A Global View of KAD

Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack
Institut Eurecom
Sophia–Antipolis, France
{steiner,ennajjar,erbi}@eurecom.fr

ABSTRACT

Distributed hash tables (DHTs) have been actively studied in literature and many different proposals have been made on how to organize peers in a DHT. However, very few DHTs have been implemented in real systems and deployed on a large scale. One exception is KAD, a DHT based on Kademlia, which is part of eDonkey2000, a peer-to-peer file sharing system with several million simultaneous users. We have been crawling KAD continuously for about six months and obtained information about the total number of peers online and their geographical distribution.

Peers are identified by the so called KAD ID, which was up to now assumed to remain the same across sessions. However, we observed that this is not the case: There is a large number of peers, in particular in China, that change their KAD ID, sometimes as frequently as after each session. This change of KAD IDs makes it difficult to characterize *end-user* availability or membership turnover.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Systems and Software – Distributed systems, Performance evaluation (efficiency and effectiveness)

General Terms

Algorithms, Measurement, Performance

Keywords

Peer-to-Peer, lookup, distributed hash table

1. INTRODUCTION

Peer-to-peer systems have seen a tremendous growth in the last few years and peer-to-peer traffic makes a major fraction of the total traffic seen in the Internet. The dominating application for peer-to-peer is file sharing. Some of the most popular peer-to-peer systems for file sharing have been Napster, FastTrack, BitTorrent, and eDonkey, each one counting a million or more users at their peak time. Since these systems are mainly used by home-users and since the content shared is typically copyright-protected, the users of these systems often stay only connected as long as it takes for them to download the content they are interested in. As a result, the user population of these peer-to-peer systems is highly dynamic with peers joining and leaving all the time.

In this paper, we focus on a single peer-to-peer system, namely KAD, which is the publishing and search network of eDonkey. We want to characterize KAD in terms of metrics such as arrival/departure process of peers or the number of new peers joining the network.

To obtain the relevant raw data needed we decided to "crawl" KAD. Each crawl gives a snapshot of the peers active at that instant. The three major challenges in crawling are

- Time necessary to carry out a single crawl, which should be as small as possible to get a consistent view of the system.
- Frequency of the crawls, i.e. the time elapsed between two consecutive crawls should be small (no more than a few minutes) in order to achieve a high resolution for metrics such as session length.
- Duration of the crawl, which should be in the order of many months, to be able to correctly capture the longest session and inter-session lengths.

To meet all these goals, we built our own crawler. Early results have been published in [12].

While peer-to-peer systems have been explored previously using a crawler, the duration of these crawls was limited to a few days at best. We were able to crawl KAD for almost six months at a frequency of one crawl every five minutes, which allowed us to obtain a number of original results. We observed that:

- the lifetime of a significant fraction of the peers observed can be as short as a single session. We could explain in part this surprising behavior by the fact that peers change their KAD ID, which is contrary to the assumption that KAD IDs are persistent.
- when classifying peers according to their geographic origin, the peers from China make about 25% of all peers seen at any point of time and Europe is the continent where KAD is most popular. We also saw a big difference between peers in China and Europe with respect to some of the key metrics such as session length or daily availability.

The remainder of the paper is organized as follows. Sections 2 and 3 present related work and background information on KAD. Section 4 presents the measurement methodology followed by Section 5 that contains the results. Section 6 concludes the paper.

2. RELATED WORK

Overnet was the first widely deployed peer-to-peer application that used a DHT, namely Kademlia. The implementation of Overnet is proprietary and its operation was discontinued in September 2006 after legal actions from the media industry. Overnet has been the subject of several studies such as [2, 4, 8] and up to 265,000 concurrent users have been seen online. The study most relevant to our work is the one by Bhagwan et al. [2]. A set of 2,400 peers was contacted every 20 minutes during two weeks. This study pointed out the *IP aliasing problem* that is due to the fact that many peers periodically change their IP address. So, in order to properly compute session times and other peer-specific metrics one needs to use the global identifier of the peer-to-peer system. This study also indicates, that for systems where peers leave permanently, the mean peer availability decreases as the observation period considered increases.

KAD is the first widely deployed open-source peer-to-peer system relying on a DHT. Two studies on KAD have been published by Stutzbach. The first explains the implementation of Kademlia in eMule [15] and the other [16] compares the behavior of peers in three different peer-to-peer systems, one being KAD. The results obtained for KAD are based on crawling a subset of the KAD ID space. We call a continuous subset of the total KAD ID space that contains all KAD peers whose KAD IDs agree in the high order k bits a **k-bit zone**. Stutzbach [16] crawled a 10-bit zone in 3-4 minutes and a 12-bit zone in approximately 1 minute. A total of 4 different zones were crawled during 2 days each.

3. BACKGROUND ON KAD

KAD is a Kademlia-based [6] peer-to-peer DHT routing protocol implemented by several peer-to-peer applications such as Overnet [9], eMule [3], and aMule [1]. The two open–source projects eMule and aMule do have the largest number of simultaneously connected users since these clients connect to the eDonkey network, which is a very popular peer-to-peer system for file sharing. Recent versions of these clients implement the KAD protocol.

Similar to other DHTs like Chord [14], Can [10], or Pastry [11], each KAD node has a global identifier, referred to as KAD ID, which is 128 bit long and is randomly generated using a cryptographic hash function. The KAD ID is generated when the client application is started for the first time and is then permanently stored. The KAD ID stays unchanged on subsequent join and leaves of the peer, until the user deletes the application or its preferences file[1]. Therefore, using the KAD ID, a particular peer can be tracked even after a change of its IP address.

3.1 Routing

Routing in KAD is based on prefix matching: Node a forwards a query, destined to a node b, to the node in his routing table that has the smallest XOR-distance. The XOR-distance $d(a, b)$ between nodes a and b is $d(a, b) = a \oplus b$. It is calculated bitwise on the KAD IDs of the two nodes, e.g. the distance between $a = 1011$ and $b = 0111$ is $d(a, b) = 1011 \oplus 0111 = 1100$. The fact that this distance metric is symmetric is an advantage compared to other systems, e.g. Chord, since in KAD if a is close to b, then b is also close to a.

[1]As we will see later, not all peers in KAD behave this way.

The entries in the routing tables are called *contacts* and are organized as an unbalanced *routing tree*: A peer P stores only a few contacts to peers that are far away in the overlay and increasingly more contacts to peers as we get closer P. For details of the implementation see [15]. For a given distance P knows not only one peer but a *bucket* of peers called contacts. Each bucket can contain up to ten contacts, in order to cope with peer churn without the need to periodically check if the contacts are still online.

For routing, a message is simply forwarded to one of the peers from the bucket with the longest common prefix to the target. Routing to a specific KAD ID is done in an iterative way, which means that each peer on the way to the destination returns the next hop to the sending node.

3.2 Publishing

A **key** in a peer-to-peer system is an identifier used to retrieve information. KAD distinguishes between two different keys:

- A **source key** that identifies the content of a file and is computed by hashing the *content* of a file.
- A **keyword key** that classifies the content of a file and is computed by hashing the tokens of the *name* of a file.

In KAD keys are not published just on a single peer that is numerically closest to that key, but on 10 different peers whose KAD ID agrees at least in the first 8-bits with the key. This zone around a key is called the tolerance zone.

Keys are periodically republished: **source keys** every 5 hours and, **keyword keys** every 24 hours. Analogously, a peer on which a source key or keyword key was published will delete the information after 5 and 24 hours, respectively. This way re-publishing is done exactly the same way as publishing.

4. MEASUREMENT METHODOLOGY

We have developed Blizzard, our own crawler for KAD, with the aim to crawl KAD frequently and over a duration of several months. Our crawler logs for each peer the time of the crawl, the IP address of the peer, and its KAD ID.

In a large peer-to-peer system such as KAD, peers are constantly joining and leaving, which makes it difficult to get a consistent view of the system. Therefore, the overall duration of a single crawl should be as short as possible. The main idea of Blizzard is to use *only one machine* and keep all relevant information in main memory. After the crawl is completed, the results are written to disk. The implementation of Blizzard is straightforward: Blizzard runs on a local machine knowing several hundred contacts to start with. It uses a simple breadth first search and iterative queries: It first queries peers among its contacts in order to get to know more peers and so on. The communication is asynchronous: requests are sent out by one thread while a second thread stores the results. In this way the delay between the request and the corresponding answers does not slow down the crawling. In fact the speed of our crawler is limited by the available bandwidth (100 Mbit/s bi–directional).

There are various pitfalls when crawling a peer-to-peer system, such as incomplete data due to crawler crashes, loss of network connectivity, or random failures due to temporary network instability. To address these problems, we run simultaneously two instances of Blizzard, one at the University of Mannheim, Germany, connected to the German research network, and a second one at Institut Eurécom, France, connected to the French academic network.

The speed of Blizzard allows us to crawl the entire KAD system (entire KAD ID space), which was never done before. Such a **full crawl** of KAD takes about 8 minutes. The first million different peers are identified in about 10 seconds, the second million in 50 seconds, thereafter the speed of discovery decreases drastically since most of the encountered peers have already be seen before during the same crawl. A full crawl of KAD produces about 3 GBytes of inbound and outbound traffic each.

A full crawl was done three times a day from 2006/08/18 to 2006/08/26 and from 2006/10/03 to 2006/10/12. Another full crawl has been started 2007/03/20 and is carried out up to now once a day.

A full crawl generates an extremely high amount of trace data and of network traffic. Carrying out just 3 crawls per day is not really sufficient to capture the dynamics of KAD at short timescales. For this reason, we decided to carry out a **zone crawl** on a 8-bit zone, where we try to find all active peers whose KAD ID have the same 8 high-order bits. Such a zone crawl, which explores one 256-th of the entire KAD ID space, takes less than 2.5 seconds. A zone crawl for the KAD IDs whose 8 high order bits are `0x5b` was done once every 5 minutes from 2006/09/23 to 2007/03/21, which is *slightly less than 6 months*.

5. GROUP–LEVEL DYNAMICS OF PEERS

In this section, we will present results about the full crawl of the system and the zone crawl, moreover we will characterize the fact of IP address aliasing and KAD ID aliasing.

5.1 Full Crawl

A good hash function should assure a uniform distribution of the KAD IDs over the entire KAD ID space. We see that the peers are indeed uniformly distributed over the hash space, except for some outliers (Figure 1). An 8-bit zone contains the peers whose KAD ID agrees in the first 8 bits, thus one zone can theoretically contain 2^{120} hash values, while we actually observe between 12,000 and 20,000 peers per zone. For some zones we see a much higher number of peers, which are due to modified KAD clients. The modified KAD clients with the same KAD ID are always limited to a single country (Korea, Spain, Israel, China, Argentina). The outlier in zone `0xe1` is a modified client, for which we counted more than 10,000 instances. Using modified KAD clients amounts for some sort of free-riding, since the load for publishing and forwarding that a single regular KAD peer carries will be distributed among all the modified peers with the same KAD ID. On the other hand, modified clients can use the peer-to-peer network like a regular client. If there were too many modified KAD clients, the application performance would decrease, since the peers using the same KAD ID cannot see one another and therefore cannot download one from another.

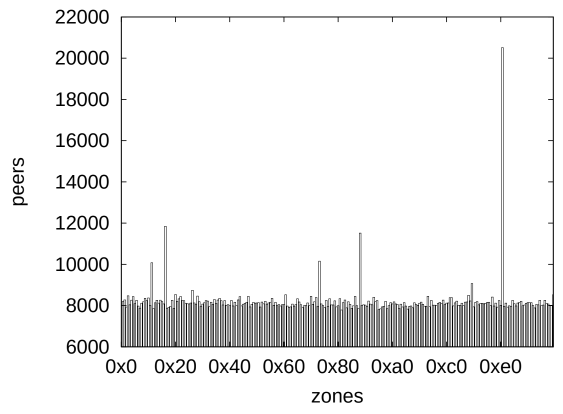


Figure 1: The distribution of the peers over the hash space. The 256 8-bit zones on the x-axis go from `0x00` to `0xff`.

During a full crawl, we found between 3 and 4.3 million different peers. Between 1.5 to 2 million are not located behind NATs or firewalls and can be *directly contacted* by our crawler. Peers behind NATs or firewalls use KAD to publish information about the content they share, but do not *participate* in storing published information, and make therefore no contribution to the operation of KAD.

In the rest of the paper we will only report statistics on the peers that our crawler could contact *directly*. As we can see in Figure 2 the number of peers seen varies according to a diurnal and a weekly patterns and reaches its peak during the weekend, where the population is about 10% higher.

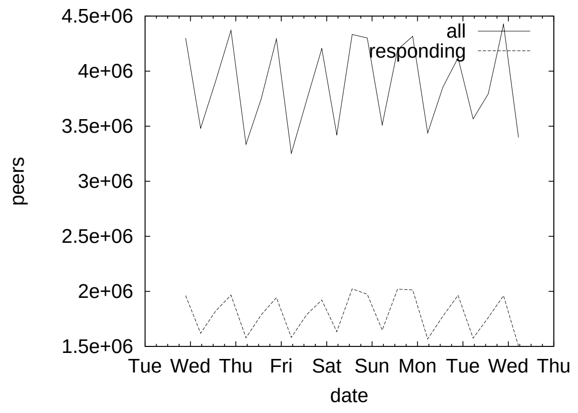


Figure 2: The number of kad peers available in entire kad ID space depending on the time of day.

In Figure 3, we plot the distribution of the percentage of peers seen per country. We used the Maxmind database [5] to resolve IP addresses to countries and ISPs. The continent with the highest percentage of peers is Europe (Spain, France, Italy and Germany), while the country with the largest number of peers is China. Less than 15% of all peers are located in America (US, Canada, and South America).

We can also see that the geographic distribution of the peers obtained with the two zone crawls of an 8-bit zone each is very close to the result obtained with the full crawl, which is to be expected since the KAD IDs are chosen at random.

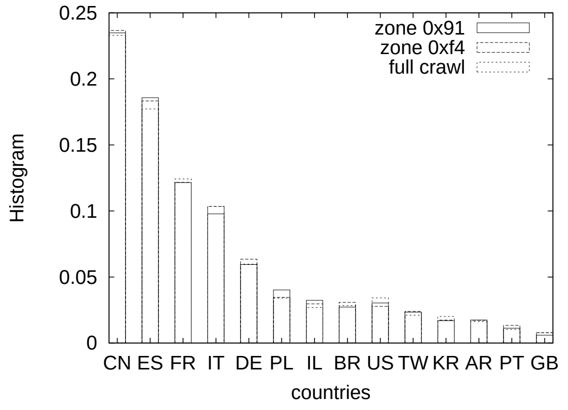


Figure 3: Histogram of geographic distribution of peers seen on 2006/08/30.

We can therefore estimate the total number of peers in KAD by simply counting the number of peers in a zone. We do a zone crawl of one 256-th of the entire KAD ID space and use Chernoff Bounds (see [7] Chapter 4) to estimate the total population size and to tightly bound the estimation error.

Since the full crawl was quite expensive in terms of resources, we will extensively rely on the zone crawl to obtain much of the relevant information about KAD.

5.2 Zone Crawl

All the results in the following subsection were obtained using the zone-crawl of the 8-bit zone `0x5b` that lasted for 179 days. During this period a total of 51,552 crawls were executed.

In Figure 4, we plot the number of peers seen that originate from China and some European countries. For the peers of each country, we can clearly identify a diurnal pattern, with a peak around 9 PM local time. The eight hour time shift between Europe and China is clearly visible.

Table 1 summarizes the basic findings on the zone crawl. The peers seen came from 168 different countries and 2384 providers. For the KAD IDs seen the 1st day of our zone crawl, we observe that about $\frac{1}{3}$ of the peers come from Europe and about $\frac{1}{4}$ from China. If we compare the **lifetime** of the peers, which is defined as the difference between the time a given KAD ID was seen the last time and the time this KAD ID was seen the first time, we notice that the lifetime of peers in China is much smaller than the one for peers in the other countries. More than half of the peers in China were seen for the duration of only *one* session. We will come back to this point in subsection 5.3.

Arrivals and Departures

Since we crawl KAD once every 5 minutes, we can determine the number of peers that join and leave between two consecutive crawls. Knowing the arrival rate of peers is useful since it allows to model the load in KAD due to newly joining peers. Each time a peer joins, it first contacts other peers for information to populate its routing table, before it publishes the keywords and source keys for all the files it will share.

In figure 5(a) we depict the CDF (cumulative distribution function) of the number of peers that arrive and that depart between two consecutive crawls. We see that the distributions for arrivals and departures are the same.

The arrival process is very well described by a Negative Binomial distribution (see figure 5(b)).

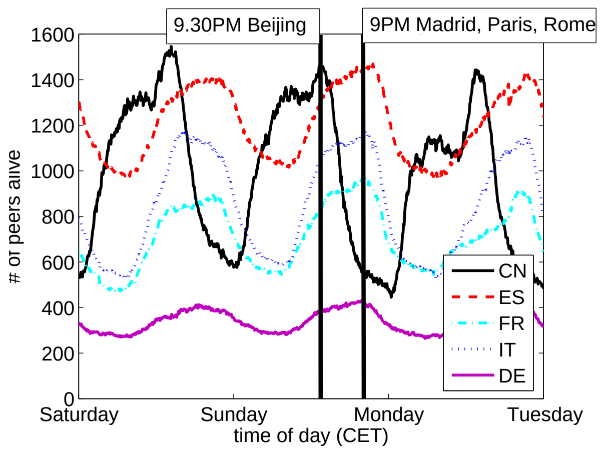


Figure 4: Peers online according to country of origin.

5.3 Aliasing

IP Address Aliasing

It has been known for quite some time [2, 4] that peers may get frequently assigned new IP addresses, which is referred to as **IP address aliasing**. We observed a total of 400,278 distinct KAD IDs and 3,228,890 different IP addresses (see table 1). As we see in figure 6, the number of different IP addresses per peer is strongly correlated with the peer lifetime.

KAD ID Aliasing

Figure 7 reports the number of *new* KAD IDs per day. i.e. KAD IDs seen for the first time, according to country of origin. More than 50% of the new KAD IDs are from peers in China, which is more than one order of magnitude more than the number of new KAD IDs seen for any other country such as Spain, France, or Germany.

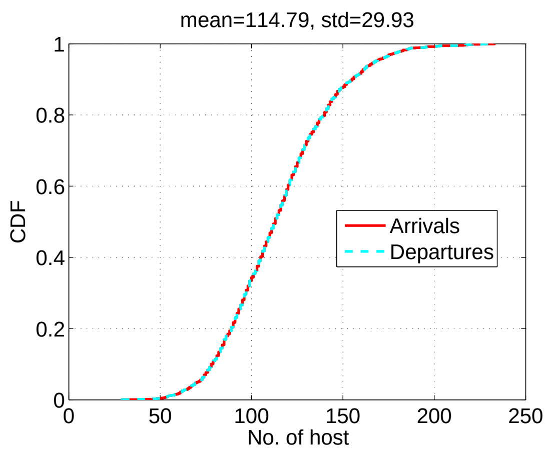
We see in our zone crawl approx. 2,000 new KAD IDs a day, which means that for the entire KAD system the number of new KAD IDs per day is around 500,000. If we extrapolate, this makes about 180 Million KAD IDs a year. It is hard to believe that there exist such a large number of different end-users of KAD.

We were curious to find out whether the end-users really stop using KAD after one session, or whether the same users come back with a different KAD ID. We refer to the phenomenon of non-persistent KAD IDs as **kad ID aliasing**, in contrast to IP address aliasing.

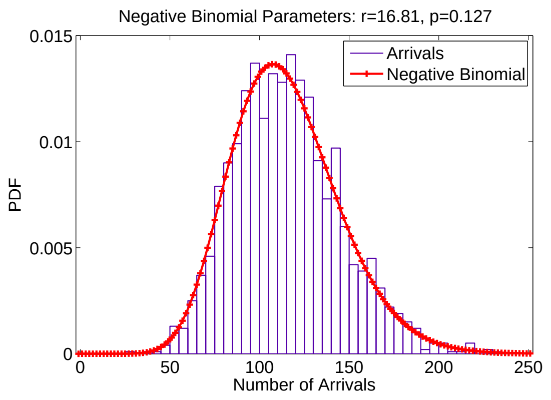
To investigate KAD ID aliasing, we need to look for peers

	Total	China	Europe	Rest
Different KAD IDs	400,278	231,924	59,520	108,834
Different IP addresses	3,228,890	875,241	1,060,848	1,292,801
KAD IDs seen for a single session	174,318	131,469	11,644	31,205
KAD IDs with LT $\leq$ 1 day	242,487	183,838	15,514	43,135
KAD IDs seen for the first time on				
- 1st crawl	5,670	455	2,879	2,336
- 1st day	18,549	4,535	6,686	7,328
- 60th day	1,893	1,083	259	551
KAD IDs seen for the first time on 1st day				
- with LT $\leq$ 1 day	2,407	1,568	286	553
- 1 day $<$ LT $\leq$ 1 week	1,368	497	393	478
- 1 week $<$ LT $\leq$ 1 month	2,735	791	944	1,000
- LT $>$ 1 month	12,039	1,679	5,063	5,297
- LT $>$ 3 months	8,423	936	3,679	3,808

Table 1: Key facts about the zone crawl spanning 179 days (LT=Lifetime).



(a) CDF of the number of arrivals and departures.



(b) PDF of the number of arrivals

Figure 5: Peer arrivals between two crawls during the first week.

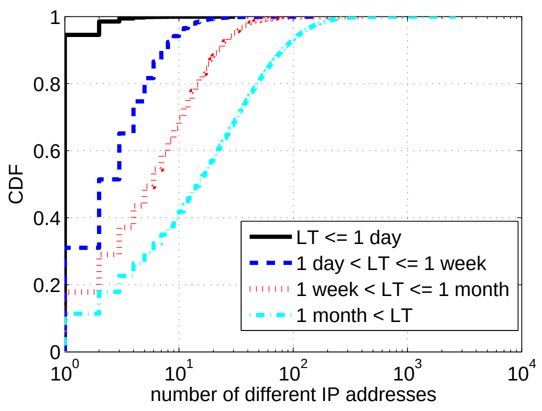


Figure 6: The number of different IP addresses reported per kad ID during a period of six months.

with static IP addresses, wich we can track for non-persistent KAD IDs. We know that, for instance in France, one of the ADSL providers (Proxad) assigns static IP addresses to customers located in areas where the service offer is completely “un-bundled”, while France Telecom-Orange changes the IP addresses of ADSL customers on a daily basis.

To find peers with static IP addresses, we started in March 20th, 2007 to carry out one full crawl a day. We take the logs of the two full crawls of March 20th and March 30th and extract the 160,641 peers that have the same IP address and the same KAD ID in both logs. We call this set of IP addresses **pivot set**. Our hypothesis is that a peer that keeps the same IP address for 10 days is assigned a static IP address. We then take the logs of the full crawls starting April 1st, 2007 to look for peers whose IP addresses are in the pivot set that have *changed their* KAD ID.

In figure 8, we plot the rate of change of KAD IDs for the peers in the pivot set. Up to now, all the publications on KAD assumed persistent KAD IDs. Our observations clearly disagree with this affirmation, since we see that a significant fraction of end-users in different countries change their KAD ID over time. The rate of change among the Chinese peers is highest with about 35% after *one month*, second is Spain with close to 20%, while the average is about 10%.

We have no good explanation for why end-users change their KAD IDs. In the case of China, we suspect that there exists a “Chinese implementation” of the KAD protocol that uses a new KAD ID for every session.

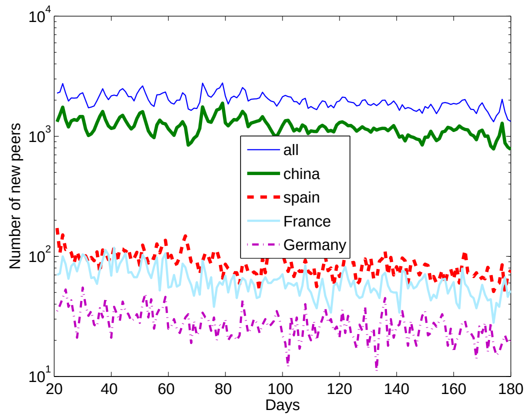


Figure 7: New kad IDs according to country of origin.

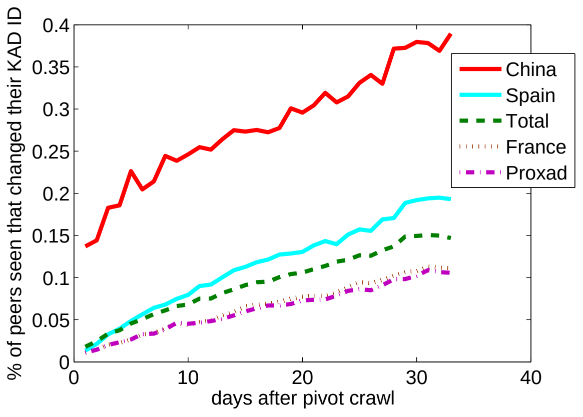


Figure 8: The fraction of peers in pivot set that changed their kad ID at least once.

Implications of KAD ID *Aliasing*

The fact that the KAD ID assigned may be non-persistent obliges us to distinguish between a peer and an end-user:

- A **peer** is an instance of KAD identified by a fixed KAD ID.
- An **end-user** is a physical person that launches a peer to participate in KAD. The same end-user can, at different times, participate in KAD via different peers.

When KAD ID aliasing occurs, it is not really possible to characterize the lifetime of *end-users*, as compared to the lifetime of peers. All we can extract from our crawl data is the lifetime of *peers*, which provides us with a lower bound on the lifetime of end-users.

6. CONCLUSION

We have presented results on the peer behavior in KAD, the largest currently deployed DHT. The duration of our crawl was 179 days, which makes it to our knowledge, the longest crawl of a peer-to-peer system ever carried out.

We saw KAD IDs are not necessarily persistent as was assumed so far. It remains an open problem to explain why KAD IDs are non-persistent and under what circumstances peers change their KAD ID.

We also obtained interesting results concerning the behavior of *individual* peers that could not be presented here for space reasons. The most important findings are that

- Session lengths have a "long tail", with sessions lasting as long as 78 days.
- The distribution of the session lengths is best characterized by a Weibull distribution,with shape parameter $k < 1$. One property of Weibull distributed session lengths is that a peer that has so far been up for t units of time will – in expectation – remain up for a duration that is in the order of $O(t^{1-k})$.
- For many peers, the amount of time a peer is connected per day varies a lot from one day to the next.

For more details, we refer the interested reader to our technical report [13].

Acknowledgment: We would like to thank D. Carra and F. Pianese for feedback on an early version of this paper and our shepherd R. Rejaie for his constructive suggestions. We also thank the Computing Center of the University of Mannheim for providing us with the resources necessary to carry out such a long crawl.

7. REFERENCES

[1] A-Mule. `http://www.amule.org/`.

[2] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, pages 256–267, 2003.

[3] E-Mule. `http://www.emule-project.net/`.

[4] K. Kutzner and T. Fuhrmann. Measuring large overlay networks - the overnet example. In *Proceedings of the 14th KiVS*, Feb. 2005.

[5] Maxmind. `http://www.maxmind.com/`.

[6] P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-peer informatiion system based on the XOR metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 53–65, Mar. 2002.

[7] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge Press, 2005.

[8] N. Naoumov and K. Ross. Exploiting p2p systems for ddos attacks. In *International Workshop on Peer-to-Peer Information Management*, May 2006.

[9] Overnet. `http://www.overnet.org/`.

[10] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. ACM SIGCOMM*, 2001.

[11] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale Peer-to-peer systems. In *Proceedings of Middleware*, Heidelberg, Germany, November 2001.

[12] M. Steiner, E. W. Biersack, and T. Ennajjary. Actively monitoring peers in kad. In *Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS'07)*, 2007.

[13] M. Steiner, T. En-Najjary, and E. W. Biersack. Analyzing Peer Bahavior in KAD. Technical report, Institut Eurecom, May 2007.

[14] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-to-peer lookup service for Internet applications. In *Proceedings of SIGCOMM*, pages 149–160, San Diego, CA, USA, 2001. ACM Press.

[15] D. Stutzbach and R. Rejaie. Improving lookup performance over a widely-deployed DHT. In *Proc. Infocom 06*, Apr. 2006.

[16] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proc. Internet Measurement Conference (IMC)*, Oct. 2006.