrapeers (45%) ran the most recent release (3.8.10), and some users (3%) even ran development versions that had not been released yet! Overall, these results reveal that while heterogeneity exists, a clear majority of Gnutella users run a recent version of LimeWire. Since most users promptly upgrade their software, new features can rapidly gain widespread use.

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 topology[3]. Figure 4 depicts the distribution of ultrapeer degree within the top-level overlay and its breakdown across different implementations for the crawl conducted on 10/11/04. Figure 5 depicts the distribution of node degree from ultrapeers towards their leaf nodes (*i.e.*, outgoing degree from the top-level overlay) as well as its breakdown across different implementations in the same crawl. Figure 5 reveals that LimeWire and BearShare ultrapeers implementation prefer to serve 30 and 45 leaves, respectively, whereas both try to maintain around 30 neighbors in the top-level over-

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



(a) Top-level Degree Distribution (b) Reachable Degree Distribution (c) Unreachable Degree Distribution

Figure 6: Different angles of the top-level degree distribution in Gnutella topology

lay (Figure 4).

3.2 Node Degree Distributions

The introduction of the semi-structure in the topology along with the distinction between ultrapeers, leaves, and legacy peers in the modern Gnutella protocol demands a closer examination of the different degree distribution among different group of peers. Clearly, such an examination was not applicable to the original Gnutella network.

Node Degree in Top-Level Overlay: Previous studies reported that the distribution of node degree in the Gnutella network exhibited a power-law distribution [4, 1] and later changed to a two-segment power-law distribution [7, 4]. To verify this property for the modern Gnutella network, Figure 6(a) depicts the distribution of node degree among all peers (both unreachable and reachable) in the top-level overlay for the three sample snapshots presented in Table 1. This distribution has a spike around 30 and does not follow a power-law.

Because we were unable to contact each 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 6(b) and 6(c). These depict the neighbor degree distribution for reachable and unreachable peers respectively. The data in Figure 6(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 data. Figure 6(c) presents the observed degree distribution among unreachable top-level peers. This data is more strongly biased low since we cannot observe the connections between any pair of these peers. In this data, a much greater fraction of peers have an observed degree below 30. Many of these peers have a degree closer to 30, with the true distribution likely similar to that in Figure 6(b).

The degree distribution among contacted top-level peers has two distinct segments around a spike in degree of 30, resulting from LimeWire and BearShare's behavior of attempting to maintain that many neighbors. The peers with lower degree are peers which have not yet established 30 connections. 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. In older versions of Gnutella, the user was allowed to set their desired degree through the GUI, which led to the power-law behavior. The most popular modern Gnutella implementations discourage this level of user-tampering with protocol internals.

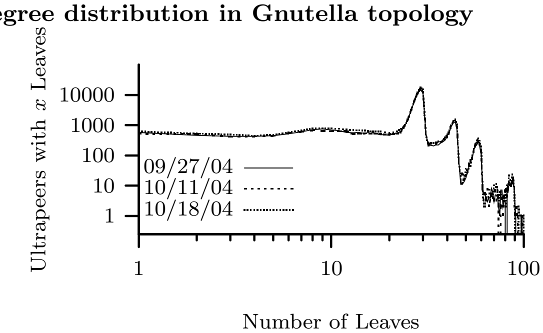


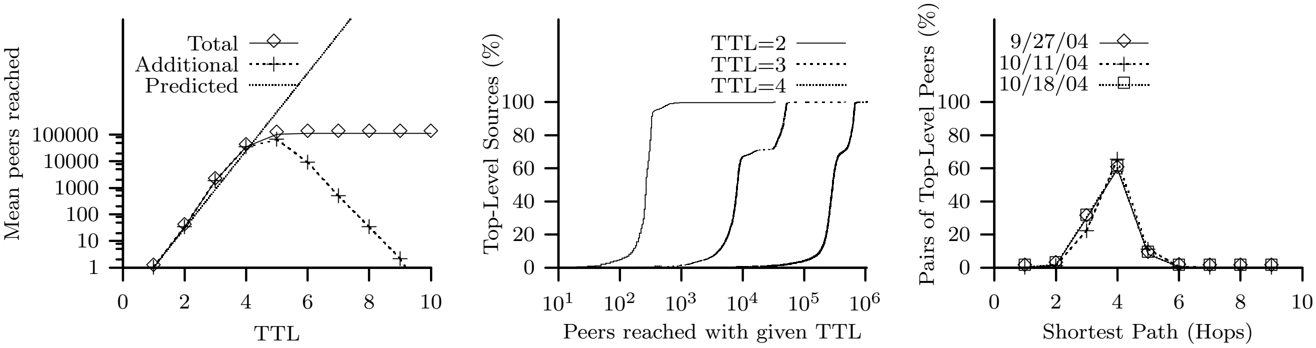
Figure 7: Degree dist. from ultrapeers to leaves

Node Degree To/From 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 7 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 (as shown in Figure 5). 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 open slots for new leaves to join the overlay.

In Figure 8, we present the degree of connectivity among leaf peers. This result reveals that most of leaf peers connect to only a small number of ultrapeers (3 or less), a small fraction of leaves connect to several ultrapeers, and few leaves connect to an extremely large (between 40 to 100) number of ultrapeers. We provide further discussion on these high degree peers in Section 4. LimeWire attempts to maintain three ultrapeers per leaf by default. Because we may be missing some leaf to ultrapeer edges, these results are biased slightly low, and many of the leaves with only one or two ultrapeers actually have two or three ultrapeers.

3.3 Reachability

Two equally important properties of an overlay topology directly affect the behavior of flood-based search: *(i)* the



(a) Mean Peers Searched by TTL

(b) Cumulative Peers Searched CDF

(c) Ultrapeer-to-ultrapeer shortest paths

Figure 9: reachability, diameter, and shortest path in Gnutella topology

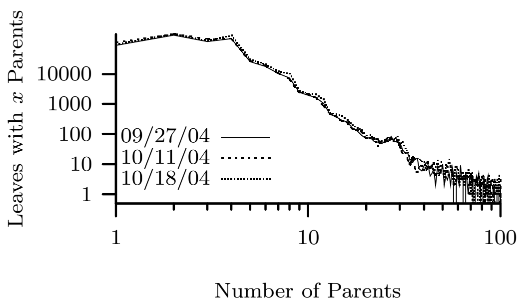


Figure 8: Leaf Parents

number of reachable peers for different TTL values, and *(ii)* the distribution of the pair-wise distance between arbitrary pairs. Our goal is to examine these two properties for today's Gnutella topology. Due to the high computational cost of calculating the all-pairs shortest path on such a large graph, these results are based on the top-level topology only. Since each leaf is directly connected to the top-level, reachability characteristics for leaves are not very different. Figure 9(a) depicts the *mean* number of newly visited peer and its cumulative value as a function of TTL, averaged across top-level peers in the 9/27/2004 snapshot of the topology. The shape of this figure is very similar to the result that was reported by Lv, et al. (Figure 3 in [7]) which was captured in October 2000, with a significantly smaller number of peers (*i.e.*, less than 5000 peers). 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 growth of the network has been balanced by the introduction of ultrapeers and an increase in node degree. Thus, while the network has changed in many ways, the percentage of newly searched peers per TTL has remained relatively stable. Figure 9(a) also shows the number of newly visited peers that is predicted by the Dynamic Querying formula (assuming a node degree of 30), that 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 9(b) shows a different angle on reachability by presenting the CDF of the number of peers searched, from all peers for different TTL values. We use a logarithmic x-scale to magnify the left part of the figure for lower TTL values. This figure shows two interesting points: First, the total number of visited peers with TTL value of n is almost always an order of magnitude higher than the same number with TTL value of $(n-1)$. Second, excluding the little step in the middle of the CDF for TTL=3, the number of newly reachable peers for each TTL is fairly consistent across a majority of participating peers for all three TTL values *These findings basically imply that all peers have roughly similar opportunity to reach other peers the network, and the number of reachable peers through flood-based query is primarily determined by the TTL value and not by a peer's location,* i.e., *the topology is pretty balanced*.

To determine the cause for the step-like change in the middle CDF (with TTL 3), we closely examined our snapshots and reached a surprising result. 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. About 30% of the top-level peers are within 2 hops of at least one of these high-degree peers. Thus, 30% of peers observe a significant increase in the number of reachable peers with TTL=3 via these high-degree peer. In a nutshell, these high-degree peers are widely "visible" throughout the overlay, and thus receive a significant portion of submitted queries and responses from other peers. To our great surprise, it appears someone is using the overlay to monitor queries and query replies, presumably to locate copyright infringement among Gnutella users!

To examine the impact of growth in network size on the pair-wise distance between participating peers, Figure 9(c) shows the distribution of shortest-path lengths in terms of overlay hops among all pairs of top-level peers from three snapshots. Ripeanu et al. [4] presented a similar result 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 indicate that the pair-wise path between peers over the Gnutella

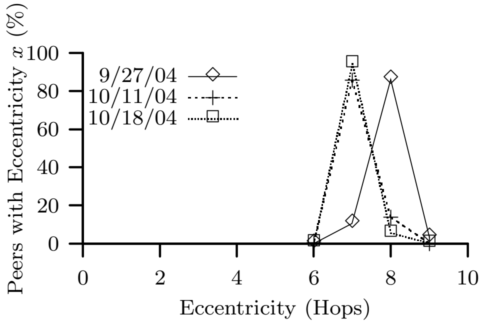


Figure 10: Distribution of Eccentricity in the Top-level Overlay

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. Also, the results from our crawls our nearly identical; in [4] there is considerable variance from one crawl to another. Thus, the path lengths have become both more homogeneous and more stable.

The longest observed path by our crawler in these three snapshots was 9 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, which is the shortest path distance from a node to its furthest neighbor. In other words, for some node i, and the function $P(i, j)$ that returns the shortest path distance between nodes i and j, the eccentricity, E_i of i is defined as follows:

$$E_i = \max(P(i, j)\forall j)$$

Figure 3.3 shows the distribution of eccentricity in three topology snapshots. This figure shows that distribution of eccentricity is homogeneous and low which is another indication that overlay graph is a well-connected mesh, rather than a chain of multiple groups of peers.

3.4 Small World

Recent studies have shown that many biological and man-made graphs (*e.g.*, collaboration among actors, electrical grid, and the WWW graph) exhibit "small world" properties. In these graphs, nodes are highly clustered and the mean pair-wise distance between nodes is small compared to random graphs with the same number of vertices and edges. A study by Jovanic et al. [10] in November-December 2000 concluded that the Gnutella network exhibit 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 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 computed using the following standard formula:

$$C_{actual} = \frac{\sum_i C(i)}{|V|}, \quad C(i) = \frac{D(i)}{D_{max}(i)}$$

$D(i)$, $D_{max}(i)$ and $|V|$ denote the number of connection between neighbors of node i, the maximum possible connections 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 links between them, so $D_{max}(A) = 3$. If only two of them are connected together, that's one link 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 several recently captured snapshots of the Gnutella top-level overlay 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 Jovanic et al. [10] and three classic small world graphs [11]. A graph is loosely identified as 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 implies that modern Gnutella still exhibits small world properties.

The high clustering coefficient is actually bad for Gnutella. It means that a flood with TTL=2 is more likely to generate redundant messages than in a random graph. Constructing a less clustered unstructured overlay in a distribution fashion is an open problem.

3.5 Variations of Network Properties with Time

In this subsection, we examine the dynamics of the topology over time. Figure 11 depicts the total number of identified peers and its breakdown among ultrapeers and leaves from snapshots of the network that were collected back-to-back over several days in April 2004. The periodic time-of-day effect is clearly visible from viewing the whole week at once. We also zoomed in and examined the time of day effects in more detail than can be seen in this figure. On weekdays, the number of users starts to increase at around 4am PDT (7am EDT), and gradually increases until around 1pm PDT (4pm EDT). It remains relatively stable until around 6pm PDT (9pm EDT), then it goes into a steep decline until 11pm PDT (2am EDT) where it levels off again. On weekends, the surge is much wider and flatter. Thus, the active population is largest during the evenings in the United States and during the day on the weekends.

While the total number of peers can change about 25% between day and night, a large number of peers (a quarter of a million) are always available. Furthermore, the ratio across different category remains surprisingly constant over time. This behavior is related to the dynamic mechanism that is used by leaf peers to promote themselves to ultrapeers when the number of ultrapeers in the system is low. Figure 12 shows the change in the leaf-to-ultrapeer

Graph	L_{actual}	L_{random}	C_{actual}	C_{random}
Modern Gnutella	4.47–4.63	3.75	0.012–0.014	0.00038
Original Gnutella [10]	3.30–4.42	3.66–5.54	0.02–0.03	0.002–0.006
Move 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

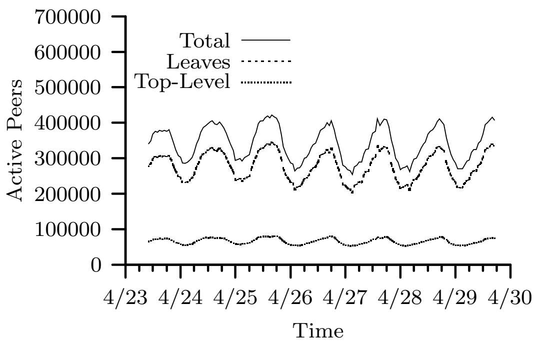


Figure 11: Peers over Time

ratio over time, and reveals that the ratio always remains between 13 to 16 over the course of these measurement. Note that this result only captures the dynamics of the population over a short window of time (*i.e.*, one week) which could be different from its long-term trends. We plan to examine longer term trends in our future work.

4. DISCUSSION

In this section, we provide further discussion on two interesting issues: *(i)* Implications of flood-based query, and *(ii)* Implications of high degree peers.

Implications on Flood-based Query: Sine the topology of the top-level overlay is denser than the topology of original Gnutella, the overhead of flood-based querying is proportionally higher because there are more redundant links. More specifically, flooding queries result in a significantly higher volume of traffic in the network. Fortunately, conservative strategies such as Dynamic Querying (as we discussed in Section 2.1) to some extent compensate for this problem by forwarding each query to a subset of neighbors. Query flooding by legacy peers could still be cause for concern. To address this, modern Gnutella client drop any query with a TTL higher than a threshold. Between the use of ultrapeers and dynamic querying, Gnutella is fairly efficient at locating popular files. However, searches for rare or non-existent files still effectively perform a flood. Efficient keyword searches for rare content in unstructured P2P networks is still an open problem.

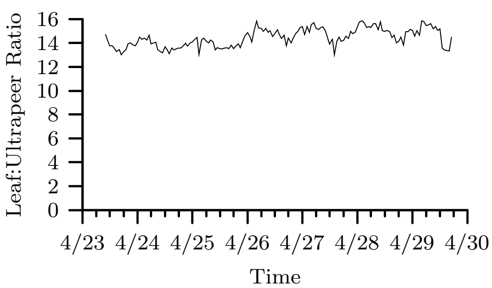


Figure 12: Ultrapeer:Leaf Ratio over Time

Implications of High Degree Peers: We have seen a few outliers with an unusually high degree of connectivity in all degree distributions in Subsection 3.2. When these peers are leaves, submitted queries by these peers reach a proportionally larger number of peers, and they may observe a proportionally larger number of queries from other peers.

However, as ultrapeers, the offered packet-forwarding load on a peer with degree of d is proportional to d^2. This is because it's incoming traffic is proportional to d and it must forward these message to $d-1$ neighbors. If these peers do not have sufficient resources (bandwidth or CPU) to manage the load, they simply drop packets (at the application level) which in turn degrades performance of the overlay. Thus, while some users may believe that increasing the degree of their ultrapeer will improve their search results, it is not actually worthwhile. Fortunately, the number of these greedy users is very small and does not have a severe impact on the system.

5. RELATED WORK

Properties of the Gnutella topology have been examined by a handful of previous studies. An interesting characterization on different aspects of the Gnutella topology was presented in [1] using their measurements prior to November of 2000. They showed that the topology consists of a few thousand peers and the distribution of node degree follows a power-law distribution. Lv, et al. [7] also examined properties of a snapshot of the Gnutella topology that was collected in October of 2000 as a sample for comparison with other graph models. They showed that the node degree distribution follows a two-segment power-law distribution. Jovanovic, et al. [10, 12] also examined the Gnutella topology based on snapshots they collected in December of 2000. They concluded that the Gnutella topology not only is properly represented by a power-law distribution of node degree, but exhibits strong small-world properties. The most recent work on mapping the Gnutella topology was conducted by Ripeanu, et al. [4] by examining snapshots of the network topology over a 6 month period (from November 2000 to May 2001) with up to 50,000 peers in a snapshot. They showed that the node degree distribution initially exhibited power-law distribution and later change to a two-segment power-law during this period. They also reported that the distribution of pair-wise shortest-paths over the topology which we used for comparison in our result section. All of these previous measurement studies on the Gnutella topology are more than two years old, and the Gnutella protocol has undergone major changes (mainly the introduction of semi-structure) during this period that directly affects its topology. The key contribution of our work is to capture and characterize the Gnutella topology after these important changes which has not been done by any previous work.

There has been a wealth of research on other aspects of Gnutella as well as other P2P systems. The widely cited work by Saroiu, et al. [5] examined aspects of both Gnutella and Napster systems, and mainly focused on characterization of the population, inter-peer bandwidth connectivity and latency, dynamics of group membership and the degree of cooperation among participating peers. However, they did not directly examine properties of the Gnutella topology. A recent measurement-based study by Karbhari, et al. [13] focuses on the impact of various bootstrapping mechanism in Gnutella's performance. Finally, there has been several modeling or simulation-based studies on improvement of search in Gnutella-like P2P networks [14, 15, 16, 17]. 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.*, [18, 19, 20]) and network topology generators (*e.g.*, [21]) are closely related to this 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.

6. CONCLUSIONS & FUTURE WORK

In this paper, we present a recent measurement-based study that characterizes several aspects of the Gnutella network topology after some major changes that have been made to the Gnutella protocol, mainly the introduction of semi-structure to the topology by grouping the peers into ultrapeers and leaf peers. We designed a crawler that leverages the loose structure of the topology to substantially reduce the crawling time which in turn significantly improves the accuracy of the captured snapshot of the Gnutella overlay topology compared to previous work. To our knowledge, this work is the only research study that has characterized the Gnutella topology during the past few years. Our findings can be summarized as follows: *(i)* the population of participating peers has dramatically grown to more than 800,000, *(ii)* a majority of peers are using recent versions of Gnutella implementations, *(iii)* despite growth in size, the diameter of the topology is still low through the increase in the degree of connectivity in the topology, *(iv)* the pair-wise path lengths are very homogeneous, *(v)* while the total number of peers in the network varies with time of day, a large number of peers are available at any time, and the ratio between leaf and ultra peers remains relatively constant, *(vi)* a good portion of ultrapeers alarmingly cannot accept incoming requests for new connections.

We plan to continue this work in several direction. We are actively monitoring the Gnutella network and plan to examine the dynamics of peer participation over short time scales, any longer term trends in the topology, and variations in several key properties (*e.g.*, small-world coefficient, degree distribution, and mean pair-wise distance) with time.

7. REFERENCES

[1] Clip2.com, Inc., "Bandwidth barriers to Gnutella scalability," Sept. 2001.

[2] Kunwadee Sripanidkulchai, "The popularity of gnutella queries and its implications on scalability," Jan. 2004, http://www.cs.cmu.edu/ kunwadee/research/p2p/paper.html.

[3] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble, "Measuring and analyzing the characteristics of Napster and Gnutella hosts," *Multimedia Systems Journal*, vol. 9, no. 2, pp. 170–184, 2003.

[4] 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*, vol. 6, no. 1, pp. 50–57, 2002.

[5] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Multimedia Computing and Networking*, San Jose, CA, Jan. 2002.

[6] Anurag Singla and Christopher Rohrs, "Ultrapeers: Another step towards Gnutella scalability," Gnutella Developer's Forum, Nov. 2002, http://groups.yahoo.com/group/the_gdf.

[7] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker, "Search and replication in unstructured peer-to-peer networks," in *International Conference on Supercomputing*, New York, June 2002.

[8] Adam Fisk, "Gnutella dynamic query protocol v0.1," Gnutella Developer's Forum, May 2003, http://groups.yahoo.com/group/the_gdf.

[9] Lime Wire LLC, "Crawler Compatability," Gnutella Developer's Forum, Jan. 2003.

[10] M. Jovanovic, F.S. Annexstein, and K.A. Berman, "Modeling peer-to-peer network topologies through "small-world" models and power laws," in *TELFOR*, Nov. 2001.

[11] D. J. Watts, "Six Degrees," in *The Essence of a Connected Edge*. 2003, ACM Press.

[12] M. Jovanovic, "Modeling large-scale peer-to-peer networks and a case study of gnutella," M.S. thesis, University of Cincinnati, 2001.

[13] Pradnya Karbhari, Mostafa Ammar, Amogh Dhamdhere, Himanshu Raj, George Riley, and Ellen Zegura, "Bootstrapping in Gnutella: A measurement study," in *Passive & Active Measurement Workshop*, Antibes Juan-les-Pins, France, 2004.

[14] Karl Aberer, Magdalena Punceva, Manfred

Hauswirth, and Roman Schmidt, "Improving data access in P2P systems," *IEEE Internet Computing*, vol. 6, no. 1, pp. 58–67, Jan./Feb. 2002.

[15] Beverly Yang and Hector Garcia-Molina, "Evaluating guess and non-forwarding peer-to-peer search," in *International Conference on Distributed Computing Systems*, 2004.

[16] Beverly Yang and Hector Garcia-Molina, "Designing a Super-Peer Network," in *International Conference on Data Engineering*, Mar. 2003.

[17] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *INFOCOM*, 2003.

[18] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos, "On power-law relationships of the internet topology," in *SIGCOMM*, 1999, pp. 251–262.

[19] Alberto Medina, Ibrahim Matta, and John Byers, "On the origin of power laws in internet topologies," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 2, pp. 18–28, 2000.

[20] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The origin of power laws in internet topologies revisited," in *INFOCOM*, 2001.

[21] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger, "Network topology generators: Degreebased vs. structural," in *SIGCOMM*, 2002.