Mapping the Gnutella Network

A mismatch between Gnutella's overlay network topology and the Internet infrastructure has critical performance implications.

Matei Ripeanu and Adriana Iamnitchi
University of Chicago

Ian Foster
University of Chicago and Argonne National Laboratory

Peer-to-peer systems have emerged as a significant social and technical phenomenon, and they are likely to gain popularity as low-cost individual computing and storage resources become more widely available and network connectivity increases. Unlike traditional distributed systems, P2P networks aggregate large numbers of computers that join and leave the network frequently and that might not have permanent network (IP) addresses. This new breed of systems creates application-level virtual networks with their own routing mechanisms that allow individual computers to share information and resources directly, without dedicated servers.

The topology of these overlay networks and the routing mechanisms used have a significant impact on application properties such as performance, reliability, scalability, and, in some cases, anonymity. The topology also determines the communication costs associated with running the P2P application, both at individual hosts and in the aggregate. Note that the decentralized nature of pure P2P systems means that these are emergent properties, determined by local decisions made by individual resources based only on local information: We are dealing with a self-organized network of independent entities.

To explore these issues, we studied the topology and protocols of the public Gnutella network. Its substantial user base and open architecture made it a good large-scale, if uncontrolled, testbed. We captured the network's topology, generated traffic, and dynamic behavior to determine its connectivity structure and how well (if at all) Gnutella's overlay network topology maps to the physical Internet infrastructure. Our analysis of the network allowed us to evaluate costs and benefits of the P2P approach and to investigate possible improvements that would allow better scaling and increased reliability in Gnutella and similar networks.

Motivations

Recent research shows that networks as diverse as those formed by molecules in cells, people in social groups, and hosts on the Internet follow similar organizational patterns[1,2]: Most nodes have only a few links while a tiny number of hub nodes support numerous links. These networks display an unexpected degree of robustness as communication between nodes is unaffected by extremely high node failure

1089-7801/02/$17.00 ©2002 IEEE

rates, but error tolerance comes at a high price: Power-law networks are vulnerable to attacks, that is, to the selection and removal of a few nodes that provide most of the network's connectivity. Our studies showed that Gnutella preserves the fault tolerance of a pure power-law network, while being less dependent on highly connected nodes that are easy to single out (and attack).

We are not the first to measure Gnutella networks' properties. The Distributed Search Solutions (DSS) group has published raw data from their Gnutella surveys,[3] for example, and the Snowtella project[4] has analyzed the characteristics of participating resources. Others have used this data to study Gnutella users' behavior,[5] to analyze search protocols for power-law networks,[6] and to forecast network growth through simulations.[7] Our network crawling and analysis goes significantly further in terms of sophistication and spatial and temporal scale than these, however, to investigate Gnutella's organizational patterns, network traffic, and efficiency in infrastructure usage.

There are two reasons for analyzing how well the Gnutella overlay network topology maps to the physical Internet infrastructure. First, efficient resource usage ultimately determines the scalability of any P2P application. Second, an Internet service providers' inappropriate overlay topology can add immense stress to the infrastructure — and increase costs for ISPs — if it does not follow the physical infrastructure. This point has been raised elsewhere,[8] but we believe we are the first to quantitatively evaluate the topology mismatch between a P2P application and the Internet.

Gnutella Protocol

Gnutella is an open, decentralized group membership and search protocol,[9] used mainly for file sharing. The term also designates the virtual network of Internet-accessible hosts that run Gnutella-speaking applications (this is the "Gnutella network" we measured) and a number of smaller, and often private, disconnected networks.

Like most P2P file-sharing applications, Gnutella was designed to meet the following goals:

- *Dynamic operability*. P2P applications must keep operating transparently although hosts join and leave the network frequently.
- *Performance and scalability*. P2P succeeds in large-scale deployments that reveal the traditional client-server paradigm's limitations. Scalability is important as P2P applications exhibit what economists call the "network effect" in which a network's value to an individual user scales with the total number of participants.[10] Ideally, when increasing the number of nodes, aggregate storage space and file availability should grow linearly, response time should remain constant, and search throughput should remain high or grow.
- *Reliability*. External attacks should not cause significant data or performance loss.
- *Anonymity*. The application should protect the privacy of people seeking or providing sensitive information.

Gnutella nodes, known as *servents,* perform tasks that are normally associated with both *servers* and *clients*. They provide client-side interfaces through which users can issue queries and view search results, accept queries from other servents, check for matches against their local data sets, and respond with corresponding results. These nodes also manage the background traffic that spreads the information used to maintain network integrity.

To join the system, a new servent connects to an available known host (such as gnutellahosts.com). Once the node establishes one or more open TCP connections with existing network nodes, it can *broadcast* messages to all attached nodes or simply *back-propagate* messages along the path taken by an initial broadcast message. The protocol includes several features that facilitate this mechanism. For example, each message has a randomly generated identifier, and each node keeps a short memory of recently routed messages to prevent rebroadcasting and to implement back-propagation. To help prevent messages being propagated indefinitely, they are flagged with time-to-live (TTL) and "hops passed" fields.

The messages allowed in the network are

- *Group membership*. A node joining the network broadcasts a PING message to announce its presence. Receivers forward the PING to their neighbors and back-propagate a PONG message that contains information such as the receiver's IP address and the number and size of its shared files.
- *Search*. Broadcast QUERY messages contain a user-specified search string that each receiving node matches against locally stored filenames. Back-propagated query responses include information necessary to download a file.
- *File transfer*. Peers download files directly from

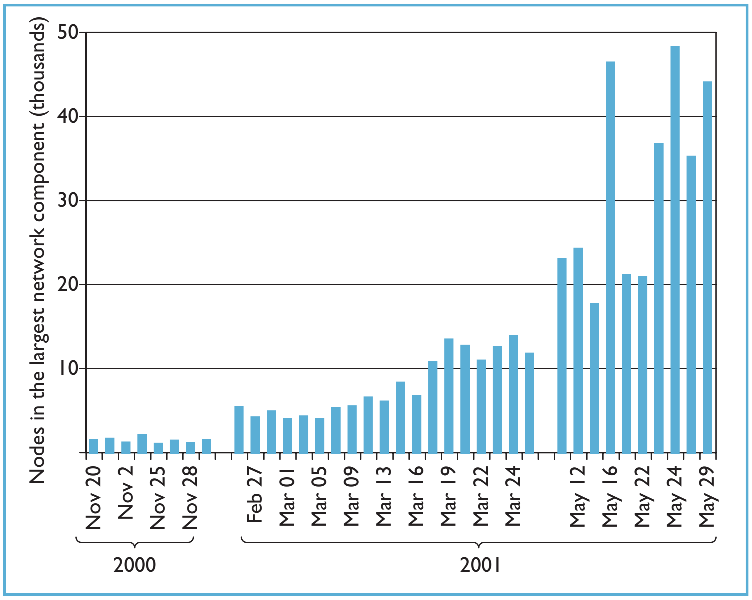

Figure 1. Network growth. The number of nodes in the largest connected component in the network grew by about 25 times over the seven months of our study.

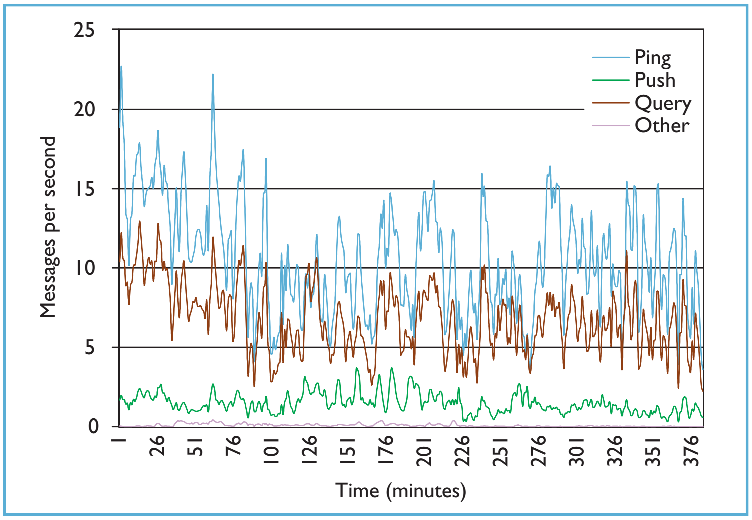

Figure 2. Generated traffic. In November 2000, overhead traffic on a randomly chosen link accounted for more than 50 percent of the total, whereas user query messages were only 36 percent.

each other using GET and PUSH messages.

To cope with Gnutella's dynamic environment, a node periodically PINGs its neighbors to discover other participating nodes. A disconnected node can always use this information to reconnect to the network. Nodes use only local information to decide where to connect and, thus, form a dynamic self-organizing network of independent entities. Gnutella servents are the nodes in this virtual application-level network, and open TCP connections form its links.

Data Collection

We developed a crawler that joins the network as a servent and uses the membership protocol (the PING-PONG mechanism) to collect topology information. The crawler starts with a list of nodes, initiates a TCP connection with each, and sends a generic join-in message (PING). It discovers the neighbors of each contacted node based on the PONG messages it receives in reply and adds them to its list along with each discovered node's IP address, port, number of files, and total space shared. From a short, publicly available list of initial nodes, we built a list of more than 400,000 nodes that have been active at some time.

We first developed a sequential version of the crawler to discover the network. In order to reduce the crawling time, we developed a client-server crawling strategy. The "server" is responsible for managing the list of nodes to be contacted, assembling the final graph, and assigning work to clients. Given this dynamic behavior of the nodes, it is important to find the appropriate tradeoff between discovery time and invasiveness of our crawler. Increasing the number of parallel crawling tasks reduces discovery time but increases the burden on the application. Obviously, the Gnutella graph our crawler produces is not an exact "snapshot" of the network, but we argue that it is close to one, in a statistical sense: All properties of the network (size, diameter, average connectivity, and connectivity distribution) are preserved.

Network Analysis

We use a conservative definition of network membership that excludes nodes that, although reported to be part of the network, our crawler could not connect to because the local servent might have been configured to limit its TCP connections or the node might have left the network before the crawler contacted it. Our study uses data gathered between November 2000 and June 2001. We start with a macroscopic analysis of the network and study its connectivity patterns, we then estimate Gnutella-generated traffic volume, and eventually evaluate the mapping of the Gnutella overlay network to the underlying networking infrastructure.

Growth Trends and Dynamic Behavior

Figure 1 illustrates the Gnutella network's growth during the seven-month period of our study. Although Gnutella's failure to scale has been predicted time and again, the number of nodes in the largest network segment grew from 2,063 hosts in November to 14,949 hosts in March and 48,195 hosts in May. This segment, which included more than 95 percent of the active nodes discovered, grew by about 25 times (admittedly from a low base) during this interval.

Using records of successive crawls, we investigated the dynamic graph structure over time and discovered that about 40 percent of the nodes leave the network in less than 4 hours. In fact, only 25 percent of the nodes remain alive for more than 24 hours.

Estimated Traffic

We used a modified version of the crawler to eavesdrop on the network traffic. Figure 2 classifies, by message type, the traffic that went across one randomly chosen link over a 376-minute period in November 2000. Adjusting for message size, we found that user-generated traffic (QUERY messages) averaged only 36 percent of the total (in bytes). The rest was overhead: 55 percent maintained group membership (PING and PONG messages), and 9 percent contained either nonstandard messages (1 percent) or PUSH messages broadcast by noncompliant servents. Newer Gnutella implementations apparently solved these engineering problems: In June, 91 percent of the generated traffic was query messages, 8 percent PING messages, and insignificant levels of other message types.

Figure 3 shows the distribution of node-to-node shortest path lengths over seven crawls of the network. We found that 95 percent of any two node pairs could exchange messages in fewer than 7 hops. As a result, given Gnutella's flooding-based routing algorithm and message time-to-live predominantly used (TTL = 7), most links support similar traffic (that is, almost all broadcasted messages reach almost all nodes). We verified this theoretical conclusion by measuring the traffic at multiple, randomly chosen nodes and found 6 Kbps per connection on average.

Based on our measurements, we estimated that the total traffic (excluding file transfers) for a large (50,000-node) Gnutella network is 1 Gbit per second: 170,000 connections at 6 Kbps per second per connection, or about 330 Tbytes per month. To put this traffic volume into perspective, we note that it amounts to about 1.7 percent of the total traffic estimated over the U.S. Internet backbone in December 2000.[11] This traffic volume clearly presents an important obstacle to Gnutella's further growth. Efficient use of the underlying network infrastructure is crucial for better scaling and wider deployment.

One interesting feature, as Figure 4 shows, is that the average number of connections per node remained constant as the network scaled up by almost two orders of magnitude. Assuming this invariant holds, we could estimate the traffic for larger networks and calculate scalability limits based on available bandwidth.

Connectivity and Reliability

When analyzing Gnutella's global connectivity and reliability patterns, we must keep in mind the behav-

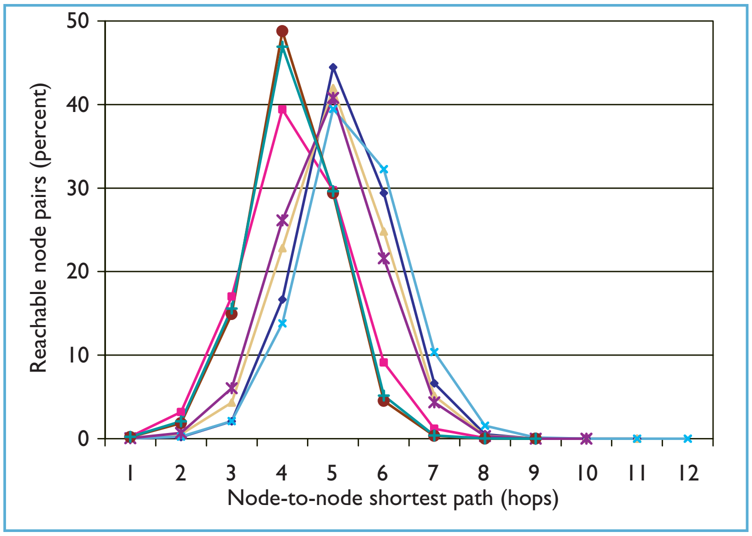

Figure 3. Node-to-node shortest paths. More than 95 percent of node pairs could be reached within 7 hops.

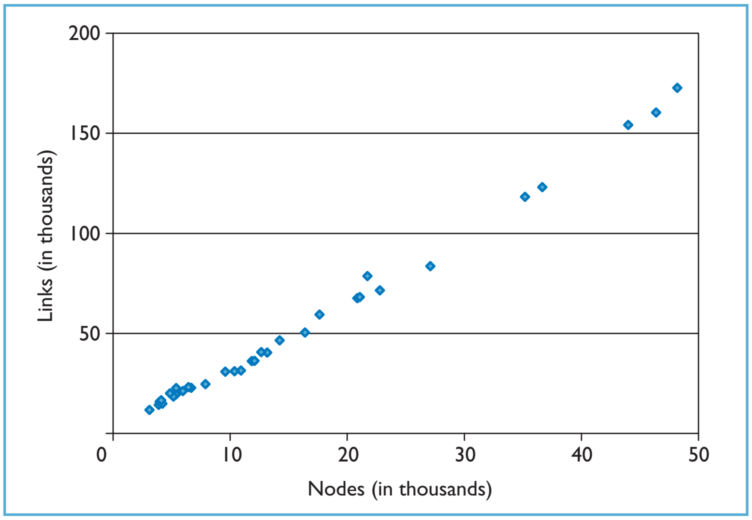

Figure 4. Average node connectivity. As the network grew between November 2000 and May 2001, the average number of connections per node remained constant at about 3.4.

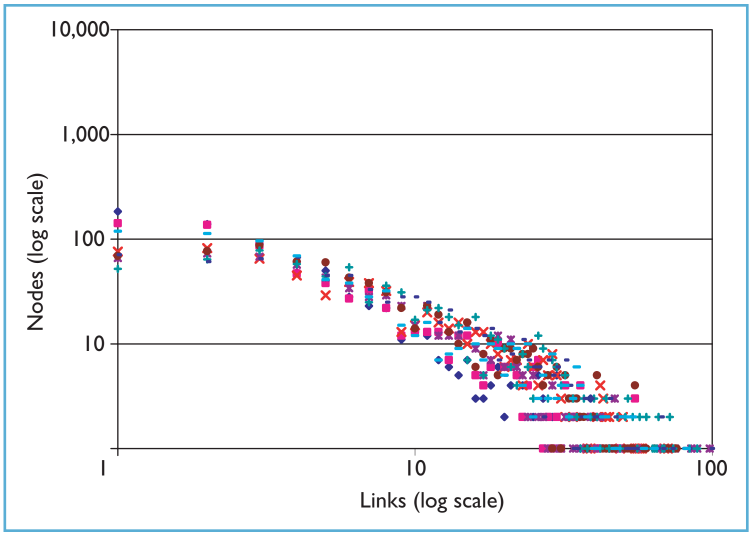

Figure 5. Connectivity distribution (November 2000). The quasi-linear distribution on this log-log plot shows that Gnutella nodes organized themselves into a power-law network.

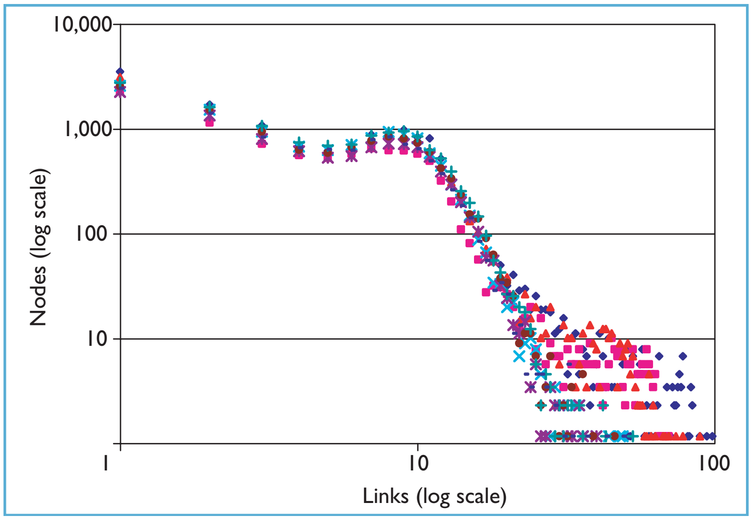

Figure 6. Connectivity distribution (March, May 2001). There are too few nodes with low connectivity to form a pure power-law network.

ior of the self-organizing network: Users decide the maximum number of connections a node should support, but each node decides which others to connect to and when to drop or add connections.

Recent research[1,2] shows that many natural networks such as molecules in a cell, species in an ecosystem, and people in a social group organize themselves as so-called *power-law networks* (more specifically, in a power-law network the number of nodes with L links is proportional to L^{-k} where k is a network-dependent constant). This structure helps explain why these are generally highly stable and resilient, yet prone to occasional catastrophic collapse.[12] Since most nodes (molecules, Internet routers, and Gnutella servents) are sparsely connected, little depends on them: A large fraction can be taken away and the network stays connected. But, if just a few highly connected nodes are eliminated, the whole system could crash. One implication is that these networks are extremely robust when facing random node failures, but vulnerable to well-planned attacks.

Given the diversity of networks that exhibit power-law structure and their properties, we were interested to determine whether Gnutella falls into the same category. Figure 5 presents the connectivity distribution we observed in November 2000. Each series of points represents one network topology. Although data are noisy because of the small size of the networks, we can easily recognize the signature of a power-law distribution.

As Figure 6 illustrates, however, more recent Gnutella networks tend to move away from this organization. There are too few nodes with low connectivity to form a pure power-law network. The power-law distribution is preserved for nodes with more than 10 links, but nodes with fewer links follow an almost constant distribution.

We believe that this new, multimodal distribution has an impact on network reliability. A more uniform connectivity distribution might preserve the network's ability to deal with random node failures and reduce dependence on highly connected nodes.

We speculate that a group of devoted users maintain the small number of Gnutella nodes with the server-like characteristics visible in these power-law distributions. These nodes have a large number of open connections and/or provide much of the content available in the network. Moreover, these server-like nodes have a higher availability: They are about 50 percent more likely than the average to be found alive during two successive crawls.

Overlay Network Topology

P2P computing changes the way we use the Internet because it lets computers at network edges act as both clients and servers. As a result, P2P applications radically change the amount of bandwidth the average Internet user consumes. Most ISPs in the U.S. use flat-rate billing, but if P2P applications become ubiquitous, they could break that business model.[8]

Proper Fit

Given the traffic volume they generate, P2P applications will be scalable only by employing avail-

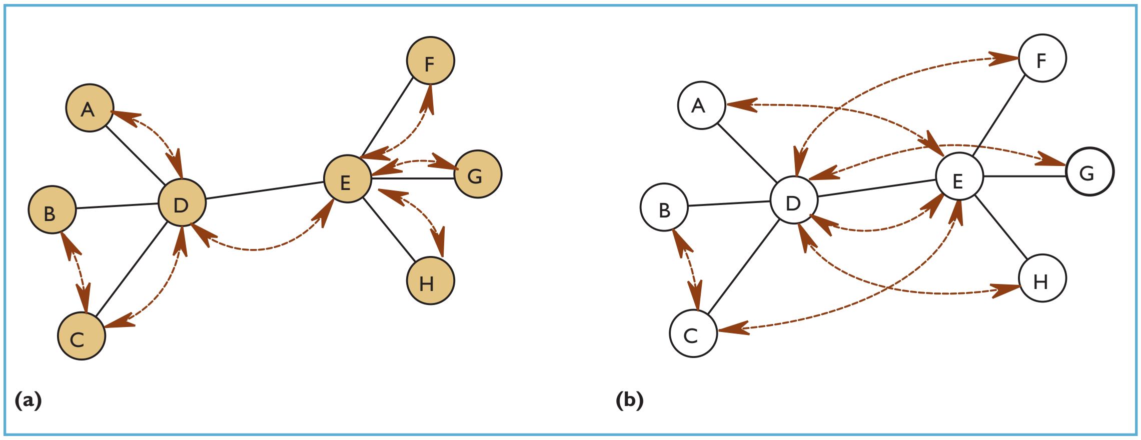

Figure 7. Mapping the overlay network topology to the network infrastructure. (a) With perfect mapping, a message inserted into the network by node A travels physical link D-E only once to reach all other nodes. (b) With inefficient mapping, the same message traverses the link six times.

able networking resources efficiently. Gnutella's store-and-forward architecture makes it extremely important that its overlay network topology map well to the physical network infrastructure.

Figure 7 highlights the importance of a "properly fitting" overlay topology. The black solid lines represent the underlying infrastructure that connects eight hosts in a Gnutella-like network, and red dotted lines denote the application's overlay topology. In Figure 7a, the overlay topology closely matches the infrastructure, and a broadcast from node *A* involves only one communication over the physical link *D*-*E*. In Figure 7b, the same broadcast involves six communications over the same link.

Unfortunately, it is prohibitively expensive to compute the Gnutella network's exact mapping because of the difficulty extracting the Internet topology and the computational scale of the problem. Instead, we performed two high-level experiments that highlighted the mismatch between the topologies.

Mismatch

The Internet is a collection of interconnected autonomous systems (AS), which are groups of local area networks under shared technical administration. From an ISP's viewpoint, traffic that crosses AS borders is more expensive than local traffic. We found that only 2 to 5 percent of Gnutella connections link nodes within a single AS, although more than 40 percent of all Gnutella nodes are located within the top 10 ASes. The fact that most Gnutella-generated traffic crosses AS borders thus increases costs unnecessarily.

In a second experiment, we assumed that the hierarchical organization of domain names mirrors the Internet infrastructure. For example, the communication costs between two hosts in the uchicago.edu domain are likely to be significantly smaller than between hosts in uchicago.edu and stanford.edu. This assumes that domain names express some sort of organizational hierarchy and that organizations tend to build networks that exploit locality within that hierarchy.

We divided the Gnutella overlay topology graph into *clusters* (hubs and their adjacent nodes) to study how well it maps to Internet partitioning as defined by domain names. We used a simple clustering algorithm, based on the connectivity distribution described earlier, and merged clusters that shared more than 25 percent of their nodes. We then assigned nodes that belonged to more than one cluster to only the largest, and formed a final cluster with the leftover nodes.

We defined the entropy[13] of a set *C* that contains $|C|$ hosts, each labeled with one of n distinct domain names, as:

$$E(C)=\sum_{i=1}^{n}\left(-p_i\log(p_i)-(1-p_i)\log(1-p_i)\right),$$

where p_i is the probability of randomly picking a host with domain name i.

We then defined the entropy of a network of size $|C|$, with k clusters of sizes $|C_1|$, $|C_2|$, … , $|C_k|$, as:

$$E(C_1,C_2,\ldots C_k)=\sum_{i=1}^{k}\frac{|C_i|}{|C_1|+|C_2|+\ldots+|C_k|}*E(C_i)$$

We base our reasoning on the property that $E(C)\geq E(C_1,C_2,\ldots,C_k)$no matter how the clusters are chosen. If the clustering matches the domain partitioning, then we should find that $E(C) >> E(C_1, C_2, \ldots, C_k)$. Conversely, if the clustering has the same level of randomness as in the initial set C, the entropy should remain largely unchanged. We essentially used the entropy function to measure how well the two partitions applied to a set of nodes — the first using the information contained in domain names, and the second using the clustering heuristic — match. Note that a large class of data mining and machine learning algorithms based on information gains use a similar argument to build their decision trees[14] (ID3 or C4.5, for instance).

We performed this analysis on 10 topology graphs collected during February-March 2001. Because we detected no significant decrease after performing the clustering (all were within 8 percent of the initial entropy value), we concluded that Gnutella nodes cluster independently of the Internet structure. The self-organizing Gnutella network thus appears to use the underlying physical infrastructure inefficiently.

Potential Improvements

Applying our measurement and analysis techniques to other P2P systems can help developers understand some important design tradeoffs. Clearly, the topology mismatch problem must be solved in order for systems like Gnutella to be widely deployed, but Gnutella also takes few precautions to ward off attacks. Security mechanisms appear essential for the network's long-term survival because the ease with which we obtained network topology information would permit highly efficient denial-of-service attacks.

Another direction for improvement would be to exploit particular distributions of query values and locality in user interests.[15] Various studies show that the distribution of Gnutella queries and Internet-based HTTP requests both follow Zipf's law. (The frequency of the r^{th} most popular request is proportional with its rank. Note that although Zipf's formulation is widely used, these distributions can also be expressed as power-law distributions.) Therefore, proxy cache mechanisms used on the Web might be useful in a P2P context as well. Moreover, a query-caching scheme could bring even greater performance improvements to a dynamic P2P network in which nodes are grouped by user interest.

Replacing the query flooding mechanism with smarter (and less expensive in terms of communication costs) routing and group communication mechanisms would provide another substantial improvement. Several recent P2P schemes, such as Content Addressable Networks,[16] and Tapestry,[17] propose a structured application-level topology that allows semantic query routing. We believe, however, that preserving the power-law characteristics that emerge in Gnutella's ad hoc network topology offers a more promising approach. Mixing dissemination schemes (perhaps based on *epidemic* protocols, which spread data similar to the way a disease spreads) with random query forwarding would provide one way to do this and still decrease network bandwidth consumption. We plan to use the data we have collected on Gnutella's operating environment in simulation studies to locate protocol alternatives that could be used in various networks. The social circumstances that have fostered Gnutella's success could change and the network might decline, but P2P is unlikely to go away.

Acknowledgments

We are grateful to Larry Lidz, Conor McGrath, Dustin Mitchell, Yugo Nakai, Alain Roy, and Xuehai Zhang for their insightful comments and generous support. This research was supported in part by the U.S. National Science Foundation under contract ITR-0086044.

References

1. M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (SIGCOMM),* ACM Press, New York, 1999, pp. 251-262.
2. A. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5489, 15 Oct. 1999, pp. 509-512.
3. Distributed Search Solutions Group, "Gnutella: To the Bandwidth Barrier and Beyond," http://www.clip2.com, Nov. 2000.
4. S. Saroiu, P. Gummadi, and S.D. Gribble, *A Measurement Study of Peer-to-Peer File Sharing Systems,* tech. report UW-CSE-01-06-02, Dept. of Computer Science and Eng., Univ. of Washington, Seattle, July 2001.
5. E. Adar and B. Huberman, "Free Riding on Gnutella," *First Monday*, vol. 5, no. 10, 2 Oct. 2000; http://www.firstmonday.dk.
6. L. Adamic et al., "Search in Power-Law Networks," *Physical Rev. E*, vol. 64, no. 4, Oct. 2001, article no. 046135.
7. T. Hong, "Performance," *Peer-to-Peer: Harnessing the Power of Disruptive Technologies,* A. Oram, ed., O'Reilly and Associates, Cambridge, Mass., 2001.
8. T. Spangler, "The Hidden Cost of P2P," *Interactive Week*, vol. 8, no. 8, 26 Feb. 2001.
9. Gnutella Protocol Specification, version 0.4; available at

http://www.clip2.com/GnutellaProtocol04.pdf.

10. M. Katz and C. Shapiro, “Systems Competition and Network Effects,” *J. Economic Perspectives,* vol. 8, no. 2, 1994, pp. 93-115.
11. K. Coffman and A. Odlyzko, “Internet Growth: Is There a ‘Moore’s Law’ for Data Traffic?” *Handbook of Massive Data Sets*, J. Abello et al., ed., Kluwer, Dordrecht, Netherlands, 2001.
12. R. Albert, H. Jeong, and A. Barabási, “Error and Attack Tolerance of Complex Networks,” *Nature,* vol. 406, 27 July 2000, pp. 378-382.
13. T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, 1991.
14. J. Han and M. Kamber, *Data Mining: Concepts and Techniques,* Morgan Kaufmann, San Francisco, 2000.
15. K. Sripanidkulchai, “The Popularity of Gnutella Queries and its Implications on Scalability,” white paper, Carnegie Mellon Univ, Pittsburgh, Feb. 2001.
16. S. Ratnasamy et al., “A Scalable Content-Addressable Network,” *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (SIGCOMM),* ACM Press, New York, 2001, pp. 161-172.
17. S. Rhea et al., “Maintenance-Free Global Data Storage,” *IEEE Internet Computing*, vol 5, no. 5, Sept./Oct. 2001, pp. 40-49.

Matei Ripeanu is a PhD candidate in computer science at the University of Chicago. His research interests are in the areas of distributed computing, performance modeling, self-organized, distributed applications, and network protocols.

Adriana Iamnitchi is a PhD candidate at The University of Chicago. Her research interests are in distributed computing, focusing on highly scalable, decentralized middleware services and applications for Grid environments.

Ian Foster is senior scientist and associate director of the Mathematics and Computer Science Division at Argonne National Laboratory, professor of computer science at the University of Chicago, and senior fellow in the Argonne/University of Chicago Computation Institute. He has published four books and more than 100 papers and technical reports in parallel and distributed processing, software engineering, and computational science. Foster cofounded the Global Grid Forum and recently coedited a book on the topic, *The Grid: Blueprint for a New Computing Infrastructure* (Morgan-Kaufmann, 1998).

Readers can contact the authors at {matei, foster, anda} @cs.uchicago.edu.