hal-00744919, version 1 - 24 Oct 2012

# A Bird's Eye View on the I2P Anonymous File-sharing Environment

Juan Pablo Timpanaro, Isabelle Chrisment*, Olivier Festor

INRIA Nancy-Grand Est, France
*LORIA - ESIAL, Université de Lorraine
Email: {juanpablo.timpanaro, olivier.festor}@inria.fr
Email: {isabelle.chrisment}@loria.fr

**Abstract.** Anonymous communications have been gaining more and more interest from Internet users as privacy and anonymity problems have emerged. Among anonymous enabled services, anonymous file-sharing is one of the most active one and is increasingly growing. Large scale monitoring on these systems allows us to grasp how they behave, which type of data is shared among users, the overall behaviour in the system. But *does large scale monitoring jeopardize the system anonymity?*
In this work we present the first large scale monitoring architecture and experiments on the I2P network, a low-latency message-oriented anonymous network. We characterize the file-sharing environment within I2P, and evaluate if this monitoring affects the anonymity provided by the network.
We show that most activities within the network are file-sharing oriented, along with anonymous web-hosting. We assess the wide geographical location of nodes and network popularity. We also demonstrate that group-based profiling is feasible on this particular network.

**Keywords:** Large scale monitoring, I2P, Security risks, Anonymous file-sharing

## 1 Introduction

Anonymous communications have been acquiring more and more interest since the past decade, either for fighting against any type of censorship, passive attacks (third-parties sniffing, traffic analysis, user profiling) or for malicious purposes (copyrighted material downloads). Within anonymous communications, anonymous file-sharing is one of the most active fields and is increasingly growing, partially due to the onrush of negative news on public file-sharing, including legal actions by governmental institutions, law-enforcement agencies and movie maker associations to major file-sharing communities, and partially because of the rising concern of both privacy and anonymity concepts within the Internet.

Large scale monitoring on file-sharing communities provides a wide view of the network[1][2][3] and allows us to answer the following questions: *what kind of content is mainly distributed? How many users does the network have? How many files does the network hold? Which users are downloading a given content?*

hal-00744919, version 1 - 24 Oct 2012

However, large scale monitoring on anonymous environments has not been widely investigated, and is mainly focused on the Tor network [4][5]. Anonymous systems often imply a decoupling between the identity of users and their activities within the system, hardening the monitoring. Among anonymous file-sharing systems, this means that it becomes very challenging to successfully link together a specific user with a specific download.

The I2P[6] network is a low-latency message-oriented anonymous network designed as a network layer, in which any two users can communicate among themselves in completely anonymous manner. This network has a full range of available applications: anonymous web-browsing, chatting, file-sharing, web-hosting, e-mail and blogging among others. Except for anonymous web-browsing that necessarily requires an out-proxy to the normal Internet, the rest of the mentioned applications interact between each other within the network boundaries.

In this paper, we provide an efficient method to monitor the I2P anonymous file-sharing community. We perform an analysis of the I2P network based on intense monitoring, which to the best of our knowledge is the first large scale monitoring effort of the network, answering the following questions: *is it possible to properly characterize the I2P anonymous file-sharing system? Does the anonymity provided by the I2P network get compromised by large scale monitoring? Does large scale monitoring introduce new security risks?*. Our goal is to properly characterize the I2P file-sharing environment, and assess if a large scale monitoring activity jeopardizes the anonymity of the network.

This paper is organized as follows: Section 2 introduces the I2P network and its main components and features, including used encryption techniques, peer profiling and its distributed database. Section 3 describes our monitoring architecture. We detail how we use I2P's distributed database to collect data and how we use this data to characterize the file-sharing side of the network. An experiment is presented in section 4, in which we deploy our monitoring architecture over several days. Section 5 points out previous and current work on anonymous file-sharing and large scale monitoring. Finally, Section 6 concludes this work.

## 2 The I2P network

The I2P network allows anonymous communications between two parties through an abstraction layer, which uncouples the association between the user and its identity. Basically, an application using I2P will not longer be reachable through an IP, but through a *location independent identifier*. The following sub-sections address the I2P network and its features.

### 2.1 Overview

The network is formed by a group of *routers*, which is the software that allows any application to communicate through I2P. Applications running on top of it will have a *destination* associated, which receives incoming connections from third

hal-00744919, version 1 - 24 Oct 2012

parties. The secret lies in which destination is associated to which router and not in the fact that a user is running an instance of the router. This decoupling between the router and the destination provides a certain degree of anonymity.

I2P uses a variation of the well-known onion routing approach[7], in which a message is routed from its originator to the final endpoint through several intermediate nodes using layered encryption. This variation is called *Garlic routing*, in which several messages along with their delivery instructions can be encapsulated into a single message and encrypted with the receiver's key. The integrity required for garlic messages are achieved through an hybrid *ElGamal/AES+Session Tags* symmetric-asymmetric approach.

The path through a selected list of nodes is called a *tunnel* and it is one of the key concepts in I2P.

### 2.2 Tunnels

Every tunnel is unidirectional, and is formed by the gateway (entry point), a set of participants (intermediate nodes) and an endpoint (exit point). Two types of tunnels exist. *Inbound* tunnels allow a user to receive data, and *outbound* tunnels to send data. A fully bidirectional communication between two users will involve four tunnels, one inbound and one outbound for each user.

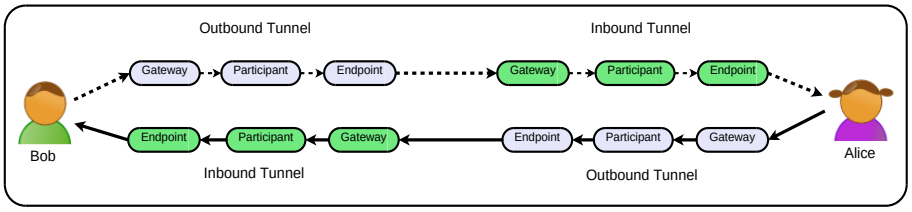



**Fig. 1.** Simple tunnel-oriented communication in I2P

Figure 1 illustrates a communication between Alice and Bob. Alice sends a message through her outbound tunnel, targeting one of Bob's inbound tunnel gateway. Once the message reaches the *gateway*, which is the entry point for Bob's tunnel, it is forwarded all the way through Bob's router. Alice does not have knowledge about Bob's inbound tunnel, but only about the entry point: Bob might have as well a tunnel composed with 1 or 100 intermediate nodes, but Alice ignores this, leading to the earlier mentioned *decoupling*.

### 2.3 Profiling algorithm

Tunnels are created every 10 minutes, and then dropped. This feature hardens a traffic analysis attack, since a 10 minutes time window is rather small to acquire sufficient knowledge of the network. Hence, every 10 minutes, a user needs to select new nodes for its tunnels, which are selected among all the routers in the network.

hal-00744919, version 1 - 24 Oct 2012

I2P leans on a constantly local profiling of all seen routers, so as to characterize every peer regarding its performance, latency, and availability. A four-tier scheme is used to classify routers: fast and high capacity, high capacity, not failing, and failing. Tunnels participants are randomly chosen from the fast and high capacity tier, and order through out the tunnel according to the *XOR* distance from a random key value. This *XOR* ordering prevents information leakage in predecessor and harvesting attacks.

Profiles are based on lookups in the network database, how often messages through remote routers fail, how many new routers are these remote routers able to introduce to us, for example. All kind of indirected behaviour is recorded and used for profiling, to the contrary of claimed performance from routers. No published performance information is used in local profiling, so as to avoid attacks based on announcing highly performant routers.

### 2.4 I2P netDB

The *netDB* is another key concept within the I2P network. It is a distributed hash table based on the Kademlia[8] protocol, used to store and share network metadata. However, on the contrary to the Kademlia protocol, not every peer in the I2P network forms part of the DHT, but only those fast and performant I2P users, the so called *floodfill* peers.

Any I2P user can become a floodfill node if two conditions are met: the number of estimated floodfills in the network is less than 250 nodes, and the user's I2P router has more than 128 KB/s of available bandwidth assigned.

There are two types of network metadata, *leasesets* and *routerinfos*. A leaseset provides information about a specific destination, like a web server, a BitTorrent client, an e-mail server, etc. A routerinfo provides information about a specific router and how to contact it, including the router identity (keys and a certificate), the address where to contact it (IP and port), several text options and a signature.

This distributed database contains an extra security feature, to harden a localized *Sybil attack*. The key used to index a record in the netDB is computed as `HASH(ID + date)`, in which the *ID* is the record ID, and *date* is the current date. A record ID remains fixed as long as the record is conserved, however the record indexing (routing) key changes every day.

This produces that at midnight, all indexing keys will be changed and therefore re-published in other locations in the DHT. Even though some queries might fail around midnight, this approach avoids an attacker to launch a simply localized Sybil attack, since the attacker will have to re-compute its Sybils IDs using a key-to-key dictionary. A deeper view of this network database is out of the scope of this document, however a further insight of the netDB and the flooding mechanism can be found in the official I2P website[6].

hal-00744919, version 1 - 24 Oct 2012

## 3 I2P monitoring approach

Our primary goal is to monitor the I2P network by qualifying file-sharing within the network, and study if this monitoring presents any kind of anonymity risk.

As earlier mentioned, I2P bases its anonymity on decoupling the identity of a user (provided by its routerinfo) from the application it is running (provided by its leaseset). Therefore, our challenge is to determine whether a given leaseset is running any known I2P file-sharing application, even if we can not link a particular user with a specific file-sharing application. This section introduces our monitoring architecture, and how it is implemented.

### 3.1 Exploting I2P netDB

As described previously, the netDB stores every routerinfo and every published leaseset. We aim to retrieve from the netDB as many leasesets as possible, for further analysis. Higher the number of leasesets retrieved and analysed, higher will be the characterization of the network.

We exploit the mechanism for becoming a floodfill, mentioned in Section 3.5, by placing a set of modified floodfill nodes in the netDB, which behave exactly as normal floodfills from the point of view of network operations (storage of routerinfos and leasesets), but perform a deeper analysis of these leasesets.

**Number of floodfills.** It is important to consider the netDB coverage of our architecture, and to determine how many monitoring nodes are needed to have a full (or good enough) network coverage.

Let $N$ be the number of floodfills and $X$ the replica factor, we consider the minimum number of floodfill monitor nodes as:

$$\texttt{nb\_monitors} = \lceil\ \texttt{N / X}\ \rceil\texttt{, N= \#floodfills , X= replica factor} \quad (1)$$

The replica factor indicates in how many netDB participants a given record (either a routerinfo or a leaseset record) is stored, for improving fault tolerance against participants going off-line. The I2P netDB defines a replica factor of 8, which means that any record is replicated in 8 participants for storing. If we consider that there are currently around 200 floodfill nodes in the I2P netDB and the replica factor of 8, we need 25 **perfectly distributed** monitor floodfill nodes out of the 200 nodes, to have a complete coverage.

**Distribution of floodfills.** In equation 1 we assume that the monitoring nodes are perfectly distributed over the DHT space, which is not always the case. Even if the SHA2 hashing function used in the netDB assures a fairly well distribution of the nodes through out the DHT-space, it might happen that our monitoring nodes are *all together*.

hal-00744919, version 1 - 24 Oct 2012

Figure 2 displays two examples of how the same set of monitoring floodfill nodes might be positioned in the nedDB: the upper one shows all the monitoring nodes grouped in one side, whilst the lower one shows a more dispersed distribution.

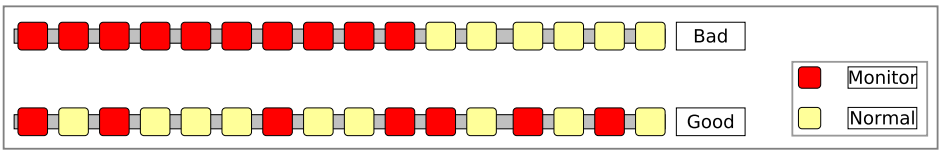



**Fig. 2.** DHT nodes distributions

It is clear that the *position* of the nodes plays a significant role in our architecture, and makes the result of equation 1 the lower bound of the floodfill monitoring nodes required. If the *positioning* of the nodes is not dispersed, we require further additional monitoring nodes.

Choosing the placement of our floodfill nodes will assure us a perfect distribution, however as mentioned in section 3.5, we would need a key-to-key dictionary to perform this. Nonetheless, we can verify whether our monitoring nodes are well distributed or not, by considering their position in the DHT-space along with the *optimal distance*. The optimal distance represents the distance between any two monitoring nodes for a correct distribution in the DHT-space.

It is important to consider that figure 2 represents the DHT-space as a linear space for an easy visualization. However, the I2P netDb is a Kademlia-based implementation, and therefore the concept of distance is based on the XOR metric, rather than in a linear distance.

Equation 2 presents the optimal distance in terms of bits for a group of monitor nodes, given a $2^{256}$ DHT-space, such as the I2P netDB. This states how many bits of difference must exist between the floodfills routing keys, to achieve a fairly dispersed distribution.

$$\texttt{optimal\_distance} = \log_2(2^{256}/MN), \texttt{ MN= Monitors nodes} \qquad (2)$$

With $2^4$ monitoring nodes for example, there must be a difference of $\log(2^{256}/2^4) = 252$ bits for a perfect distribution and an optimal coverage of the DHT-space.

### 3.2 File-sharing application analysis

I2PSnark, IMule and I2Phex are the three main I2P available file-sharing applications. We analyse each of them in the following manner:

**I2PSnark.** Firstly, we establish a connection with the leaseset (similar to a TCP socket-like connection). Then we send a first message, a well-formed BitTorrent message, requesting a non-existing torrent ID. If that given leaseset is actually running an I2PSnark client, it will immediately close the connection. Secondly,

hal-00744919, version 1 - 24 Oct 2012

we re-open the connection and send a malformed BitTorrent message. In the only case this response timeouts, then we can conclude the given leaseset is running an I2PSnark client.

**IMule.** An IMule client needs two different leasesets, one for its TCP connections (for file-transfer) and one for UDP messages (for indexation). We send an eDonkey `hello_packet` in case for a TCP-like leaseset, and a KAD `hello_request` in case for an UDP-like leaseset. If any of the messages have a response, then we conclude this leaseset is running an IMule client.

**I2Phex.** A `GNUTELLA CONNECT` message is sent for testing a leaseset against an I2Phex Gnutella client. If there is a Gnutella-protocol response, then the leaseset is running a I2Phex client.

### 3.3 I2P monitoring architecture

Figure 3 presents our complete monitoring architecture. This architecture has a set of distributed probes (monitor floodfills) dispersed around the I2P netDB for data collection (`routerinfo` or `leaseset` records), while a group of databases are used to determine the geographical location of these records. All retrieved data is stored in a central database, for a further correlation analysis and a final display in our statistics website.

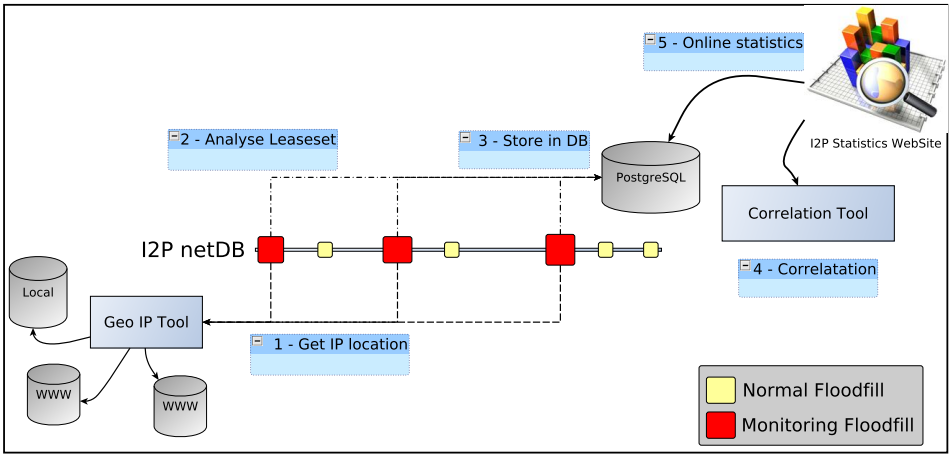



**Fig. 3.** Monitoring architecture

When a monitor floodfill receives a `routerinfo store message`, it looks up the estimated location of its IP address and stores the record in the central database. If a `leaseset store message` is received, the monitor floodfill runs the analysis for file-sharing applications. A periodically analysis is computed,

hal-00744919, version 1 - 24 Oct 2012

including correlation between top countries and most used file-sharing applications, top cities, top file-sharing application, correlation between file-sharing applications, number of retrieved leasesets and routerinfos. All the results obtained are shown in a website, along with the state of the monitoring architecture.

On the one hand, our architecture is completely flexible: due to the autonomous nature of the monitoring floodfill nodes, the total number of these nodes can be increase at any time, increasing the amount of network metadata retrieved, therefore increasing the accuracy of the analysis.

On the other hand, the implementation of our monitoring floodfill nodes is freely available, along with its source code, hence any user willing to contribute to the analysis of the network can download a monitoring floodfill node and deploy it.

## 4 Experiments

In this section, we present our experiments on monitoring the I2P network for file-sharing applications.

### 4.1 Experiment set-up

We monitored the network for 90 hours, from 2012-06-07 15:00:00 (UTC+2) to 2012-06-11 09:00:00 (UTC+2). We contemplated a weekend in our measurements, being these days usually more suitable for file-sharing.

Based on the estimated number of 220 floodfill nodes in the I2P network[1] during our experiment, and in the equations presented in section 3.1, we needed a total of $\lceil\ 220\ /\ 8\ \rceil = 27.5$ floodfill monitor nodes for a perfect coverage, with a distance in terms of bits of $\log\left(2^{256}/27.5\right) = 251.67$ bits.

The PlanetLab test-bed was employed for this particular experiment, and due to technical limitations, only 20 monitor floodfills were deployed. By the time of the experiment, all of these floodfill nodes had an average previous uptime of 24 hours, so as to assure a good integration with the network.

Due to the particular indexation of records in the netDB, which includes the current date in the routing key calculation, the monitor floodfill nodes shift from one location to another one every day. We take our first day of experimentation, and calculates our monitor nodes distribution within the DHT-space.

A distance of 252 bits was computed between all of our floodfill nodes, which indicates that our floodfill nodes are fairly well distributed through the DHT-space. Taking into account this pseudo-perfect distribution, we can estimate that with 20 floodfill out of 27.5 we cover 72% of the network the first day of the experiment. Being that the monitor nodes move in the DHT-space, this estimation need to be done in a daily-basis.

It is important to consider that because the pseudo-perfect distribution of our nodes, we can cover 72% of the network. However, in a completely random

[1] `http://stats.i2p.to/`

hal-00744919, version 1 - 24 Oct 2012

distribution of monitor floodfill nodes within the netDB, the network coverage will drop. A further complete analysis can be found at `www.i2pstats.loria.fr`.

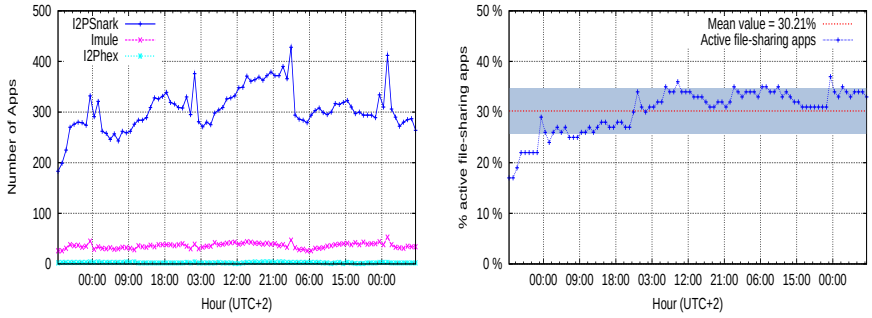
### 4.2 File-sharing applications within I2P

Figure 4(a) displays the individual number of active file-sharing applications during the course of our analysis, for I2PSnark, IMule and I2Phex clients.

I2PSnark clients usage highly exceed the rest of the measured applications, in which I2Phex presents an almost zero usage during the experiment, with as few as 11 clients detected.

Figure 4(b) presents the file-sharing usage when compared to the total amount of analysed applications, in which it can be observed that during the weekend (right side of the graphic) the file-sharing usage increases up-to 38% of the total network usage, mainly guided by the I2PSnark clients, and it decreases during week days, which is an usual file-sharing characteristic. We compute an average usage of 30.21%, with a standard deviation of 4.38%.

4 well-defined peaks appear in the chart, all around midnight. This increase does not represent an increase in the real I2PSnark users base, but rather an exact view of how the netDB is implemented: at midnight every routing key changes, which means that every record (*routerinfo* or *leasest*) is re-located in another DHT location. Our floodfill monitor nodes receive these new storing requests, which increase the total number. However, previous stored records are not longer kept within our floodfill monitor nodes, therefore after the peaks at midnight, the measurement returns to normal.




(a) I2PSnark / IMule / I2Phex (b) File-sharing usage

**Fig. 4.** File-sharing applications within I2P

*Why is it that I2PSnark prevail over IMule or I2Phex clients?* I2PSnark is a BitTorrent-based client, which is fairly popular among file-sharing applications, and it is additionally built-in within the I2P software, which makes it rather easy to be used. Furthermore, the low number of IMule clients measured suggests a low number of sources (peers having the entire content) for IMule files, which leads to new clients choosing I2PSnark rather than IMule for file-sharing: less sources is translated into slow download rates or even no download at all.

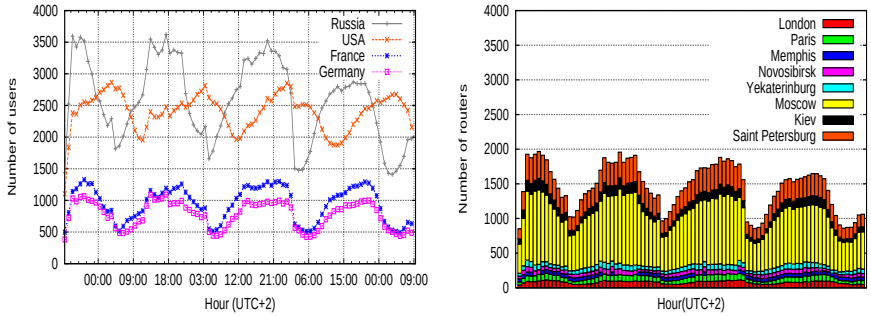
hal-00744919, version 1 - 24 Oct 2012

### 4.3 Country-based analysis

In this section we analyse users location, to determine which countries and cities contribute the most to the I2P network. End-users geographical location is represented by the geographical location of their routers within the I2P network.

Figure 5(a) presents the most active countries within the I2P network: Russia, The United States, France and Germany.

Russia and The United States present both a participation over 1500 routers at any point in time (with few exceptions). However their activity is inversely proportional through time in this graphic, probably because Moscow and Saint-Petersbourg (most seen Russian cities), and Memphis (most seen U.S. city) have a time difference of 10 hours, therefore the U.S. curve is out of phase with the *European* time-zone, which does not imply that U.S. participants behave different that European ones.




(a) Top 4 active countries (b) Top active cities

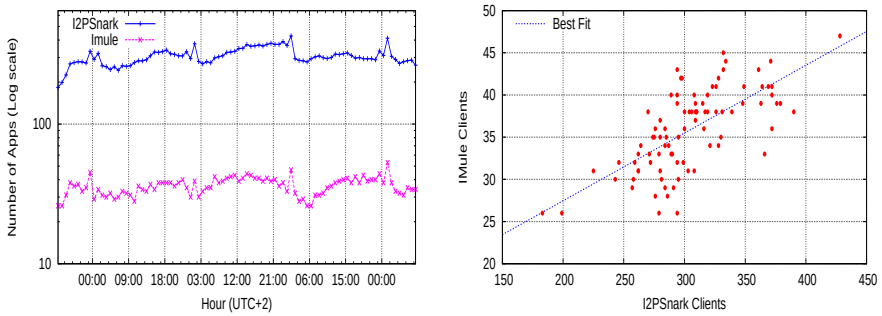
**Fig. 5.** Country/City distribution

Figure 5(b) displays the most seen cities, in which Russian cities predominate over the rest of the cities. Moscow seems to be the most active Russian city detected, with more than 50% of participation in most cases. In both cases, country-based and city-based, Russia contributes the most to the I2P usage, with an average of 40% of the total participants in the network.

During the whole experiment, we detected 140 countries in total, and considering the 4 top countries, we identified 500 different Russian cities, 2130 different U.S. cities, 1670 different French cities, and 1654 different German cities. It is clear that I2P usage is not confined to a narrow group of countries or cities, but rather distributed all over the world.

### 4.4 Correlation between file-sharing applications

In this section we try to determinate if users of different file-sharing applications behave the same way: *does an increase of I2PSnark clients necessary imply an increase of IMule clients?*. We do not consider I2Phex clients, due to its low number of measured clients in the network.




(a) Detailed view (b) I2PSnark/IMule Pearson-correlation

**Fig. 6.** File-sharing applications correlation

Figure 6(a) displays a detailed view (log scale on the Y-axis) of figure 4(a), in which an increase on IMule clients can be deducted when an increase of I2PSnark clients occurs. Figure 6(b) presents a scatter plot with I2PSnark and IMule clients usage, revealing a positive 0.7106 correlation value among the usage of these two file-sharing clients. In this case, file-sharers behave the same way regardless which file-sharing applications they are using.

In our previous work[9] we show that eMule/aMule clients stay connected for longer periods of time, on the contrary to BitTorrent users. This behaviour is not seen in this case, most likely due to the fact that for I2PSnark or IMule applications to run, an I2P user needs to have a running I2P router. Being that I2P users are more active at night, file-sharing applications will also be active during night time, independently of the type of the applications, either I2PSnark or IMule.

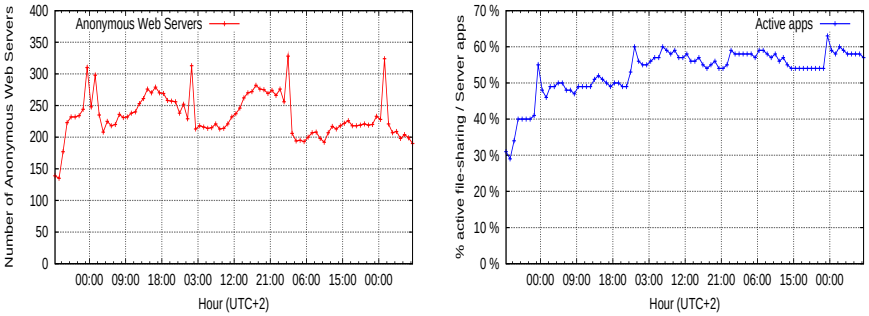
### 4.5 Non file-sharing applications

The I2P network allows anonymous web sites, called *eepSites*, in which any user can host an anonymous web server, and through a DNS-like service, I2P users can resolve domain names such as `tracker2.postman.i2p` (a major I2P tracker). Due to this built-in feature, we additionally measured anonymous web servers within I2P, by sending a `GET` message to a leaseset and processing the answer.

Figure 7(a) displays the number of active anonymous web servers measured during our experiments, with an average value of 240 online servers at any time and an average of 22.9% out of the total usage of the network. In this case, there is no noticeable increase of servers during the weekend, which indicates that anonymous servers are quite stable within the network. In this measurement we can observed the same behaviour as mention in Section 4.2, in which at midnight new record storing messages arrive.

Both file-sharing and anonymous hosting reach to almost 65% of the total usage of the network, as seen in figure 7(b). The increase in the right part of the graphic is driven by file-sharing users, more than anonymous web servers.

hal-00744919, version 1 - 24 Oct 2012




(a) Anonymous Web Servers (b) Aggregated usage

**Fig. 7.** Anonymous web servers usage

## 5 Related work

Among anonymous low-latency networks, the *Tor* network is probably the one that has been receiving more academic attention.

Chaabane et al.[4] conduct a traffic analysis in Tor by quantifying BitTorrent and HTTP protocols, and successfully detecting that Tor exit nodes are used as 1-hop SOCKS proxies rather than in a full-length onion tunnel.

Le Blond et al.[10] present a monitoring study in Tor, in which they successfully trace TCP streams within the network, and unveil BitTorrent users using the Tor network.

McCoy et al.[5] present a monitoring approach which analyses the application layer of outgoing traffic to determinate the protocol distribution in the Tor network, based on a Tor exit node. They detect that most of the connections through Tor are interactive http traffic, and that a very few of them are for BitTorrent traffic. However, these few BitTorrent connections (3.33%) consume a disproportionally amount of bandwidth (40.20% of the total measured).

Loesing et al.[11] introduce a measurement of sensitive data on the Tor network, such as country of connections of users and exiting traffic by port. Furthermore, Loesing [12] measures the *trends* of the Tor network from directory information.

Nevertheless, the Tor network has a central *directory server* and hence the monitoring approaches based on this central component can not be applied in the I2P network, which does not include any kind of centralized directory.

In our previous work[13] we take the first step through monitoring the application layer of the I2P network by means of a centralized architecture and measure a single file-sharing application along with I2P's hidden services.

Adrian Crenshaw[14] exhibits a mechanism to identify I2P's hidden services, called *eppSites*. He successfully links an anonymous website to its real IP address, taking advantage of the lack of control in the application layer, and misconfiguration of web servers over I2P. Crenshaw's approach includes crawling his local list of known participants along with the list of known *eepSites*, whereas our approach focus on the complete netDB.

hal-00744919, version 1 - 24 Oct 2012

hal-00744919, version 1 - 24 Oct 2012

Herrmann et al.[15] conducted an attack on the I2P network, so as to determining the identity of peers hosting *eepSites*, which can be considered a monitoring technique for anonymous websites. They proposed a three-step attack, in which an adversary with modest resources can identify an I2P user as the host of an *eepSite*. Their focus on a particular victim, rather than on the entire network.

Large scale monitoring on further popular anonymous low-latency networks, such as JAP, Freenet or GNUnet, has not been performed or even proposed in the literature.

## 6 Conclusion

We have designed and successfully implemented and deployed the first large scale monitoring architecture for the I2P network, mainly focused on anonymous file-sharing. We are able to provide deeper insights about the behaviour of the network, providing us with correlation values among users locations and file-sharing applications usage, which clearly states an anonymity warning: group-based identification for file-sharing applications is possible within the network, even if single-user identification is not.

We have measured that file-sharing within I2P obtains an average of 30%, peaking 35% during weekends, in which the I2PSnark client is the most used over IMule and I2Phex. There appears also to be a correlation among Russian users and file-sharing applications even in a 90-hours analysis. An ongoing and longer monthly analysis confirms this correlation.

In addition, we measured 22.9% of anonymous web servers hosted within I2P, which along with file-sharing users, add up to the 65% of the total network performing these two activities.

A real-time analysis of the network can be found on our I2P statistics web site, `www.i2pstats.loria.fr`, which presents the top countries, top cities, amount of file-sharing applications, amount of online users, correlation among file-sharing applications and countries, etc. We additionally include a raw access to our database, so as to researchers can query and create their own requests.

Future work targets the design of an improved technique for optimal placement of our monitor floodfill nodes, in which every monitor node chooses its placement in the DHT-space, achieving a perfect DHT distribution, thus increasing the total coverage of the network, whilst reducing the required number of monitoring nodes.

## References

1. Georgos Siganos, Josep M. Pujol, and Pablo Rodriguez. Monitoring the Bittorrent Monitors: A Bird's Eye View. In *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, PAM '09, Seoul, Korea, April 2009. Springer-Verlag.

hal-00744919, version 1 - 24 Oct 2012


2. Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. A global view of Kad. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, San Diego, California, USA, October 2007. ACM.
3. Anirban Banerjee, Michalis Faloutsos, and Laxmi Bhuyan. The P2P War: Someone Is Monitoring Your Activities! In *Proceedings of the 6th International IFIP-TC6 Networking Conference*, Networking '07, Atlanta, GA, USA, May 2007. Springer.
4. Abdelberi Chaabane, Pere Manils, and Mohamed Ali Kaafar. Digging into Anonymous Traffic: A Deep Analysis of the Tor Anonymizing Network. In *Proceedings of the 2010 4th International Conference on Network and System Security*, NSS '10, Washington, DC, USA, September 2010. IEEE Computer Society.
5. Damon Mccoy, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the Tor network. In *Proceedings of the 8th Privacy Enhancing Technologies Symposium*, PETS '08, Leuven, Belgium, July 2008. Springer.
6. I2P. The I2P network. `http://www.i2p2.de/`.
7. D. Goldschlag, M. Reed, and P. Syverson. Hiding routing information. In *Proceedings of the 1st International Workshop on Information Hiding*, IH '96, Cambridge, UK, May 1996. Springer.
8. Petar Maymounkov and David Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Revised Papers from the 1st International Workshop on Peer-to-Peer Systems*, IPTPS '02, Cambridge, MA, USA, March 2002. Springer-Verlag.
9. Juan Pablo Timpanaro, Thibault Cholez, Isabelle Chrisment, and Olivier Festor. When KAD meets BitTorrent - Building a Stronger P2P Network. In *Proceedings of the 8th International Workshop on Hot Topics in Peer-to-Peer Systems*, HotP2P '11, Anchorage, Alaska, USA, May 2011. IEEE Computer Society.
10. Stevens Le Blond, Pere Manils, Abdelberi Chaabane, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One bad apple spoils the bunch: exploiting P2P applications to trace and profile Tor users. In *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, LEET '11, Boston, MA, March 2011. USENIX Association.
11. K. Loesing, S. Murdoch, and R. Dingledine. A case study on measuring statistical data in the Tor anonymity network. In *Proceedings of the 14th Financial Cryptography and Data Security*, FC '10, Tenerife, Canary Islands, Spain, January 2010. Springer.
12. K. Loesing. Measuring the Tor network from public directory information. In *Proceedings of the 4th Hot Topics in Privacy Enhancing Technologies*, HotPETS '11, Seattle, WA, USA, August 2009. Springer-Verlag.
13. Juan Pablo Timpanaro, Isabelle Chrisment, and Olivier Festor. I2P's Usage Characterization. In *Proceedings of the 4th International Workshop on Traffic Monitoring and Analysis*, TMA '12, Vienna, Austria, March 2012. Springer.
14. Adrian Crenshaw. Darknets and hidden servers: Identifying the true IP/network identity of I2P service hosts. In *Black Hat 2011*, Washington, DC, USA, March 2011.
15. M. Herrmann and C. Grothoff. Privacy-Implications of Performance-Based Peer Selection by Onion-Routers: A Real-World Case Study using I2P. In *Proceedings of the 11th Privacy Enhancing Technologies Symposium*, PETS '11, Waterloo, Canada, July 2011. Springer-Verlag.