Deconstructing the Kazaa Network

Nathaniel Leibowitz
Expand Networks
natan@expand.com

Matei Ripeanu
Computer Science Department,
The University of Chicago
matei@cs.uchicago.edu

Adam Wierzbicki
Polish-Japanese Institute for
Information Technology
adamw@icm.edu.pl

Abstract

Internet traffic is experiencing a shift from web traffic to file swapping traffic. Today a significant part of Internet traffic is generated by peer-to-peer applications, mostly by the popular Kazaa application. Yet, to date, few studies analyze Kazaa traffic, thus leaving the bulk of Internet traffic in dark. We present a large-scale investigation of Kazaa traffic based on logs collected at a large Israeli ISP, which capture roughly a quarter of all traffic between Israel and US.

1. Introduction

In a brief period of time the composition of Internet traffic shifted dramatically from mainly Web traffic to traffic generated by peer-to-peer (P2P) file-sharing applications like Kazaa, Morpheus or iMesh. Both network measurements and anecdotic evidence support this statement. Internet2 administrators report that about 16% of the traffic carried by their network is P2P traffic while a further 54% is unidentified traffic most likely to be generated by applications in the same class [1] (January 2003). Between 15% and 30% of residential subscribers on several large ISPs surveyed were using Kazaa or Morpheus [2]. Downloads of P2P applications progress at incredible rates: 3.2 million per week for Kazaa and 200,000 per week for Gnutella [3] (February 2003).

Yet, to date, few studies have analyzed Kazaa traffic, thus leaving the bulk of Internet traffic in dark. This paper describes a large-scale investigation of this traffic structured along three main guidelines:

- Firstly, *we try to identify the salient features of Kazaa traffic*. We confirm that the traffic is highly concentrated around a small minority of large, popular items. We find however, that this concentration is even more pronounced than previously reported. This is a strong indication that caching can bring significant savings in this context.
- Secondly, *we study the dynamics of network content* to better understand both the underlying trends in the user community and in its tastes, and the potential for caching. We are interested in the rate of appearance of new content, as well as in the stability properties for the sets of the most popular items.
- Thirdly, *we study the virtual relationships that form among users* based on the data they download. We model the network as a *data-sharing graph* and uncover its small-world characteristics. We believe that these small-world characteristics can be exploited to build efficient data location and data delivery mechanisms.

The rest of this paper is structured as follows. In the next section we describe our data collection setup and the main trace processing steps. Section 3 surveys related work on peer-to-peer traffic characterization. Section 4 comprises the bulk of our analysis structured along the three guidelines mentioned above. We summarize our findings in Section 5.

2. Data Collection and Processing

Few details are publicly available about the Kazaa protocol. Apparently, Kazaa nodes dynamically elect 'super-nodes' that form an unstructured overlay network and use query flooding to locate content. Regular nodes connect to one or more super-nodes to query the network content and in fact act as querying clients to super-nodes. Once desired content has been located Kazaa uses HTTP protocol to transfer it directly between the content provider and the node that issues the download request. In fact, to improve transfer speed multiple file fragments are downloaded in parallel from multiple providers. While traffic flowing on the control channel (queries, network membership information, software version information, etc.) is encrypted, traffic on the download channel (i.e. all HTPP transfers) is not encrypted. We are using information collected from the download channel for an, admittedly non-exhaustive, study of Kazaa network.

2.1. Data Collection

To collect Kazaa traces we use a setup similar to [4]. We briefly describe the trace collection setup below and refer to [4] for a complete description. A caching server is installed at the border between the local user base of a large ISP and the Internet cloud. For each TCP connection, regardless if direction (both in and out), a Layer 4 switch inspects the fist few packets to detect Kazaa download traffic. If download traffic is detected then the switch redirects it through the caching server. Thus, the caching server is able to intercept all downloads performed by local users, cache, and serve cached data. We note that we focus on downloads performed by local users and completely ignore downloads performed by outside users from local file providers (in other words we are only interested in incoming traffic).

Additionally, our content-based technique to detect traffic of interest has significant advantages over traditional, port based techniques: we find that in February 2003 about 38% of all download sessions were not using the standard Kazaa port (1214).

It is difficult to define Kazaa downloads in the terms originally coined for describing standard file downloads, the salient difference being that the download of a single file is usually composed of tens of smaller downloads of different fragments of the file from different providers. This complicates both the terminology and the computations involved in analyzing the data. We use the terms and methods introduced in [4] to circumvent these problems: The term *download* or *session* describes a single TCP session between two nodes, over which a portion of a file (none, part, or all of the file) is transferred. The term *download cycle* describes the logical transfer of a whole file, which may consist of tens of sessions, and extend over hours or even days. Finally, we use the following scheme to quantify the number of download cycles for each file: if an accumulated value of X bytes of file Y have been transferred over the network, then we estimate that ${}^{X}/_{FileSize\text{-}of\text{-}Y}$ download cycles of the file have passed over the network.

2.2. The Traces

The caching server has been continuously logging traffic over the past year. As we do not see qualitative changes in traffic characteristics during this period we use only a part of these logs for most of our analysis below.

We eliminate from our logs all control channel connections and use only the inbound download sessions (i.e. data flowing to local users from both other external or other internal users) for our analysis. Table 1 summarizes the main characteristics of the traffic captured.

Table 1: Characteristics of collected Kazaa traces.

Data collection period	1/15 – 2/15/03
Number of download sessions	$7 * 10^6$
Number of control sessions	$24 * 10^6$
Bytes transferred	20 TB
Concurrent sessions (avg.)	1200
Concurrent sessions (peak)	3000
Bandwidth used (average)	75 Mbps
Bandwidth used (peak)	145 Mbps
Number of unique files	~300,000
Number of unique users	≥ 50,000

3. Related Work

A number of recent studies cast more light on the nature of P2P traffic in particular on traffic generated by FastTrack (KaZaa, KaZaa Lite) and Gnutella (Morpheus, LimeWire, etc) family applications that have started to generate a significant share of Internet traffic.

Sen et al. [5] use TCP flow-level data gathered from multiple routers across a large Tier-1 ISP to analyze three P2P applications (Kazaa, Gnutella and DirectConnect). While this data does not reveal application level details and cannot give insights explaining the behavior observed, it is an important step in characterizing these applications from a network engineering perspective. For example, Sen et al. [5] report that although the distribution of generated P2P traffic volume is highly skewed at the individual host level, the fraction of the traffic contributed by each network prefix remains relatively unchanged over long time intervals.

At the application level, Gnutella's open protocol has made the analysis of this network somewhat simpler. A number of studies [5-9], based mostly on data collected from the control channel, explore the topology of the Gnutella overlay, its mapping on the Internet physical infrastructure, the behavior of Gnutella users, and the main characteristics of Gnutella nodes.

Two recent studies [4, 10] use the fact that although Kazaa's protocol (FastTrack) is proprietary, Kazaa uses HTTP to move data files: thus this traffic can be logged and cached. Both these studies monitor HTTP traffic on costly links: traffic from a large Israeli ISP to US and Europe [4], or from University of Washington campus to its ISP [10] (in the following we refer to these traces as UW traces).

Leibowitz et al. [4] note that Kazaa traffic constitutes most of the Internet traffic, show that a tiny number of files generates most of the download activity, suggest the feasibility of traffic caching, and empirically demonstrate its benefits. Saroiu et al. [10] reaffirm these findings and compare Kazaa traffic with traffic generated by traditional content distribution systems (i.e. Akamai and Web traffic).

We believe the traces analyzed in this study and in [4] complement well the traces analyzed in [10]: our traces reflect a more diverse user population with significant heterogeneity in network connectivity and interests. Additionally we analyze a community where users pay network usage charges upfront, as opposed to a university where users charged for network usage indirectly or not at all.

One interesting observation highlights the difference in user population (and respective usage patterns) in these two studies: while the University of Washington user community acts as net provider of Kazaa content (Saroiu et al. report that at UW outbound Kazaa traffic is at least seven times larger than inbound traffic) the user population in our study is a net content consumer (the ratio outbound to inbound traffic is almost reversed).

In this paper we expand results presented in [4], and, while we note that the basic characteristics remain valid on current traffic, we investigate new aspects of the traffic, user behavior and network structure that have not been previously explored.

4. Analysis

4.1. Counting Downloads

Kazaa's user interface reports hundreds of millions of files available in the network. We cannot confirm or refute this claim, as this would require a global view on the entire network. We analyze, however, the traffic generated by local users downloading files from the rest of the Internet. In this section we analyze one month long Kazaa trace presented above. Since files are generally downloaded from multiple sources we process the logs to compute the number of download cycles for each file. We then produce a list of files sorted by the number of download cycles and used it to generate a CDF (Cumulative Distribution Function) that shows the percent of downloads cycles for each progressing subset of the most downloaded files.

In Figure 1 we observe that only about a half of the requested 300,000 files have been downloaded a significant number of times. Also, 65% of all download cycles go to the 20% most popular files (60,000 files). To provide more detail, Figure 2 zooms-in and plots the CDF for the 10% most popular files: it becomes obvious that about 30% of all download cycles go to the 1% most popular files.

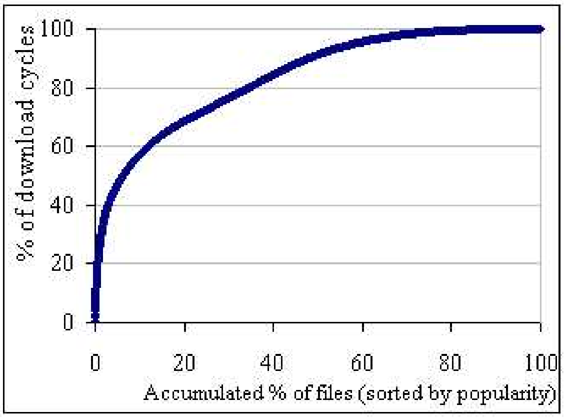

Figure 1: CDF for file download cycles. Note that more than 10% of all download sessions attempted actually fail.

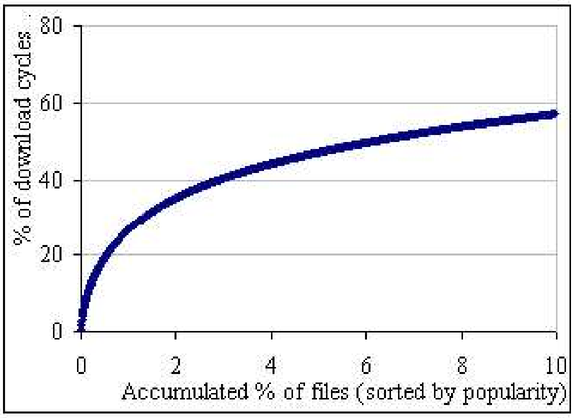

Figure 2: CDF of file download cycles for the 10% most downloaded files. Note that 30% of all download cycles are generated by the 1% most popular files.

4.2. File Download Distribution by Bytes

The analysis above treats each download cycle as a unit value, and ignores file size variability. As a consequence, it does not indicate how much traffic is concentrated around the subset of the most popular files. To investigate this aspect, we weigh each download cycle with the corresponding file size, and obtain for each file, the total amount of traffic that it generated. We then produce a list of files sorted by volume of generated traffic and create a CDF similar to that presented in the previous section: we plot the percent of traffic for each progressing subset of the most popular files. Figures 3 and 4 plot this CDF for byte popularity distribution for the top 10% and respectively 1% most popular files.

The behavior we notice in the previous graph is much more pronounced: we observe that as little as 2500 files (a mere 0.8% of all detected files) account for as much as 80% of the traffic.

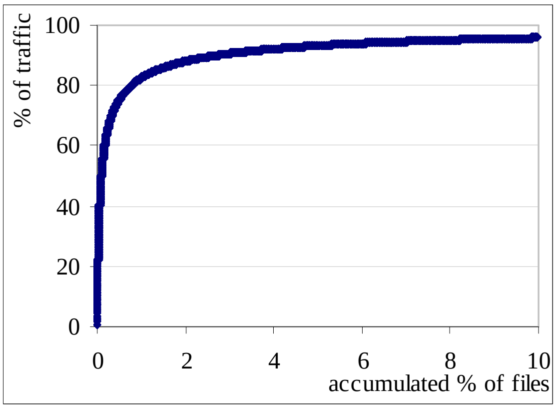

Figure 3: CDF for byte popularity distribution for the 10% most popular files. Note that most of the generated traffic is concentrated around a very small set of files (1% of all files).

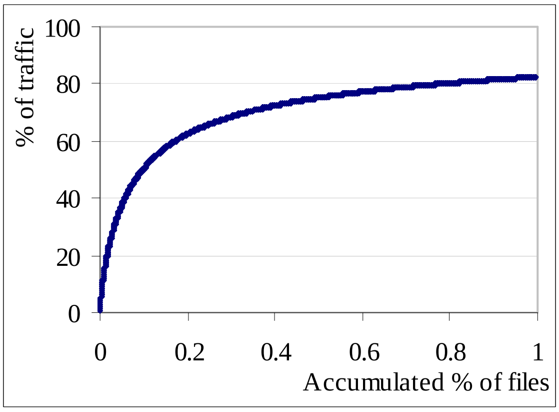

Figure 4: CDF for byte popularity distribution for the top 1% most popular files. Note that 50% of total traffic is generated by the 0.1% most 'bandwidth-hungry' files.

We note that our measurements show a byte popularity distribution significantly more skewed than UW traces [10]. While in the UW traces the most popular 1% of all files account for 'only' about 50% of all bytes transferred, here the same 1% most popular files account for more than 80% of all traffic. To provide better insight, Figure 4 zooms-in and plots the CDF for the 1% most popular files: it becomes obvious that generated traffic is concentrated around the most popular files: as little as 0.1% of the most bandwidth-hungry files generate 50% of the traffic.

4.3. File Sizes

We now switch gears and analyze file size distribution. Figure 5 presents a CDF for file sizes. The 'steep' regions of the plot reflect ranges with a large number of files. As we expect these are: roughly 100KB for pictures, 2-5MB for music files, 50-150MB for applications and movie clips, and larger than 100MB for movies files.

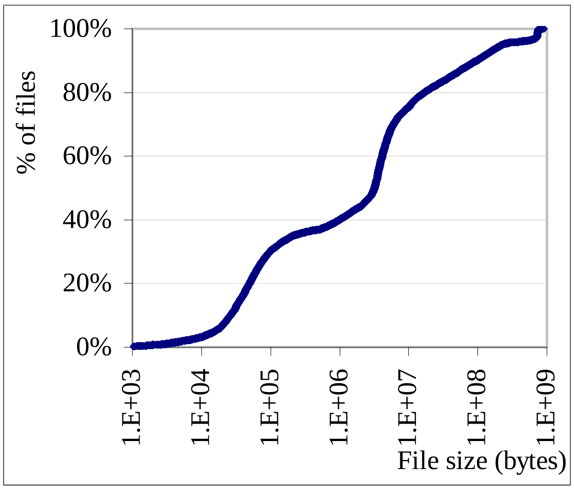

Figure 5: File size cumulative distribution function. The 'steep' portions of the distribution reflect ranges with a larger number of files: 2-5MB for music files, around 100KB for pictures and larger than 700MB for movies probably. (Note the logarithmic scale on X axis in this figure and the normal scale in Figure 6).

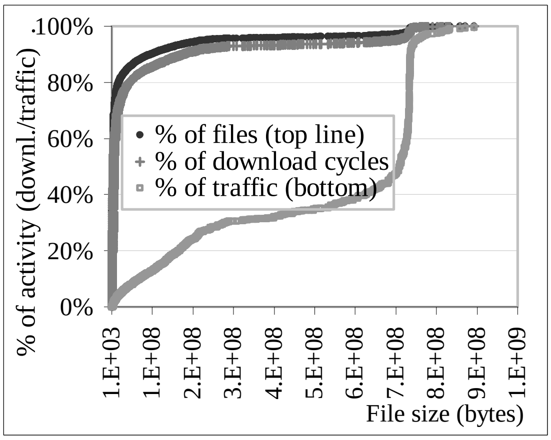

Figure 6: Activity CDF. The plot line in the middle presents the CDF for number of download cycles while the bottom plot presents the CDF for the generated traffic. The top plot representing size CDF is present for reference. The same regions visible in Figure 5 are present (although less accentuated since we use a normal scale on X axis). Note again that files in the 700-900MB range generate most of the traffic.

Additionally, similar to the analysis in the previous section, we are interested in two other aspects: the number of download cycles, and the generated traffic volume. In Figure 6, we weigh each file size by the number of download cycles, and, respectively, by the traffic generated to download the file. We sort files in increasing order of their sizes and plot the usage CDF (where usage is defined as number of download cycles or bytes transferred respectively). The file size CDF plotted in Figure 5 is presented for reference (the top line in the plot).

While the plots have similar structure, the plot representing the CDF of generated traffic weighted by bits transferred, has more pronounced features. It emphasizes the fact that most of the traffic is generated by the largest files (60% of the traffic is generated by file larger than 700MB). It is interesting to note that little traffic is generated by files in the 200-700 MB range, indicated by the plateau in that range – indeed user experience indicates that most files are either smaller than 150 MB (clips and applications) or larger than 700MB (movies and games).

Figure 7 uncovers the roughly linear correlation between a file size and the activity generated in terms of bytes transferred

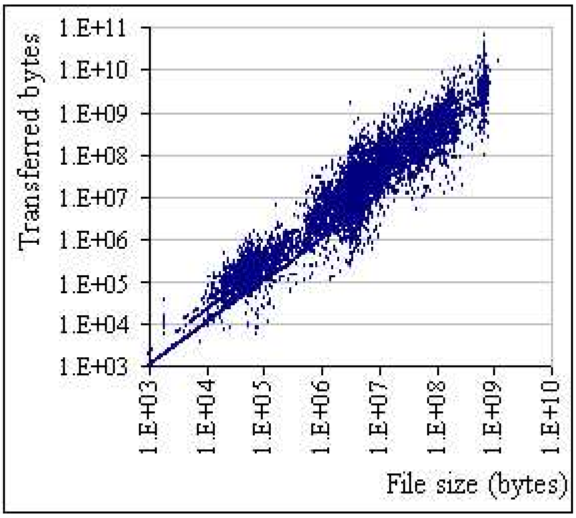

Figure 7: Roughly linear correlation between the file size and the traffic generated by downloading each file (logarithmic scales on both X and Y axes).

4.4. Dynamic Properties of Network Content

4.4.1 Quantity and Rate of Distinct Files.

Kazaa claims its users share millions of files. However it is unclear how many of these files are distinct, or how many are actually transferred over the network, and at what rate. These questions are important for understanding the diversity of the network content, the heterogeneity in user interests, and are crucial from a caching perspective.

The data we use in this section are a detailed log of all Kazaa traffic through our server during a 17 days period (early February 2003). They consist of approximately 3 million downloads which altogether accessed some 150,000 distinct files.

We process these logs in three different time units: minute, hour and day. Our strategy for answering the above questions is to compute the number of distinct *new* files observed during each time unit (*new* files are files not observed from the beginning of the experiment). The first time units should measure high values of new files, and later new files will be encountered less frequently.

Figure 8 plots the number of distinct new files observed in consecutive one hour periods. Initially the rate at which new files are encountered is extremely high and then declines sharply after a few hours, indicating a large temporal locality: once a file is requested it will be requested again soon. The seasonal pattern observed on Figure 8 follows a period of 24 hours with night-time peaks. This seasonality is easily explained, since the majority of our users are in the same time zone.

In order to evaluate the rate of change, we show in Figure 9 a close-up for the first hour, computed at a 1-minute resolution. Initially we encounter 200 new distinct files a minute (a new file every 0.3 seconds). This value declines sharply attains a relative stability within 20 seconds at a value of 50 new filer per minute. This stability, however, is superficial, as evident by the constant slope at the hourly resolution plotted in Figure 8.

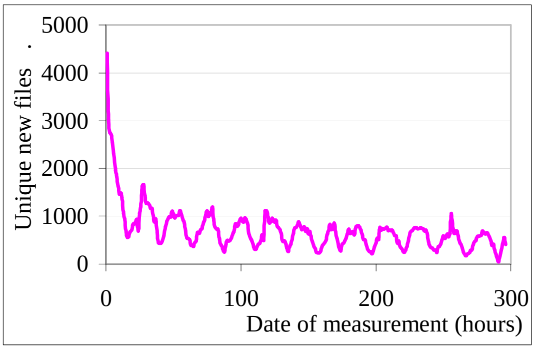

Figure 8: New files encountered during one hour long intervals for our 17-day trace.

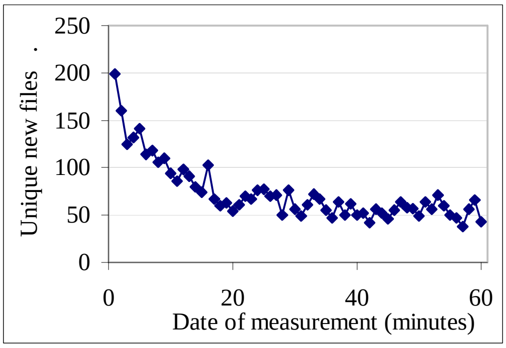

Figure 9: New unique files by minute for the first hour in our trace

In order to better understand the behavior and enable extrapolation, in Figure 10 we plot the same values at a 1-day resolution, which avoids the daily cycle. The persistent decrease in the rate of encountering new files, even after 16 days is clearly visible.

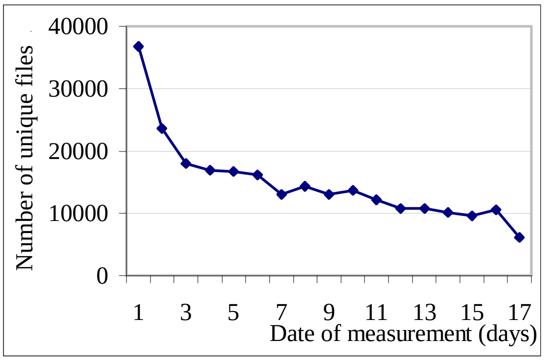

Figure 10: New files encountered during a one day interval for our 17-day trace.

During the period of observation, the number of new unique files did not decrease to zero and did not stabilize at a constant level. However, it is reasonable to suppose that this value would stabilize during a longer observation period. We suggest an interesting explanation for the steady state value: it indicates the rate at which new files enter the network, in other words the rate at which new songs, movies games and the like are created.

4.4.2 Rate of Change

An interesting question, both from a caching perspective and from the perspective of understanding usage patterns, is the rate variation for the set most popular files. Consider for instance compiling every day the list of 100 most popular files. How would these lists change over time? Would it be possible to identify files that are always on these lists (all time favorites), or would the list change very quickly (equivalent of one-day stars)?

To investigate this question, we determine the N most popular files during consecutive observation periods, where $N \in \{4, 50, 400\}$. The observation periods are approximately 24 hour intervals. The popularity of a file is measured by the number of download cycles of the file.

The first part of our analysis investigates how much the lists of most popular N files change from one observation period to another. Let x_t be the set of files that were on both the N-most-popular-list of observation period t-1 and t. We calculate $100 * x_t / N$ to obtain a percentage of the popular files that have persisted between the two observations. This value is plotted on Figure 11 for different values of N.

For N=4, the percentage of recurrently popular files is almost always *50%*, which means that during all the observation periods *2* files persistently occupied the top 4 lists. Based on accumulated user experience with the Kazaa application, we assume these files are most likely the Kazaa software installation packages, which circulate frequently in the network. For higher values of N, the situation changes. The percentage of recurrently popular files seems to be stable at about *30%*, slightly decreasing for large N. This suggests that caching could be quite effective for Kazaa traffic.

We investigate the characteristics of these files that remain popular from one observation period to the following. For each new observation period, we intersect the list corresponding to that period with the intersection of the lists from all previous observation periods. In Figure 12, we plot the percentage of the files in the first list that remained in this intersection after t observation periods. The percentage of files that are popular in all observation periods stabilizes at about *15%*. This suggests that there are indeed a number of "all-time favorites" items during our observation.

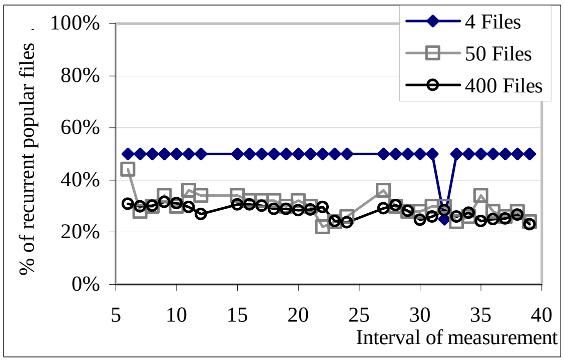

Figure 11: Ratio of the popular files set that remains stable during consecutive time periods.

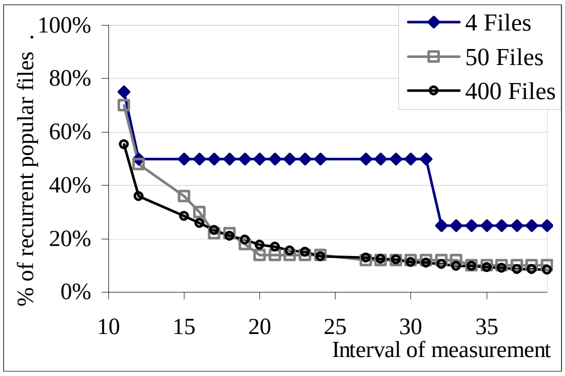

Figure 12: Ratio of the popular files set that remains stable when compared with a base period.

The number of files that remain popular in the following observation period is larger than the number of files that are popular in all observation periods. This suggests that the set of cacheable files changes over time, since only about half of these files are present in all observation periods. A longer experimentation period is required to determine how persistent is this group and further quantify their rate of change over months.

Summarizing these two experiments, we obtain that 15% of the highly popular files, remain popular throughout the experiment, while the rest are popular shorter time intervals. This indicates that the popular files are composed of two sets: a set of persistently popular files and a set of transiently popular files whose popularity is short lived.

4.5. Data-Sharing Relationships among Users

This section explores the virtual relationships that form among Kazaa users based on the files they try to download. We are inspired by recent studies [11, 12] that analyze the Web and a high-energy physics collaboration and uncover in both these systems small-world patterns emerging in users' data-sharing relationships.

When users install and configure Kazaa application they have the opportunity to choose a *user name.* Our traces capture user names and we use them to identify users. We investigate the distribution of download activity (generated traffic and number of download sessions) over the set of user names and discover that three users generate one order of magnitude or more activity than any other users in our set (about 20% of all system activity is generated by these three users; Figure 13). We believe these are 'outliers': in fact multiple users that have not configured their software and thus run under the default user name (in fact their usernames: *defaultuser* or *kazaliteuser* strengthen this intuition). Therefore, for the analysis in this section, we purged out all activity generated under these user names.

We follow closely the technique described in [11]. We define the *data-sharing* graph as the graph whose nodes are the Kazaa users; edges connect pairs of nodes whose activity satisfies a similarity criterion: two users are connected if they (try to) download at least *m* common files during a time interval *T*. For this analysis we use a one week long Kazaa trace and we vary *m* from 1 to 5 and T from 4 to 48 hours.

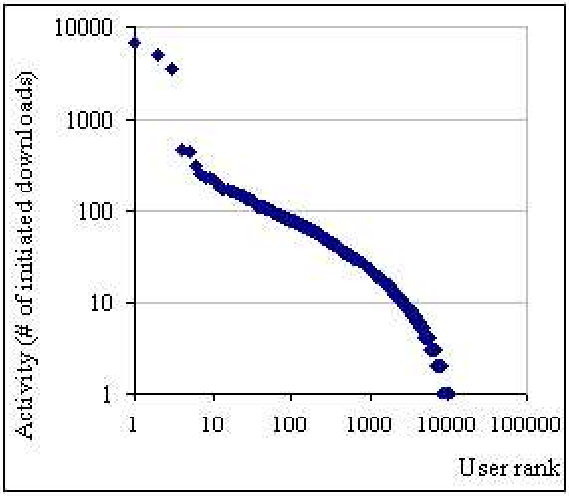

Figure 13: Activity distribution over the user name space. Users are ordered in decreasing order of the number of downloads they initiate. (logarithmic scales on both X and Y axes).

We discover that *data-sharing graph displays small-world properties*. Small-world graphs are defined by comparison with random graphs with the same number of nodes and edges: first, a small-world displays a small average path length, similar to a random graph; second, a small-world has a significantly larger clustering coefficient than a random graph of the same size. The clustering coefficient captures how many of a node's neighbors are connected to each other. One can picture a small-world as a graph constructed by loosely connecting a set of almost complete sub-graphs. Social networks, in which nodes are people and edges are relationships; the Web, in which nodes are pages and edges are hyperlinks; and neural networks, in which nodes are neurons and edges are synapses or gap junctions, are a few of the many examples of small-world networks [13-16].

Table 2 presents the average path-length and the clustering coefficient (averaged over multiple intervals of equal duration) for the data-sharing graphs defined by a few different similarity criteria. We compare these metrics with those of random graphs of similar sizes. Note that despite diversity graph definitions (i.e., similarity criteria), and graph sizes, the values are remarkably close.

Table 2: Clustering coefficient and average path length for graphs constructed under various similarity criteria (together with the same characteristics for random graphs of similar size).

Similarity criteria used	Graph size (avg.)		Average path length (avg.)		Clustering coefficient	
	# nodes	# links	DS graph	Rand. graph	DS graph	Rand graph
m=1, *T*=4h.	1585	8546	4.01	4.41	0.653	0.0070
m=1,*T*=8h.	2038	14267	3.76	4.08	0.645	0.0068
m=1, *T*=12h	3033	29991	3.31	3.50	0.605	0.0065
m=2, *T*=24h	1311	5227	3.72	4.51	0.483	0.0040
m=3, *T*=48h	914	2200	3.93	7.76	0.410	0.0052

Figure 14 compares these data-sharing graphs with a selection of well-known, small-world graphs, including citations network, power grid, movie actors, Internet, Web [16]. Axes represent ratios between the metrics of interest of these graphs and random graphs of the same size. As above, for our data sharing graphs, each point in the plot represents averages for all graphs constructed using one similarity criterion.

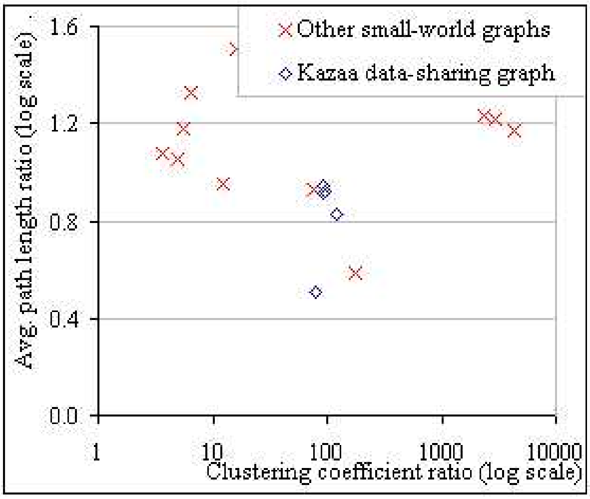

Figure 14: Comparing Kazaa's data-sharing graphs with a selection of well-known, small-world graphs, including citations network, power grid, movie actors, Internet, Web.

5. Summary

We present a study of current (early 2003) Kazaa traffic, which has been dominating the Internet traffic for the past two years. We confirm previous findings that Kazaa traffic is highly concentrated around a small minority of large, popular items. We find however, that this concentration is even more pronounced than previously reported. This is a strong indication that caching can bring significant savings in this context.

We study the dynamics of network content to better understand both the dynamics of the user community and of its tastes, and the potential for caching. We are interested in the rate of apparition of new content, as well as in the stability properties of sets of the most popular items. Based on detailed logs of several weeks of Kazaa traffic, we measure the rate at which new files are encountered in the Kazaa network, and use it to estimate the rate at which new files are created and entered into Internet circulation. We also discover that the set of popular files is composed of two subsets: a small number of files are constantly popular while the rest lose their popularity within days. We note that a longer experimentation period and further analysis are required to quantify these conclusions.

Additionally, based on the intuition of virtual relationships between users that employ similar subsets of data, we model the network as a data-sharing graph and uncover its small world characteristics. We believe that the small-world characteristics of the data-sharing graph can be exploited to build efficient data-location and data-delivery mechanisms

6. Acknowledgements

The authors are grateful to Ian Foster, Adriana Iamnitchi and Anne Rogers for insightful discussions generous support.

This work was supported in part by the National Science Foundation under contract ITR-0086044 (GriPhyN).

7. References

[1] Internet2, "http://netflow.internet2.edu/weekly/," 2003.

[2] InformaticsOnlineWebSite, "http://www.infomaticsonline.co.uk/News/1134977," 2002.

[3] Download.Com_Site, "http://download.com.com," 2003.

[4] N. Leibowitz, A. Bergman, R. Ben-Shaul, and A. Shavit, "Are File Swapping Networks Cacheable? Characterizing P2P Traffic," presented at 7th International Workshop on Web Content Caching and Distribution (WCW'03), Boulder, CO, 2002.

[5] S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks," presented at Internet Measurement Workshop (IMW 2002), Marseille, France, 2002.

[6] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," *Internet Computing Journal*, vol. 6, 2002.

[7] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," presented at Multimedia Computing and

Networking Conference (MMCN), San Jose, CA, USA, 2002.
[8] E. Adar and B. A. Huberman, "Free Riding on Gnutella," *First Monday*, 2000.
[9] F. S. Annexstein, K. A. Berman, and M. A. Jovanovic, "Latency effects on reachability in large-scale peer-to-peer networks," presented at 13th ACM Symposium on Parallel Algorithms and Architectures, Crete, Greece, 2001.
[10] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An Analysis of Internet Content Delivery Systems," presented at 5th Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, 2002.
[11] A. Iamnitchi, M. Ripeanu, and I. Foster, "Data-Sharing Relationships in the Web," presented at The Twelfth International World Wide Web Conference (WWW-12), Budapest, Hungary, 2003.
[12] A. Iamnitchi, M. Ripeanu, and I. Foster, "Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations," presented at 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 2002.
[13] D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness*: Princeton University Press, 1999.
[14] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph," presented at ACM New Security Paradigms Workshop, 2002.
[15] L. Adamic, "The Smal World Web," presented at European Conference on Digital Libraries, Paris, France, 1999.
[16] R. Albert and A. L. Barabasi, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, vol. 74, pp. 47--97, 2002.