

Deconstructing the Kazaa Network

Nathaniel Leibowitz
Expand Networks
natan@expand.com

Matei Ripeanu
The University of Chicago
matei@cs.uchicago.edu

Adam Wierzbicki
Warsaw University of Technology
adamw@icm.edu.pl

Abstract

Internet traffic is experiencing a shift from web traffic to file swapping traffic. Today about half 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 file-sharing applications like Kazaa, Morpheus or iMesh. Both network measurements and anecdotic evidence support this statement. Internet2 administrators report that about 20% of the traffic carried by this network is P2P traffic with a further 50% unidentified traffic most likely to be generated by applications in the same class [1]. Between 15% and 30% of residential subscribers on several 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].

Yet, to date, few studies analyze Kazaa traffic, thus leaving the bulk of Internet traffic in dark. This paper aims for a large-scale investigation of this traffic that makes up the majority of Internet flow. Our investigation is 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 very large, very 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 changes in user community and in 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 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

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 server is installed at the border between the local user base of a large ISP and the Internet cloud. Based on destination port number for each TCP connection a Layer 4 switch redirects all Kazaa traffic to this server. Thus, the server is able to intercept all downloads performed by local users from the external Internet. We note that in the data we analyze 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).

It is difficult to define Kazaa downloads in the terms originally coined for describing standard file downloads, the salient difference being that a single file download is usually composed of tens of smaller downloads of different fragments of the file from different file 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 users, 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, we estimate that $X/\text{FileSize-of-Y}$ download cycles of the file have been passed over the network.

2.2. The Traces

The 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) for our analysis. Table 1 summarizes the main characteristics of the traffic captured.

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

Table 1: Main characteristics of collected Kazaa traces.

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 come to dominate the 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 for the behavior observed, it is an important step in characterizing these applications from a network perspective. For example, [5] reports that although the distribution of generated P2P traffic is highly skewed at the individual host level, the fraction of 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] explore the topology of 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 (FastrTrack) 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].

[4] were the first to characterize the Kazaa traffic. They note that Kazaa traffic constitutes most of the Internet traffic, show that a tiny number of files generates most of the download activity, postulate the feasibility of traffic caching, and empirically demonstrate its benefits. [10] reaffirms these findings and compares Kazaa traffic with traffic generated by traditional content distribution systems (i.e. Akamai and Web traffic).

We believe the traces analyzed in these two studies are complementary: the user population in our traces reflects a more diverse user population with significant heterogeneity in user connectivity, and a community where users pay network usage charges upfront (as opposed to an university where users are not charged for network usage).

In this paper we expand results presented in [4] (we note that the basic characteristics remain valid on current traffic), and 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 logs of Kazaa traffic: this amounts to 20 TB of traffic, about 300,000 different files, and at least different 50,000 users. Since files are generally downloaded from multiple sources we processed the logs to compute the number of download cycles for each file as detailed above. We then produced a list of files sorted by the number of download cycles. We used it to generate a CDF which

shows the percent of downloads cycles for each progressing subset of the most downloaded files.

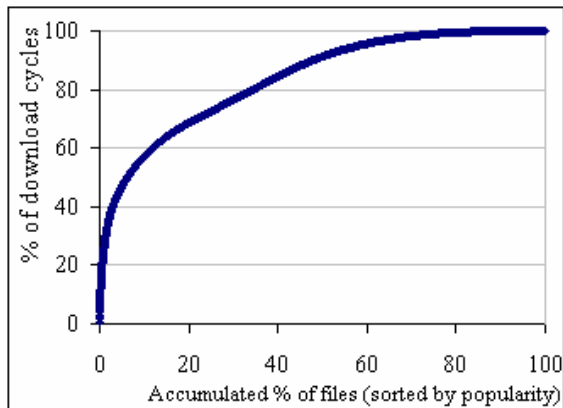


Figure 1: CDF for file download cycles. Note that less than 90% of all download sessions attempted actually succeed.

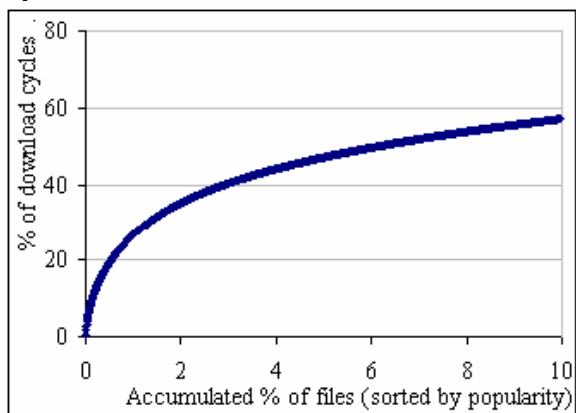


Figure 2: CDF of file download cycles for the 10% most downloaded files. Note that 35% of all downloads go to the 2% most popular files.

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.

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 its 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 the above, plotting 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 noticed in the previous graph is much more pronounced: we observe that as little as 2500 files (a mere 1% of all detected files) account for as much as 80% of the traffic.

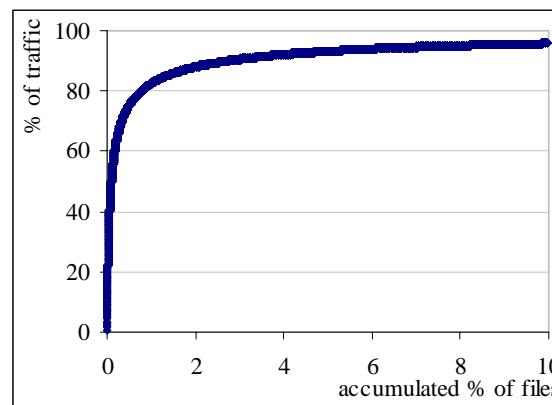
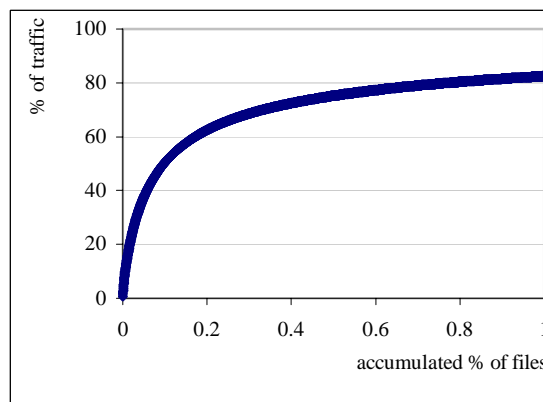


Figure 3: CDF for byte popularity distribution for the 10% most popular files.

Figure 4: CDF for byte popularity distribution for



the top 1% most popular files.

We note that our measurements show a byte popularity distribution significantly more skewed than UW traces [10]. While in 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 popular files generate 50% of the traffic.

4.3. File Sizes

We now switch gear 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. Roughly these are: 100KB for pictures, 2-5MB for music files, 50-150MB for applications and movie clips, and larger than 100MB for movies files.

Additionally, similar to the analysis in the sections above, we are interested in two other aspects: the number of downloads, and the 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 blue plot).

While the plots have similar structure, the plot representing the CDF of file sizes 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 large file are either smaller than 150 MB (clips and applications) or larger than 700MB (movies and games).

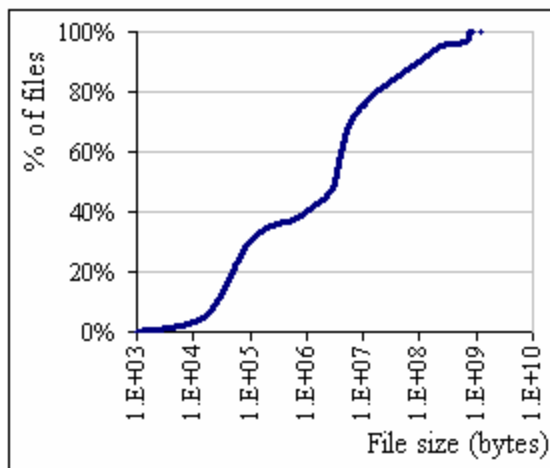


Figure 5: File size cumulative distribution function. The ‘steep’ portions of the distribution reflect ranges with a larger number of files: 2-5MB for music files, around 100KB for pictures and larger than 700MB for movies probably. (Note the logarithmic scale on X axis in this figure and the normal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

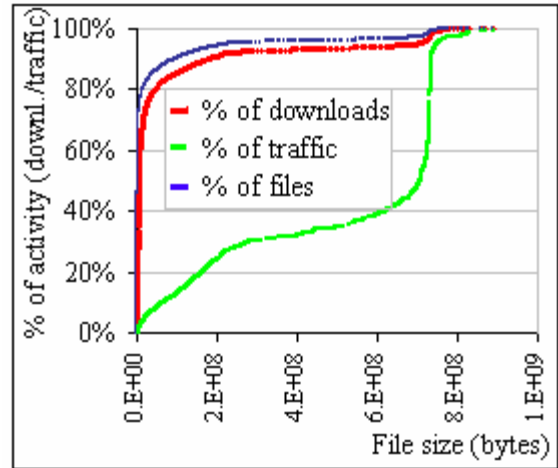


Figure 6: Activity CDF. Red plot presents the CDF for number of download cycles while the green plot presents the CDF for the generated traffic. The blue plot representing size CDF is present for reference. The same regions visible in Figure 5 are present. Note again that files in the 700-900MB range generate most of the traffic.

Figures 7 and 8 try to uncover possible correlations between file sizes and the activity generated in terms of bytes transferred (Figure 7) or number of completed download cycles (Figure 8).

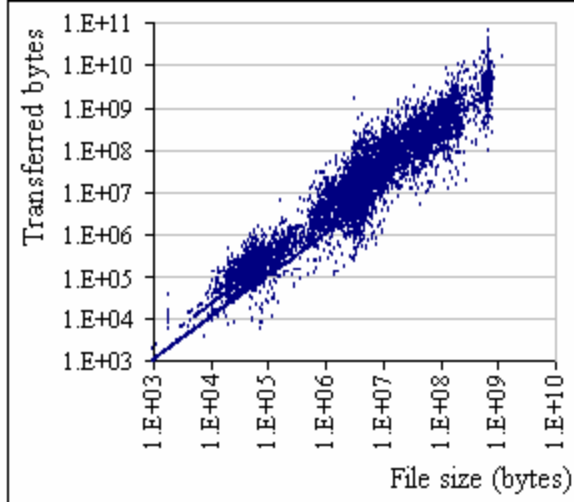


Figure 7: Roughly linear correlation between the file size and the traffic generated by downloading each file (logarithmic scales on both X and Y axes).

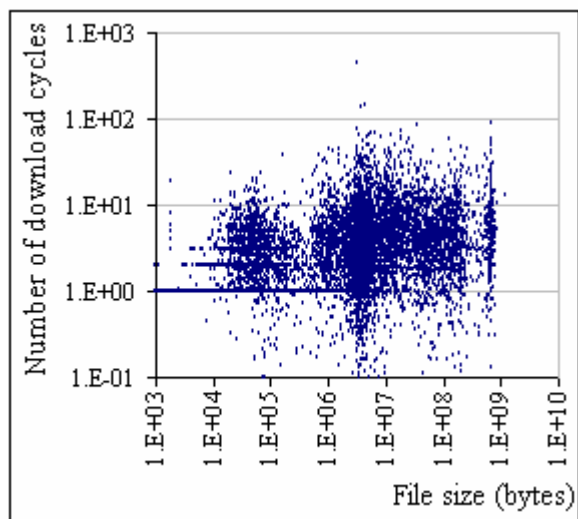


Figure 8: The four categories of files are evident again: around 100K pictures, music files roughly around 5MB, applications and movie clips 50-150MB, and movies files larger than 700MB.

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, heterogeneity of 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. 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 for each time unit the number of distinct *new* files that were observed during that time unit, in the sense that they have not yet been 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 9 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 of the requests (files once requested will be requested again soon). The seasonal pattern observed on Figure 9 follows a period of 24 hours with night-time peaks. This seasonality is easily explained, since the majority of the users are in the same time zone.

In order to evaluate the rate of change, we show in Figure 10 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 files per minute. This stability, however, is superficial, as evident by the constant slope at the hourly resolution plotted in Figure 9.

In order to better understand the behavior and enable extrapolation, Figure 11 we plot the values at a 1-day resolution, which avoids the daytime/nighttime cycle. The persistent decrease in the rate of encountering new files, even after 16 days is clearly visible.

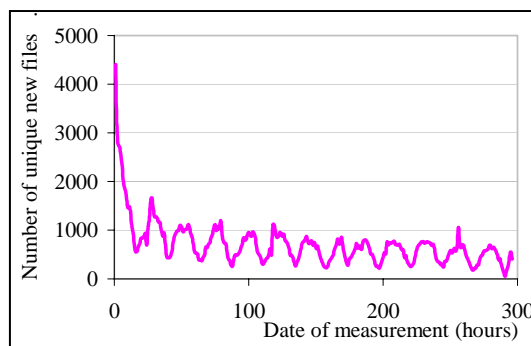


Figure 9: New files encountered during one hour long intervals for our 17-day trace.

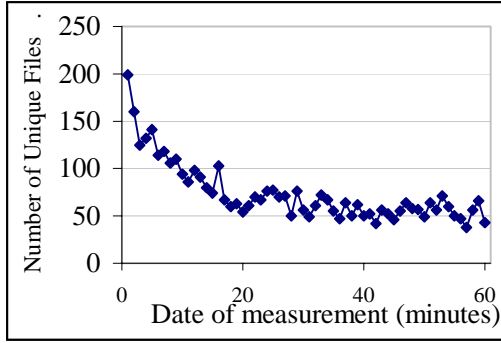


Figure 10: New unique files by minute for the first hour in our trace

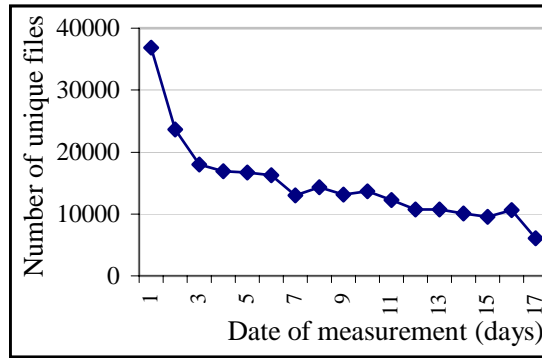


Figure 11: New files encountered during a one day interval for our 17-day trace..

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 for each 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 $N \in \{4, 6, 12, 24, 50, 100, 200, 400\}$. The observation periods are approximately 24 hour

intervals. The popularity of a file is measured by the number of download cycles of the file.

We investigate the characteristics of those files that persist between observation periods. First, we compare the lists of most popular files of all the observation periods with a chosen base list. As the base list we chose the N popular files from the first observation period (and later verify that results are not affected by the choice of the base list). We find the intersection of each of the lists with this base list, and calculate the intersection size as percent of the base size. In Figure 12 we plot these values as function of time, for $N \in \{4, 50, 400\}$, since for other values of N the results did not differ significantly. We obtain a measure of the percentage of recurrent popular files after t observation periods.

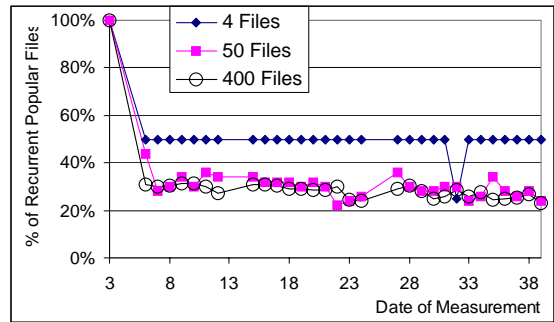


Figure 12: Ratio of the popular files set that remains stable during consecutive time periods.

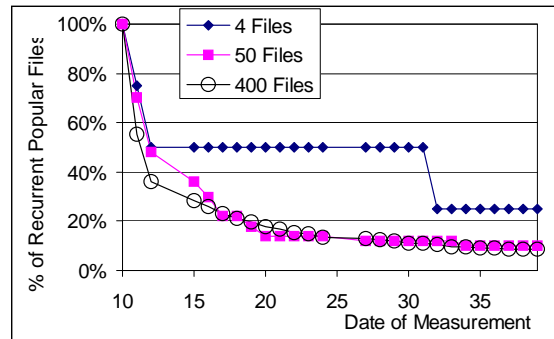


Figure 13: Ratio of the popular files set that remains stable when compared with a base period.

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.

In the second part of our analysis we calculated an intersection of subsequent lists of N most popular files. For every new observation period, we intersected the list with the intersection of the lists from all previous observation periods. On Figure 13, 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 t observation periods stabilizes at about 15% for increasing t . This suggests that there are indeed a number of "all-time favorites" during our observation.

The number of files that remain popular after t observation periods is larger than the number of files that are popular in all the observation periods. This suggests that the set of cachable 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 the group of 30% of cachable popular files, and further quantify their rate of change over months.

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 a recent study [11] that analyzes the Web and a high-energy physics collaboration and uncovers 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 explored the distribution of download activity (traffic and number of requests) over the set of user names (Figure 14) 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 between the three of them). We believe these are 'outliers': in fact multiple users that have not configured their software and this run under the default user name (e.g. *defaultuser* or *kazaliteuser*). Therefore, for the analysis in this section, we purged out all activity generated under these user names.

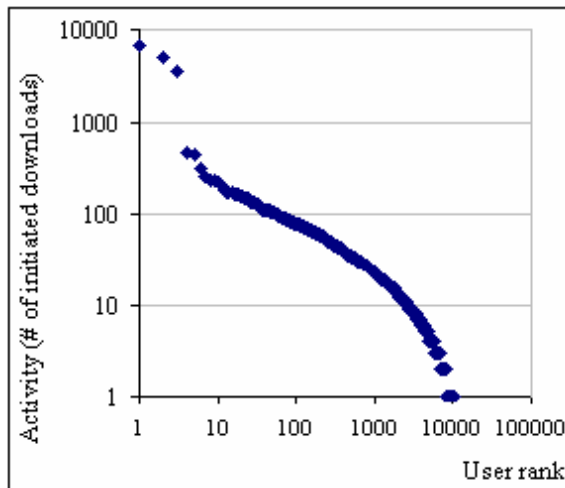


Figure 14: Activity distribution over the user name space. Users are ordered in decreasing order of the number of downloads they initiate. (logarithmic scales on both X and Y axes).

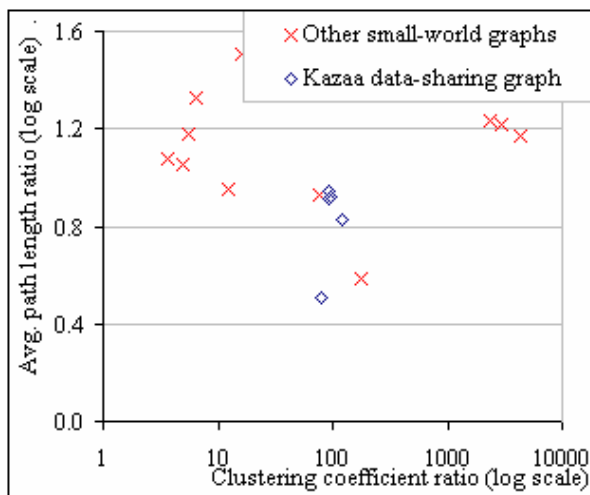


Figure 15: Comparing Kazaa's data-sharing graphs with a selection of well-known, small-world graphs, including citations network, power grid, movie actors, Internet, Web.

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 2 days long Kazaa trace and we vary m from 1 to 5 and T from 4 to 48 hours.

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

Similarity criteria used	Graph size (avg.)		Average path length (avg.)		Clustering coefficient	
	# nodes	# links	D S graph	R and.g raph	DS graph	Ra nd graph
$m=1, T=4h.$	1 585	854 6	4 .01	4 .41	0.6 53	0.0 070
$m=1, T=8h.$	2 038	142 67	3 .76	4 .08	0.6 45	0.0 068
$m=1, T=12h.$	3 033	299 91	3 .31	3 .50	0.6 05	0.0 065
$m=2, T=24h.$	1 311	522 7	3 .72	4 .51	0.4 83	0.0 040

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 [12].

Table 2 presents the average path-length and the clustering coefficient (averaged over multiple intervals of equal duration) of 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.

Figure 15 compares these data-sharing graphs with a selection of well-known, small-world graphs, including citations network, power grid, movie actors, Internet, Web [13]. 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.

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 very large, very 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 set of files that are constantly popular, and a larger set which lose their popularity within days. We note that a longer experimentation period and further analysis are required to quantify these measures.

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

Authors would like to thank Ian Foster, Adriana Iamnitchi and Anne Rogers for discussion and insightful comments.

Funding: Griphyx ?

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] M. Ripeanu, A. Iamnitchi, and I. Foster, "Data-Sharing Relationships in the Web," University of Chicago technical report, TR-2003-01, Chicago February 2003.
- [12] D. J. Watts, *Small Worlds: The Dynamics of Networks between Order and Randomness*: Princeton University Press, 1999.
- [13] R. Albert and A. L. Barabasi, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, vol. 74, pp. 47--97, 2002.