Measuring Freenet in the Wild: Censorship-resilience under Observation

Stefanie Roos†, Benjamin Schiller‡, Stefan Hacker‡, Thorsten Strufe†

†Technische Universität Dresden, <firstname.lastname>@tu-dresden.de
‡Technische Universität Darmstadt, <lastname>@cs.tu-darmstadt.de

Abstract. Freenet, a fully decentralized publication system designed for censorship-resistant communication, exhibits long delays and low success rates for finding and retrieving content. In order to improve its performance, an in-depth understanding of the deployed system is required. Therefore, we performed an extensive measurement study accompanied by a code analysis to identify bottlenecks of the existing algorithms and obtained a realistic user model for the improvement and evaluation of new algorithms.
Our results show that 1) the current topology control mechanisms are suboptimal for routing and 2) Freenet is used by several tens of thousands of users who exhibit uncharacteristically long online times in comparison to other P2P systems.

1 Introduction

Systems that allow users to communicate anonymously, and to publish data without fear of retribution, have become ever more popular in the light of recent events[1]. Freenet [1–3] is a widely deployed completely decentralized system focusing on anonymity and censorship-resilience. In its basic version, the Opennet mode, it provides sender and receiver anonymity but establishes connections between the devices of untrusted users. In the Darknet mode, nodes only connect to nodes of trusted parties. Freenet aims to achieve fast message delivery over short routes by arranging nodes in routable small-world network. However, Freenet's performance has been found to be insufficient, exhibiting long delays and frequent routing failures [4].

In this paper, we investigate the reasons for the unsatisfactory performance of the deployed Freenet. The evaluation of Freenet so far has mainly been based on theoretical analyses and simulations, relying on vague assumptions about the user behavior. Such analytical or simulative user models, however, often differ significantly from reality. We consequently measured the deployed system to shed light on two critical points. First, we analyzed the topology of Freenet and its impact on the routing performance. In particular, we considered the neighbor selection in the Opennet and the interaction between Opennet and Darknet.

[1] http://www.theguardian.com/world/the-nsa-files

Secondly, we measured the user behavior in Freenet with regard to number of users, churn behavior, and file popularity.

Our results indicate that the real-world topology differs largely from the assumptions made in the design, thus identifying a potential reason for the lack of performance. Over a period of 8 weeks, we discovered close 60,000 unique Freenet installations. With respect to their online behavior, the Freenet users exhibit a medium session length of more than 90 minutes, which is slightly longer than in other Peer-to-Peer systems. The session length distribution can be well modeled by a lognormal distribution and a Weibull distribution.

The results were obtained using both passive and active large-scale monitoring adapted to deal with the specific constraints of the Freenet protocol. They provide new insights into the actual workings of Freenet and can be used to design improved algorithms.

2 Background

In this Section, we introduce Freenet and present related work on measurements in P2P systems in general.

2.1 Freenet

Freenet was originally advertised as a censorship-resilient publication system [1, 2], referred to as Opennet. During the last years, the system has been extended to include a membership-concealing Darknet [3], where connections are only established to trusted users. Furthermore, the functionalities of Freenet have been extended beyond simple publication of content: Freesites, complete websites hosted in Freenet, offer the possibility to store and retrieve vast amounts of information[2]. An instant messaging system[3] and an email system[4] have been built on top of Freenet as well. All of these components use the same application-independent algorithms and protocols for storing, finding, and retrieving content, which are discussed in the following. First, we explain how users and files are identified in Freenet. Afterwards, we discuss how data is stored and retrieved, before detailing how the topology of Opennet and Darknet is created. Our descriptions are based upon [1, 2] for the Opennet, and [3] for the Darknet, as well as on the source code at the time of the respective measurement.

In Freenet, users and files are identified and verified using cryptographic keys. A user's public and private key are created upon initialization of her node and used to sign published files. In addition, each node has a location, i.e., a key from the key space that files are mapped to. In analogy to a peer's identifier in a distributed hash table, Freenet nodes are responsible for storing files whose key is close to their location. For files, various keys exist that all share the same key space derived from the SHA-1 hash function: The content hash key

[2] `https://wiki.freenetproject.org/Freesite`
[3] `https://freenetproject.org/frost.html`
[4] `https://freenetproject.org/freemail.html`

(*CHK*) is the hash of the file itself and can be used for checking its integrity. Keyword signed keys (*KSK*s) are the hash of a descriptive human-readable string enabling keyword searches. The signed subspace key (*SSK*) contains the author's signature for validating a file's origin. Recently, *SSK*s are often replaced by updateable subspace keys (*USK*s), which allow versioning of files. Public keys, required for the validation of signatures, can be obtained directly from the owner or from Freenet indexes, i.e., Freesites that provide lists of publicly available files, their descriptions, and keys.

File storage, discovery, and retrieval is based on a deterministic routing scheme, a distance-directed depth-first search. Unless a node can answer a request, it forwards the message to its neighbor whose location is closest to the target key. Each request is identified by a random message ID enabling nodes to detect and prevent loops. In case a node cannot forward the message to another neighbor, backtracking is applied (see [1]).

During a storage request, the file is stored by any node on the path whose location is closer to the file key than any of its neighbors, by the last node on the path, and by any node that was online for at least 20 hours during the last two days. When a file is found, it is sent back to the requesting node on the inverse path. The contact information of the responding node is added but probabilistically changed by any node on the path to conceal the origin's address. This should provide plausible deniablility, i.e., uncertainty which node actually provided the file.

In Opennet and Darknet, the overlay topology is established differently. Opennet nodes send join requests to publicly known seed nodes that forward the request based on the joining node's location. The endpoints of such requests can be added as neighbors. The maximum number of neighbors depends on a node's bandwidth. Binding the degree of a node to the bandwidth provides an incentive to contribute more bandwidth because high-degree nodes receive a better performance on average.[5]. Based on their performance in answering requests, neighbors can also be dropped to make room for new ones. In the Darknet mode, nodes only connect to trusted contacts, which have to be added manually. Instead of accepting new neighbors with close locations, Darknet nodes adapt their location to establish a better embedding into the key space [5]. Both the neighbor selection in Opennet and the location adaption in Darknet are supposed to structure the network such that the probability to have a neighbor at distance d scales with $1/d$ for $d \geq c > 0$ for some constant c. The design is motivated by Kleinberg's model: Nodes are arranged in a m-dimensional lattice with *short-range links* to those closest on the lattice. Furthermore, nodes at distance x are chosen as *long-range contacts* with a probability proportional to $1/d^r$. Kleinberg showed that the routing is of polylog complexity if and only if $r = m$ equals the number of dimensions [6]. Consequently, a distance distribution between neighbors that asymptotically scales with $1/d$ would be optimal for the 1-dimensional namespace of Freenet.

[5] `https://wiki.freenetproject.org/Configuring_Freenet#Connecting_to_the_Opennet`

2.2 Related Work

Most scientific publications on Freenet focus on the performance [5, 7] and attack resilience [8–10] of the routing algorithm. Their evaluations are based on theoretical analysis, simulations, and small-sized testbeds. The simulations in the original paper are based upon rather unrealistic assumptions such as no or uniform node churn, uniform content popularity, and uniform storage capacities [1,3]. So far, only two measurement studies have been performed in the real system, both with a rather small scope: The first, conducted in 2004, was an 18 days passive monitoring of the connection duration between neighbors. The average observed connection time was 34 seconds, indicating that Freenet nodes frequently change neighbors [11]. The second study, aiming at an estimation of Freenet's network size, was performed in 2009. For measurement purposes, 80 Freenet nodes were inserted into the network. These nodes were then manipulated to drop and establish new connections at a higher rate to increase the number of discovered nodes. During 80 hours of measurements, 11,000 unique node location were found. The number of concurrently online nodes was measured to be between 2,000 and 3,000 [4]. Hence, measurements on Freent so far are outdated and focus on single aspects of the protocol or user behavior only. The results are too general to suggest improvements and provide an accurate churn model for evaluating them. Alternative designs to Freenet for anonymous or membership-concealing P2P systems have been discussed in [4, 12–14]. However, they have not been widely deployed or rely on unstructured systems, which do not allow efficient resource discovery.

In contrast, there is vast related work on measurements in P2P systems in general. We briefly summarize their results regarding the user behavior in order to compare Freenet users to users of large-scale file-sharing networks without enhanced security protocols. The most frequently studied aspects of such systems are network size and churn. For the latter, the session length, i.e., the time a node stays online at a time, is of particular interest. The network size is usually determined by counting all nodes encountered during a certain time period. A subset of these nodes is then regularly contacted to track their online time and then derive a churn model from the observed data. How such a concept can be realized highly depends on the system under observation. In Freenet, contacting arbitrary nodes other than a node's direct neighbors is not possible. Hence, existing approaches can not be applied directly and are thus not discussed here in detail. The churn behavior of users has been measured in most large-scale P2P systems, in particular Napster [15], Gnutella [15], FastTrack [16], Overnet [17], Bittorrent [18, 19], and KAD [20, 21]. The observed median session length lies between 1 minute and 1 hour [22]. Measurements indicate that the shape of the session length distribution resembles a power-law: Exponential [18], Pareto [23], Weibull [21], and lognormal [21] distributions have been fitted. Our results show that the Freenet session length can be fitted reasonably well to a lognormal distribution, but the median online time is slightly higher than in all existing measurements of P2P-based systems.

3 Methodology

The data required for addressing most questions could be obtained using passive monitoring, i.e., using nodes that only observe the system and output additional log information. The analysis of users' churn behavior required us to perform active monitoring, i.e., running instrumented nodes that periodically request information.

We used Freenet version *1407* for all measurements prior to August 2012, version *1410* for measurements in September and October 2012, version *1442* for measurements in Spring 2013 and version *1457* for all later measurements[6].

In the remainder of this Section, we detail the two different monitoring approaches and describe how we extracted the desired information from the collected logs.

Locations of monitoring nodes were chosen uniformly at random unless stated otherwise. More sophisticated placement strategies would require additional knowledge of the global topology, which is not straightforward to obtain. The number of monitoring nodes varies over the experiments, depending both on the type of the measurement (e.g. local samples vs. global information needed) and the available resources at the time.

3.1 Passive Monitoring

We applied passive monitoring by inserting a set M of monitoring nodes in the network. They executed the normal code and followed the protocol like any regular node. We extended the Freenet logging mechanism to store all messages sent to and received from other nodes. The logged data allowed us to observe all changes in the neighborhood as well as all requests and the corresponding replies passing through these monitoring nodes.

Passive monitoring was used to collect data for the analysis of the neighbor selection, for determining the network size and the origin of users, for investigating file popularity and user activity, and for analyzing the impact of parallel Darknets.

Distance and Degree Distribution : The goal was to find out if the distances between neighbors in the overlay actually follow the distribution from Kleinberg's model [6]. In addition, we measured the degree distribution, which influences the routing success observed in the system.

Upon establishing a connection, nodes provide each other with their own location and the locations of their neighbors. Whenever the neighborhood changes, all neighbors are informed of the change. Hence, by logging all such messages, we obtained the degree of all neighbors of monitoring nodes and the distances between them and their neighbors. Denote the measurement duration by T. We took snapshots of the neighborhood of our monitoring nodes each t time units.

[6] `https://github.com/freenet/fred-staging/releases`

Let $G_k = (V_k, E_k)$ be a snapshot after $t \cdot k$ minutes for $k = 0 \dots K$ with $K = \lfloor T/t \rfloor$. The node set V_k consisted of our monitoring nodes M, the neighbors of nodes in M, and their neighbors. The subgraph G_k was induced, i.e., the edge set E_k consisted of all edges between nodes in V_k. We determined the empirical distance distribution of neighbors as the weighted average over all snapshots. Let $l(e)$ be the distance between the endpoints of edge e. Recall that for any set A, the indicator function $\mathbf{1}_A(x)$ is 1 if $x \in A$ and 0 otherwise. Then the empirical distance distribution $\hat{L}$ was computed by

$$P(\hat{L} \leq x) = \sum_{k=0}^{K} \sum_{e \in E_k} \frac{\mathbf{1}_{[-\infty,x)}(l(e))}{\sum_{k=0}^{K} |E_k|}. \tag{1}$$

When obtaining the degree distribution, our own nodes might not represent a good sample for the average user with regard to bandwidth and uptime. Since both influence the degree of a node, we only considered the sets $N_k(m) \setminus M$ of neighbors of $m \in M$ at time $t \cdot k$. Let $deg(v)$ denote the degree of a node v. Analogously to the distance distribution, the empirical degree distribution of neighbors $\hat{D}'$ was then obtained as [7]

$$P(\hat{D}' = x) = \sum_{k=0}^{K} \sum_{m \in M} \sum_{v \in N_k(m) \setminus M} \frac{\mathbf{1}_x(deg(v))}{\sum_{k=0}^{K} \sum_{m \in M} |N_k(m)|}. \tag{2}$$

Then, note the probability of being a neighbor of a node is proportional to the degree of a node. If the degree distribution of the network is D, the degree distribution D' of randomly chosen neighbors is given by

$$P(D' = x) = \frac{xP(D = x)}{\mathbb{E}(D)}. \tag{3}$$

Our measurements provided the empirical degree distribution of neighbors $\hat{D}'$. So an empirical degree distribution $\hat{D}$ was obtained by solving a system of linear equations based on Eq. 3. Let d_m denote the maximal observed degree. The system of linear equations consisted of $d_m + 1$ equations with $d_m + 1$ variables $P(\hat{D} = x)$ for $x = 1 \dots d_m$ and $\mathbb{E}(\hat{D})$. The first d_m equations were derived from transforming Eq. 3 to $xP(D = x) - P(D' = x)\mathbb{E}(D) = 0$. The last equation used that $\hat{D}$ is a probability distribution, so that $\sum_{x=1}^{d_m} P(\hat{D} = x) = 1$. The system of equations thus could be solved using Gaussian elimination.

Darknet : In order to evaluate the impact of small Darknets with few links into the Opennet, we manually created a Darknet topology consisting of 10 nodes. These nodes were connected in a ring topology of which 4 nodes established a connection to a monitoring node m that participated in the Opennet. The node m logs all file requests and the corresponding responses that pass through

[7] It is intended that nodes in the intersection of two neighborhoods are counted multiple times in order to obtain $\hat{D}$ from $\hat{D}'$

it. Based on the logs, we then distinguish between requests forwarded into the Opennet by m and requests forwarded into the Darknet. The difference of the success rate between forwarding to Opennet and to Darknet nodes then indicates the impact of such small Darknets.

Network Size and User Origin : We logged Freenet locations, IP addresses and ports of the Opennet neighbors of monitoring nodes. Each Opennet node is uniquely characterized by a persistent location, in contrast to Darknet nodes, which change location in order to adapt to the topology. For the Opennet, we hence uniquely identify Freenet instances by their location. Note that a user participating with multiple instances is counted several times. In contrast to the location, the IP address of a user changes over time. Furthermore, a Freenet node might advertise several IP port combinations. We logged the IP address only for obtaining the geolocation of users, not as an identifying feature.

Popularity Analysis : All requests for files seen by a monitoring node were logged, in particular the routing key of each file. We then obtained a popularity score for a key k by dividing the number of requests for k by the total number of requests.

3.2 Active Monitoring

We used active monitoring for tracking the online times of nodes. In the active mode, monitoring nodes periodically sent messages into the network to determine if a certain node is online. This approach allowed us to determine to what extend it is possible to track a user's online time in Freenet. Also, we established a churn model for Freenet users including session length, intersession length, and connectivity factor.

Up to September 2012, using messages of type *FNPRoutedPing* allowed us to query for nodes by their location. The message is routed through the network like any normal request. If a node with the specified location is found, a reply is sent back to the requester. From September 2012, information about nodes outside of the second neighborhood could only be obtained by using the *FNPRHProbeRequest*. As a reply to this message, one specified information, e.g. the location or the uptime, about a random node from the network is returned. The node is chosen by executing an random walk with Metropolis-Hastings correction for 18 hops, so that every node should be selected close to uniformly at random [8]. Note that the message type *FNPRoutedPing* clearly allowed tracking of nodes, whereas *FNPRHProbeRequest* abolishes the possibility to query for a specific node. Hence, we also show that tracking is possible with *FNPRH-ProbeRequest*, a message that is still supported by the current Freenet version (*1459*).

In both approaches, we estimated the session starts $S(u)$ and endpoints $E(u)$ of a node u based on our measurements. From these sets, we characterized churn behavior as follows: Let $s_j(u)$ and $e_j(u)$ denote the j-th smallest element in $S(u)$

[8] `https://wiki.freenetproject.org/index.php?title=FCPv2/ProbeRequest`

and $E(u)$, respectively. The total time of the measurement was T. The length of the j-th session of node u was then computed as $sess_j(u) = e_j(u) - s_j(u)$ given that u is online for at least j sessions. Similarly, the j-th intersession length was computed as $inter_j(u) = s_{j+1}(u) - e_j(u)$. Session and intersession length provide information on the reliability of nodes and the amount of maintenance required to keep the structure of the network intact. The connectivity factor of a node u is then defined as the fraction of time u was online, i.e., $conn(u) = \frac{\sum_{j=1}^{|S(u)|} sess_j(u)}{T}$. The connectivity factor is decisive for determining how often a file is available at a node. Moreover, we analyzed the number of nodes in the network to see if there are diurnal patterns. The fraction of online nodes for each point in time t and set of observed nodes Q are given by $f(t) = \frac{|\{u \in Q : \exists j : s_j(u) \leq t, e_j(u) \geq t\}|}{|Q|}$.

Using *FNPRoutedPing* The methodology using *FNPRoutedPing* was to first collect locations of nodes and then ping each of those locations every X timeunits. However, pings are routed within the Freenet network and are thus not guaranteed to find a node even if it is online. We solved this problem by pinging a node multiple times from different monitoring nodes. The maximal number of pings per node was chosen empirically such that the probability that a node would answer at least one of our pings was found to be sufficiently high.

We hence conducted the measurement as follows: First, we distributed our monitoring M equally in the key space, i.e., at locations $i/|M|$ for $i = 0 \ldots |M| - 1$. We divided n nodes to ping in sets of size $n/|M|$. Every X timeunits, each monitoring node pinged $n/|M|$ nodes and reported to a central server, which nodes had answered the requests. Nodes that had not been found were rescheduled to be pinged by a different monitoring node. After a node had been unsuccessfully pinged by k monitors, it was considered to be offline. k was chosen empirical by pinging our own monitoring and choosing k such that an online node would be detected with probability at least p[9]. We obtained the session starts and ends from the logged data as follows: The total time of our measurement was divided into K intervals $I_1, \ldots, I_K$ of length X. For any node u, we determined a sequence of boolean values $on_0(u), on_1(u), \ldots, on_K(u), on_{K+1}(u)$, so that $on_i(u)$ is true if u has been detected in interval $i = 1 \ldots K$ and $on_i(u) = false$ for $i = 0, K + 1$. Then $S(u)$ consisted of the start times of all intervals in which u was discovered but has not been discovered in the proceeding interval, i.e., $S(u) = \{(i-1)X : i \in \{1, \ldots K\}, on_i(u) = true, on_{i-1} = false\}$. Analogously, $E(u) = \{iX : i \in \{1, \ldots K\}, on_i(u) = true, on_{i+1} = false\}$.

Using *FNPRHProbeRequest* The methodology using *FNPRHProbeRequest* was to send a large number of requests for node locations into the network from different locations and gather all replies together with a timestamp. A node was considered offline if no reply from it had been received for at least time τ.

[9] We are aware that the estimation is only valid under the assumption that our monitoring nodes are representative for all nodes.

More precisely, we obtained an ordered set $R(u) = \{r_1(u), \ldots, r_{|R(u)|(u)}\}$ with $r_i(u) \in [0, T]$ of reply dates for each user/location u. The start of a session was assumed to be the first time a node had replied after not replying for τ timeunits, i.e., $S(u) = \{r_i(u) \in R(u) : i = 1 \text{ or } r_i(u) - r_{i-1}(u) \geq \tau\}$. Analogously, the end of a session was defined as the point in time of the last received reply $E(u) = \{r_i(u) \in R(u) : i = |R(u)| \text{ or } r_{i+1}(u) - r_i(u) \geq \tau\}$. For choosing a suitable value for τ, let req be the number of answered requests per time unit. Assuming that indeed all nodes are selected with equal probability, the probability that a node does not respond to any of the $req \cdot \tau(p)$ requests is given by

$$1 - p = (1 - 1/n)^{req \cdot \tau(p)} \tag{4}$$

for a network of n nodes. $p \in \{0.9, 0.925, 0.95, 0.975, 0.99, 0.999\}$ was used. A low p indicates that the probability to accidentally cut one session into multiple session is high, in particular for long sessions. With increasing p, the probability to merge multiple sessions into one increases as well.

3.3 Data Set and Privacy

Our research was conducted in agreement with the German Federal Data Protection Act (in particular §28 and §40). In order to protect the privacy of Freenet's users, we carefully made sure to erase all identifying information from our collected data after computing the necessary statistics. The collected IP addresses were the potential link between Freenet users and their real-world identity. Note that the IP addresses were only required for obtaining the geolocation and the count of diverse IPs, and were deleted afterwards. We did not record the IP address in our database for all remaining measurements, in particular the tracking of users was done solely based on their Freenet location, which is unrelated to the real-world identity. The recorded data is available upon request.

4 Topology Characteristics

In this Section, we present the results regarding the distance and degree distribution of the Opennet. Using simulations, we then show that Freenet's current ID selection fails to provide the desired routing performance. Finally, we discuss the impact that separate Darknets attached to the main Opennet topology have on the routing quality of the overall system.

4.1 Distance and Degree Distribution

The number of hops, also called the routing length, needed to discover a file is essential for the performance of a P2P system. It is mainly influenced by the number of neighbors a node has and the locations of these neighbors in the key space.

The distance distribution between neighbors is supposed to be close to Kleinberg's model. However, nodes connect to those answering requests independently of their location, so that we would rather expect the distance between neighbors to be distributed uniformly at random. The degree distribution is directly related to the bandwidth of the nodes, i.e., a higher degree should correspond to a high bandwidth. The degree distribution of neighbors is expected to show nodes with a degree above average, since they are more likely to be selected as neighbors.

Setup: The data for this analysis was obtained from a two week measurement in May 2013 using 12 instrumented Freenet clients.

Results: Figure 1a shows the cumulative distance distribution observed in our measurements in comparison to the function $1/d$ for $d > 0.01$. Indeed, each node had a high number of close neighbors. However, contacts at distance exceeding 0.05 seemed to be chosen uniformly at random, as indicated by the linear increase of the distribution function.

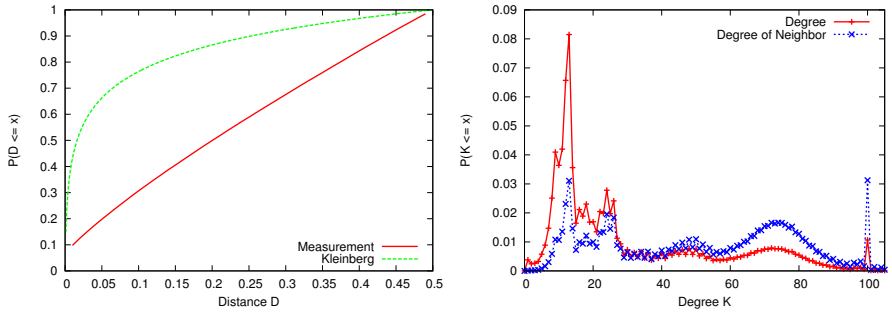
With regard to the degree distribution, there are several peaks in the degree distribution around 13, 50, 75 and 100 (cf. Figure 1b). Indeed, these seem to correspond to typical bandwidth, e.g. for 2 Mbit/s 100 neighbors are allowed. Note that we observed nodes with a degree of up to 800, but nodes with a degree of more than 100 make up less than 1 %. Nodes with a degree of less than 10 are likely to be in the start-up phase since by default a node is allowed at least 14 neighbors.

Discussion: We have seen that nodes have a high number of close neighbors. These are probably found by announcements sent via the seed nodes and routed towards a node's own location. However, the long-range contacts are chosen uniformly at random, i.e., with a probability proportional to $\frac{1}{d^0}$ rather than with probability of $\frac{1}{d^1}$. The routing cost when nodes are connected independently of their locations is of order $n^{2/3}$ [6].

4.2 Simulation study of Freenet's routing performance

To illustrate the impact of our previous derivation, we performed a simulation study of the Freenet routing algorithm.

Setup: We generated a ring topology with $15,000$ nodes corresponding to the network size estimated in Section 5.1. Each node was assigned a random location in $[0, 1)$, corresponding to Freenet's key space. Each node was connected to the k closest nodes on the ring. In addition, for each node a random integer l was chosen according to the empirical degree distribution we observed in the Freenet network. The node was then given $d = max\{l-2k, 0\}$ long-range contacts chosen proportional to $1/d^r$ for $r = 0$ (independent of the distance as in Freenet) and $r = 1$ (anti-proportional to the distance suggested by Kleinberg).



(a) Distance (b) Degree

Fig. 1: Distance Distribution of neighbors, Degree Distribution, and the Degree Distribution of neighbors

Reults: The average routing length was less than 13 hops for an optimal distance distribution ($r = 1$), but 37.17 hops for $r = 0$, i.e., the distance distribution we found in Freenet. When connecting each node to the 3 closest nodes on the ring, i.e., $k = 3$, the average routing length for $r = 0$ decreased to 28 because progress was made using the additional short-range links, but the average routing length for $r = 1$ increased by 30% to 17 hops. These results show that Freenet's performance can be drastically improved by, e.g., dropping and adding connections based on the distance of node identifiers. A Kademlia-like bucket system [24] could be used to achieve the desired distance distribution while still allowing a wide choice of neighbors. So, the decision of dropping a neighbor can be made both on its performance and its location. The number of buckets of the number of contacts per bucket and hence the degree can be chosen dependent on the bandwidth a node contributes to the system, in order to retain this incentive of the current neighbor selection scheme. An alternative approach can be to include Opennet in the location swapping algorithm used by Darknet nodes, which has been shown to achieve a Kleinberg-like distance distribution in [5] for a static network. An in-depth simulation study is required to give concrete guidelines.

4.3 Darknet

We expected that requests forwarded into the Darknet would fail more frequently because the Opennet node responsible for the requested key is not topologically close to Darknet nodes with similar locations.

Setup: The measurement was conducted for a duration of 140 hours in April 2014. We manually set up a small Darknet consisting of 10 nodes and connected two of these nodes to one monitoring node in the Opennet.

Results: In total, the monitoring node received $3,540,000$ requests and forwarded 47.94% into the Darknetnet. While 8.46% of the requests forwarded into the

Opennet were successful, only 0.08% of the Darknet requests returned the requested resource. Overall, only 4.4% of the requests forwarded by the monitor were successful.

Discussion: The performance decrease only considers requests forwarded via our monitoring node, and thus the impact of one small Darknet on the overall performance is low. However, we have seen that forwarding messages into the Darknet can clearly decrease the success rates if Darknet and Opennet are only connected by one link. If such Darknets exist in large numbers, they might be partly responsible for low success rate of Freenet routing. Including Opennet nodes into the location swapping can potentially solve the problem of parallel ID spaces, but as stated a detailed study is needed to show if the overall performance is actually improved.

5 User Behavior

In this Section, we present the results of our measurements in Freenet concerning the actual network size, origin of nodes, churn behavior and file popularity.

5.1 Network Size and Origin

We expected to discover a few thousand of concurrently online nodes, as observed in earlier measurements [4]. As the main goal of Freenet is to provide censorship-resilience, we also expected to find users from countries where either Internet censorship is applied or at least heavily discussed. While in the first case, services such as Tor [25] or Freenet are needed to retrieve the desired content, the use of anonymous and censorship-resilient communication might be increased due to a heightened awareness of potential privacy breaches in the second case.

Setup: Our measurements were conducted for 8 weeks in June to August 2012 using 55 instrumented Freenet clients.

Results: During the eight week measurement period, we observed a total of $58,571$ unique locations. The number of distinct IP addresses was $102,376$. Most locations were discovered during the first two weeks, afterwards only one or two new locations were found most days. On some days, however, several tens of new locations were discovered within one hour. The sudden increase was probably due to measurement activities by other institutions. Excluding these bursts, we see a convergence in the number of discovered locations, indicating that we were aware of most active Freenet clients. The observed difference between the number of locations and IPs is explained by the frequent use of non-static IPs. While the increase in discovered IPs is largest in the first days, the numbers grow constantly throughout the measurement, as can be expected if active users regularly change their IP. In addition, nodes can advertise more than one IP address at a time. Whereas the majority of nodes (84.4%) had only a single IP

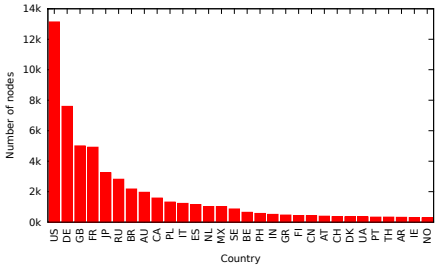


Fig. 2: Distribution of Freenet nodes over countries

p	$\tau(p)$	$\theta(q_i(p))$: mean,min,max
0.900	3:27	0.993,0.989,0.996
0.925	3:53	0.993,0.989,0.996
0.950	4:29	0.992,0.989,0.995
0.975	5:31	0.991,0.987,0.994
0.990	6:54	0.989,0.983,0.993
0.999	10:22	0.984,0.979,0.989

Fig. 3: FNProbeRequest Statistics: Time $\tau(p)$ without reply until a node is declared offline, and the estimation $q_i(p)$ of detecting an online node

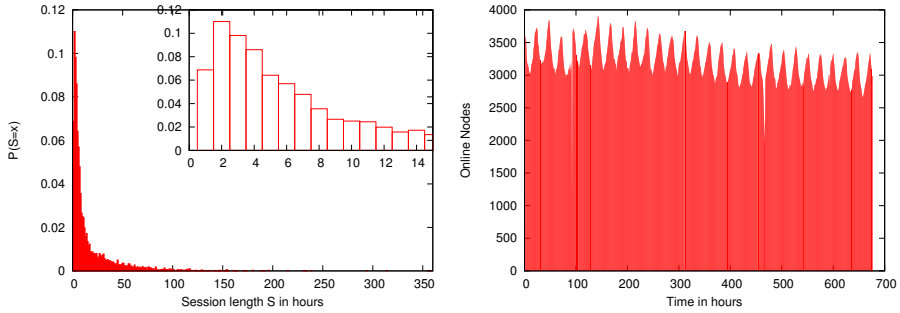
address over the whole period, about 10% advertised 2 and 3.6% 3 different IPs. On a closer look, nodes with more than 10 IP addresses were commonly located at universities, but also at the Tor proxy network *TKTOR-NET*, indicating that some users aim to hide their IP address in the Opennet by using Tor. At the time of the measurement, TKTOR-NET provided three exit nodes that participated in Freenet. IPs from various anonymous VPN were discovered as well. The discovered nodes were mainly traced back to Europe and North America, as can be seen in Figure 2. Nearly a quarter of the discovered installations were located in the USA, an eighth in Germany. Together with France and Great Britain, these countries made up more than half of all encountered nodes.

Discussion: Our results show that Freenet is widely used. We discovered close to 60,000 active Freenet installations. So there clearly is demand for privacy-preserving communication and publication. Nevertheless, the typical Opennet user does not seem to be located in countries typically associated with Internet censorship. However, our study does not shed light on Darknet and Tor users.

5.2 Churn

In this Section, we discuss and compare the results for the two methods to measure churn behavior in Freenet introduced in Section 3.2. In all measurement studies of file-sharing systems, very short medium session length of less than 1 hour were observed. We expected to see such short sessions as well, corresponding to down- or uploads of one specific data item, especially if the content is sensitive and online times are short to minimize the risk of capture. However, Freenet users are advised to leave their clients running for at least 24 hours, so that we expected a comparable high fraction of long session as well. For both measurements, we first state the set-up and the results, but leave the discussion until the end of this subsection. In addition, we shortly discuss both the accuracy of our measurement as well as the additional load on the network created by the measurement.

Setup: The first measurement study was used to analyze the long-term behavior of a large set of nodes over more than a month, identifying daily and weekly patterns. The second measurement was needed because nodes were not contacted



(a) Session Length (b) Nodes Online

Fig. 4: Churn characteristics using *FNPRoutedPing* for node discovery

frequently enough to provide an accurate description of the session length distribution. The differences in the methodology were due to a change in the Freenet code between the first and the second measurement, which abolished the *FNPRoutedPing* message used for locating specific nodes.

Using *FNPRoutedPing* We performed the measurements querying every node at most $k = 5$ times. In order to observe the long-term behavior of nodes, the measurement period was chosen to be $X = 1h$. The value of k was chosen, such that our own nodes replied with a probability of 99.9%. The measurements were executed over a period of 28 days in August and September 2012 using 55 instrumented Freenet clients.

Results: The session length distribution is shown in Figure 4a, using bins of 1 hours in agreement with our measurement period. The majority of session lasted less than two hours, only 1.7% of the sessions lasted longer than 100 hours. The longest observed session was 357 hours. Note that there was a drop in the session length at about 8 and 17 hours, most probably because some nodes are only online during certain parts of the day.

The inter-session time follows a similar distribution: Roughly 10% of the inter-sessions are between 1 and 2 hours. Potential reasons are the missed probing due to the probabilistic nature of the measurements, crashes, and short-time connectivity breaks, e.g., when moving a laptop from home to work. Furthermore, there is a peak at the about 8 hours, in agreement with the corresponding peak of session length of roughly 16-17 hours. The results indicate that some users only run their clients during the day. The average connectivity factor of all nodes was rather high, namely 0.19.

The average number of discovered nodes was $3,207$ of the $15,503$ pinged nodes. The number of discovered nodes over time can be seen in Figure 4b. Diurnal patterns can be clearly identified. There was a maximum in the number of users at 10 PM CEST and a minimum at 10 AM CEST. In general, the

number of online nodes in our sample varied between $2,500$ and $3,600$. So the network size changed periodically, but not drastically.

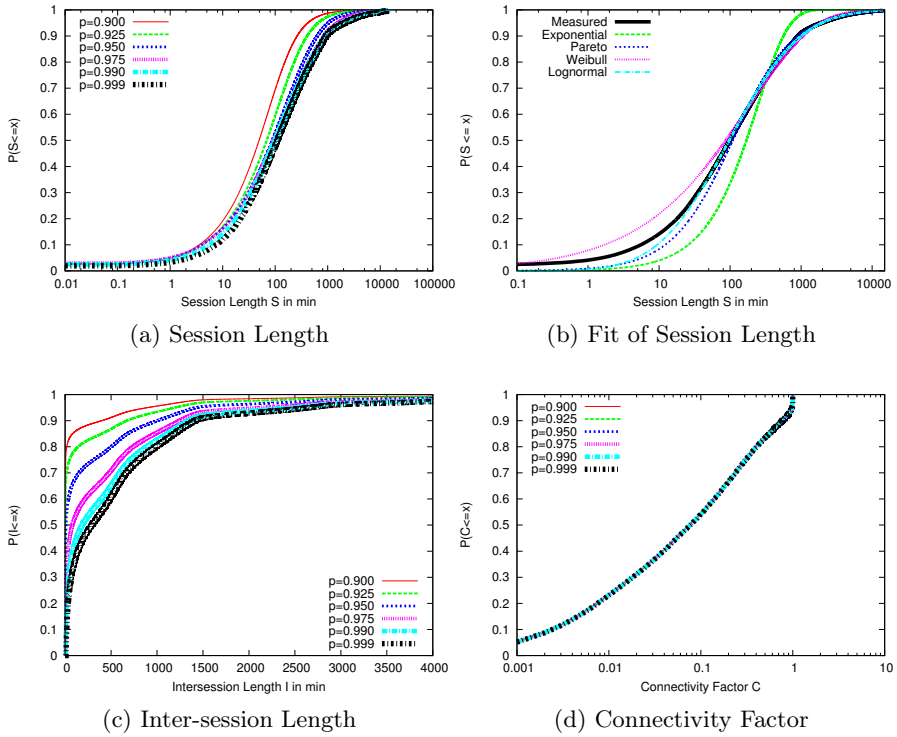
Accuracy and Load: The session length is only estimated within an accuracy of $2X = 2$ hours, hence we only considered the long term behavior in this measurement. Note that the results represent a lower bound on the fraction of long session because nodes can be accidentally declared offline during a session. As for the measurement cost, we found that without an measurement, a Freenet node forwarded on average around $13,000$ file requests and replies per hour, not considering maintenance costs. The average maintenance traffic produced by our measurement was less than 500 messages per node per hour.

Using *FNPProbeRequest* The measurement was conducted in November 2013 over a period of 9 days using 150 instrumented clients. We varied p, the lower bound on the probability that an online node replies within a time $\tau(p)$, between 0.9, 0.925, 0.95, 0.975, 0.99, and 0.999 as described in Section 3.2. Our monitoring nodes received at least $req = 10,000$ replies per minute. Choosing $\tau(p)$ according to Eq. 4 with an estimate of $n = 15,000$ resulted in intervals of roughly 3 ($p = 0.9$) to 10 ($p = 0.99$) minutes as can be seen in Table 3. Note that p is a lower bound on the probability to discover a node since we consider a lower bound on req and an upper bound on n.

Results: The median session length of the second measurement was between 49 to 110 minutes, depending on p. In particular, the median session lengths for $p = 0.975$ and $p = 0.99$ were 95 and 99 minutes, respectively. The distribution of the session length is shown in Figure 5a. We fitted the distribution to the most commonly used models for the session length (e.g., [21]), in order to see if they provide adequate accuracy to be used as models of Freenet user behavior in simulations. The non-linear least square fit function in R[10] was used to fit the distribution for $p = 0.99$: an exponential distribution with cdf $1 - exp(-ax)$ for $a = 4.086 \cdot 10^{-3}$, a shifted Pareto distribution $1 - (1 + x/b)^{-a}$ for $a = 1.054$ and $b = 116.3$, a Weibull distribution $1 - exp(-(b * x)^a)$ for $a = 0.4788$ and $b = 5.355 \cdot 10^{-3}$, and a lognormal distribution $\Phi((log(x) - a)/b)$ for $a = 4.5773610$ and $b = 1.8235325$ with Φ denoting the cumulative normal distribution. The residual errors were minimized for the Weibull distribution (about $8 \cdot 10^{-3}$). However, the lognormal distribution also achieved an residual error of only 0.019. The error of the lognormal distribution is mostly due to its underestimation of the fraction of short sessions, as can be seen from Figure 5b. Since the session length was underestimated by our measurement methodology in general, the error is acceptable and can be seen as a correction. The fitted Weibull distribution, on the other hand, overestimated the fraction of short sessions, while the exponential and Pareto distribution did not model the shape of the distribution accurately.

The distribution of the inter-session length is displayed in Figure 5c. The median inter-session length varied greatly between less than 10 minutes ($p = 0.9$)

[10] `http://stat.ethz.ch/R-manual/R-patched/library/stats/html/nls.html`



(a) Session Length (b) Fit of Session Length

(c) Inter-session Length (d) Connectivity Factor

Fig. 5: Session length for a) all considered p, and b) $p = 0.99$ fitted to common session length models, c) inter-session length, and d) connectivity factor

and close to 6 hours ($p = 0.999$). All distributions show a strong increase in the distribution function of the inter-session length at roughly 8 to 10 hours as well as at roughly 16 to 17, indicating that a lot of users only run their clients during certain hours of the day. Due to these spikes, the inter-session length could not be fit to any of the standard models. The distribution of the connectivity factor, displayed in Figure 5d, shows that most users were online during a small fraction of the measurement, but also more than 5% of the users have a connectivity factor of nearly 1. Note that in contrast to the session length, the results for the connectivity factor are very close for all p, due to the fact that the overall online time is not largely influenced by splitting one session into multiple sessions. The average connectivity factor is around 0.22.

Accuracy and Load: We show that indeed our method selected nodes uniformly at random, and captured more than 98% of all online nodes. As stated in Section 3, assuming that the *htl* counter is set high enough, all nodes should reply with roughly equal probability. In particular, the number of requests answered by our monitoring nodes should be approximately normal distributed. We performed a Kolmogorov-Smirnoff test, which indicates a normal distribution (p-value of roughly 0.06). So nodes seem to be selected uniformly at random, which allowed

us to obtain a lower bound on the probability of detecting an online node as follows. The size of a static network can be estimated by performing two samples and considering the size of their intersection [26]. Note that in a dynamic network only a lower bound is obtained since the population changes in consecutive intervals and the intersection consists of at most all nodes online in both intervals. We split the measurement period into intervals of length $\tau(p)$, and determined the sample A_i of all nodes responding to a probe in interval i. We then computed the fraction of the intersection $f_i = \frac{|A_i \cap A_{i+1}|}{|A_i \cup A_{i+1}|}$. For the probability q_i to sample a node during interval i. The probability that a node is sampled in interval i and $i+1$ is $q_i q_{i+1}$, and the probability that it is sampled in at least one interval is $1-(1-q_i)(1-q_i q_{i+1})$. For a static network and constant q_i, the expected value of f_i would be $\mathbb{E}(f_i) = \frac{q_i^2}{1-(1-q_i)^2}$. We hence obtained an unbiased estimate $\theta(q_i) = \frac{2f_i}{1+f_i}$ by transforming $f_i = \frac{q_i^2}{1-(1-q_i)^2}$. The values computed for mean,minimal, and maximum $\theta(q_i)$ over all intervals exceed 0.98 (but for the minimum in case of $p = 0.999$ as displayed in Table 3), so that indeed we captured the majority of online nodes per interval. For long intervals $\tau(p)$, the estimate on the accuracy decreases below p since the changes in the population outweighed the improved accuracy of an increased number of probes. However, the probability to be detected in every interval decreases exponentially with the session length and the reciprocal of interval length $\tau(p)$. For a probability of 0.98 to detect a node, the chance to be accidentally declared offline during 1 hour (more than 15 times $\tau(p)$) is still close to 30 % for $p = 0.9$ and $p = 0.95$, explaining the short median session length for low values of p and the high number of short intersessions of less than 10 minutes. Hence, the higher values

The overhead produced by FNPRoutedPing is about 2000 messages per hour, which makes up a noticeable but not large fraction of the roughly $13,000$ requests and replies that need to be processed normally.

Discussion: We conducted two measurements. The first one was a long-term measurement over more than 4 weeks, in order to find diurnal and weekly pattern. We found that the fraction of long sessions was considerably higher in Freenet than in BitTorrent. Pouwelse [19] found that at most 3.8% of BitTorrent users stay longer than 10 hours and only 0.34% longer than 100 hours. In comparison, we observed close to 2% of sessions lasting longer than 100 hours. We clearly observed diurnal patterns, though they are not as distinct as in other applications, such as in Facebook [27]. The second measurement study was conducted to obtain more fine-grained results on the session and inter-session length, in order to evaluate the applicability of common churn models used in simulators. We discovered that the session length is reasonably well modeled by lognormal or Pareto distributions, but not by a Weibull or exponential distribution. In contrast, Stutzbach's results from 2006 indicate that churn in structured P2P systems is well modeled by lognormal and Weibull distributions [21]. The median session length was 4 hours in the first measurement, but less than 2 hours in the second measurement. Potential reasons are the high inaccuracy of the first

measurement. For example, a session length of slightly more than 2 hours can accidentally be declared as 3 hours. Furthermore, nodes are only pinged every hour, so that short inter-sessions can be missed. However, both measurements indicate a longer median session length than the 1 to 60 minutes observed in Napster [15], Gnutella [15], FastTrack [16], Overnet [17], Bittorrent [19], and KAD [20, 21]. The inter-session length could not be modeled by commonly used distributions such as Pareto, because both measurements exhibited local maxima at about 8 and 16 hours. Such behavior has not been remarked in the related work, to the best of our knowledge. In summary, our results indicate that Freenet users are online longer than users of common file-sharing applications. Furthermore, clear diurnal patterns can be observed by considering the number of online nodes and the inter-session length.

An ulterior result of the churn analysis is that the online time of nodes can be reliably tracked, even without the possibility to ping a specific node. In this measurement, we only tracked the nodes by their location. However, locations of Opennet nodes can be mapped to IP addresses by inserting monitoring nodes in the system and tracking the location and IP of neighbors as presented in Section 5.1. The knowledge of online time now enables intersection attacks on the anonymity [28]. As a consequence, the seemingly harmless *FNPProbeRequest*, which returns information of a random node in the network, can potentially be abused for harming the anonymity. Because the focus of our study was the efficiency rather than the security of the system, we did not perform a detailed study on the potential damage. However, the reliability in tracking our own nodes indicates that *FNPProbeRequest* should be removed from the set of Freenet's functionalities. It mainly seems to be used by Freenet developers to obtain statistics about the network, but as seen above, the data is poorly anonymized and can be potentially abused.

5.3 File Popularity, User Activity, and Content

The popularity of files in file-sharing systems is assumed to be Zipf-distributed, i.e., the majority of requests address a small number of files. In contrast to P2P-based content distribution systems, Freenet provides the storage and retrieval of Freesites and blogs, which are clearly different from regular popular media. Hence, it is unclear if the aforementioned properties also hold for Freenet.

Setup: The measurement was conducted in Autumn 2012 using 11 instrumented Freenet clients. Their locations were chosen uniformly at random.

Results: During the measurement, we logged several hundred thousands of file requests. The 1,000 most popular files all received more than 21,000 requests, indicating that the majority of regular Freenet users requested those files. Our results indicate a Zipf-distribution for file popularity in agreement with the results on BitTorrent [20, 29]. The most popular file accounts for 0.73% of seen requests, the second most popular file only for 0.45%. The 30-th popular file only accounts for 0.25% of the requests. Hence, after the fast decrease in popularity for the first files, the decrease is then slower and steadier.

Discussion: Our analysis of file popularity and user activity mostly agrees with the common assumptions. There are few very popular files, and the majority of the files is not requested frequently. Similarly, most files are published by a small set of users. We did not fit the popularity distribution, since local caching of popular files is bound to reduce the number of actually observed requests for popular files in comparison to less popular files. Consequently, our measurements underestimate the popularity of popular files, and the actual numbers are not reliable. However, the existence of a Zipf-like distribution can be assumed from the results, even if the actual shape of the distribution is skewed. Hence, the Least-Recently-Seen caching used in Freenet and designed for such popularity distributions should be very effective.

6 Conclusion

We showed how to conduct measurements in Freenet despite its obfuscation protocols. The results verify that the routing in Freenet is insufficient with regard to the neighbor selection and the interaction between Opennet and Darknet. Furthermore, we obtained a realistic churn model of Freenet users. In the future, we aim to evaluate our proposed neighbor selection and routing algorithms in a trace-driven simulation model based on the user behavior measurements and integrate them into the Freenet client code.

Acknowledgments

We thank Jan-Michael Heller and Christina Heider for their help in conducting the measurements, and Rob Jansen and the anonymous reviewers for their valuable comments.

References

1. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Workshop on Design Issues in Anonymity and Unobservability*, 2000.
2. Ian Clarke, Theodore W. Hong, Scott G. Miller, Oskar Sandberg, and Brandon Wiley. Protecting free expression online with freenet. *IEEE Internet Computing*, 2002.
3. Ian Clarke, Oskar Sandberg, Matthew Toseland, and Vilhelm Verendel. Private communication through a network of trusted connections: The dark freenet. 2010.
4. Eugene Y. Vasserman, Rob Jansen, James Tyra, Nicholas Hopper, and Yongdae Kim. Membership-concealing overlay networks. In *CCS*, 2009.
5. Oskar Sandberg. Distributed routing in small-world networks. In *Workshop on Algorithm Engineering and Experiments (ALENEX06)*, 2006.
6. Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Symposium on Theory of Computing*, 2000.
7. Stefanie Roos and Thorsten Strufe. A contribution to darknet routing. In *INFOCOM*, 2013.

8. Nathan S. Evans, Chris GauthierDickey, and Christian Grothoff. Routing in the dark: Pitch black. In *ACSAC*, 2007.
9. Benjamin Schiller, Stefanie Roos, Andreas Hoefer, and Thorsten Strufe. Attack resistant network embeddings for darknets. In *SRDSW*, 2011.
10. Guanyu Tian, Zhenhai Duan, Todd Baumeister, and Yingfei Dong. A traceback attack on freenet. In *INFOCOM*, 2013.
11. Curt Cramer, Kendy Kutzner, and Thomas Fuhrmann. Bootstrapping locality-aware p2p networks. In *ICON*, 2004.
12. Prateek Mittal and Nikita Borisov. Shadowwalker: peer-to-peer anonymous communication using redundant structured topologies. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 161–172. ACM, 2009.
13. Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas E. Anderson. Privacy-preserving p2p data sharing with oneswarm. In *SIGCOMM*, 2010.
14. Prateek Mittal, Matthew Caesar, and Nikita Borisov. X-vine: Secure and pseudonymous routing using social networks. *arXiv preprint arXiv:1109.0971*, 2011.
15. P. Krishna Gummadi, Stefan Saroiu, and Steven D. Gribble. A measurement study of napster and gnutella as examples of peer-to-peer file sharing systems. *Computer Communication Review*, 2002.
16. Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. *Networking, IEEE/ACM Transactions on*, 2004.
17. Ranjita Bhagwan, Stefan Savage, and Geoffrey M. Voelker. Understanding availability. In *IPTPS*, 2003.
18. Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *IMC*, 2005.
19. Johan A. Pouwelse, Pawel Garbacki, Dick H. J. Epema, and Henk J. Sips. The bittorrent p2p file-sharing system: Measurements and analysis. In *IPTPS*, 2005.
20. P. Krishna Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *SOSP*, 2003.
21. Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC*, 2006.
22. Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a dht. *Computer Science*, 2003.
23. Fabian Bustamante and Yi Qiao. Friendships that last: Peer lifespan and its role in p2p protocols. *Web Content Caching and Distribution*, pages 233–246, 2004.
24. Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Peer-to-Peer Systems*, 2002.
25. Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.
26. Sandeep Mane, Sandeep Mopuru, Kriti Mehra, and Jaideep Srivastava. Network size estimation in a peer-to-peer network. *University of Minnesota, MN, Tech. Rep*, 2005.
27. Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. Understanding online social network usage from a network perspective. In *IMC*, 2009.
28. David Isaac Wolinsky, Ewa Syta, and Bryan Ford. Hang with your buddies to resist intersection attacks. In *CCS*, 2013.
29. Mohamed Hefeeda and Osama Saleh. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Trans. Netw.*, 2008.