Anonymity Technologies and Usability: Latency in Tor

Florian Goertz, Mathias Nitzsche

Humboldt Universität zu Berlin, Institut für Wirtschaftsinformatik

{florian.goertz@student.hu-berlin.de, mathias.nitzsche@student.hu-berlin.de}

Abstract. The Onion Router is a low-latency network to anonymize online traffic, with currently more than 100,000 users per day. Like in all other anonymity networks, lower performance is the tradeoff the user has to make to gain anonymity. This fact is broadly known, but little quantified data exists in what extent the performance is actually slowed down.
This paper analyzes the performance gap by conducting different automated experiments using over 7,500 HTTP and about 20,000 DNS requests. The empiric data the study is based on revealed a loss factor of 5 while requesting websites and a factor 40 to resolve hostnames. Analysis of upload and download speed showed the same trend.

Keywords: Tor, anonymity, network, Internet, performance, latency

1 Introduction

The need for security while using the Internet is broadly known. Security mechanisms and programs are commonly focusing on encryption, the detection of malware and avoiding unauthorized access to systems. An often underestimated aspect in this area is anonymity.
Especially in countries with repressive political regimes, but also in western democracies ongoing trends in law, policy, and technology threaten anonymity and privacy as never before. Governments, companies and even criminals are surveilling and logging online activities connected to certain individuals. Well known recent examples are the censorship in China and even the “EU Vorratsdatenspeicherung”. To achieve online anonymity and so defend personal freedom a widely used technique are anonymity networks in general and a network called Tor ("The Onion Router") in particular. [1a]
This paper will introduce Tor with a short overview of the history, the usage and the functionality. Afterwards the usability of Tor and the actual provided anonymity will be discussed briefly.
The focuses of this work are experiments to provide quantitative data on the performance of the network. Factors of influence will be considered and as far as possible eliminated. Therefore an experimental environment will be established to conduct several automated and repeatable experiments.
The gained results will be summarized and evaluated. Furthermore as a result of this work several scripts and analysis tools are provided to allow the execution of further even long term research in this field.

2 Tor: A Low Latency Anonymity Network

By design anonymity does not exists in the Internet, since the source of every action can be determined by its unique IP-address. Anonymity networks try to provide technical solutions to this issue, all with the goal of hiding user’s activities within a large set of other users.

2.1 History

The idea behind modern anonymity systems dates back to the 1980s to Chaum's Mix-Net design. He proposed to wrap messages in layers of cryptography and replay them through different "mixes". Each mix decrypts, delays, reorders and sometime resizes messages before forwarding them. This worked fine for the initial intention of anonymizing email sending, but was too inefficient for more time critical communication like chatting or SSH. Subsequently the research and development in this area demerged into two main directions.

On the one hand high latency systems like Babel, Mix-Master and Mixminion are aiming to maximize anonymity at the cost of comparatively large and variable latency. On the other hand low-latency anonymity networks, like Java Anon Proxy, Freedom, MorphMix or Tor are trying to offer as much anonymity as possible, while keeping the latency within an acceptable range. [2]

2.2 Usage

Currently the most popular of these networks is the low latency Tor network, with around 100.000 users per day from all over the world. [3]
The anonymity provided within the network attracts many different groups of users like companies, governments, military or private individuals. [4] [21] Furthermore the Tor community is proud of providing anonymity to journalists and activist groups in countries with repressive political regimes, who fear repression or even persecution for their political opinions, if their identity is unveiled. A recent study showed significant growth of Tor users in China, when governmental censorship was increased and in Iran, when the riots after the presidential election have taken place. [5]
On the other side also criminals are using the anonymity provided by the network for illegal purposes, like slander, distribution of child pornography, illegal threats, racial agitation or fraud. The community behind Tor is very aware of this issue and tries to minimize possible harm, but argues that criminals who are willing o break the law or invest money, already have options available providing a better anonymity than Tor. Moreover it's very unlikely that forbidding Tor would stop criminals from their activities. [1]

2.3 Functionality

Tor is designed to anonymize TCP based traffic like web browsing or SSH. Therefore a Tor client chooses a random path through the network and builds a circuit, in which each Tor node (also called relay) only knows its predecessor and successor. Data is sent over this circuit in fixed-size cells, which are unwrapped (like the slices of an onion) with a symmetric session key at each node of the route. The last node (called exit node) unwraps the last layer of encryption and then knows to which host a packet needs to be forwarded.[6] This implies that for sensitive data a further end-to-end application layer encryption, like SSL for web browsing is needed. Otherwise it has been shown, that an adversary who runs a Tor relay can easily obtain private information like passwords for email accounts. [7]
Other published attack scenarios point out, that in opposite to high latency systems Tor cannot resist strong global adversaries, who can observe both ends of the communication or a regional adversary who control a significant number of Tor nodes. In these cases the adversary could statistically correlate incoming and outgoing data to identify the source of a message. [8] [9]
These vulnerabilities are considered somehow theoretical in everyday use, and are outweigh by the advantages regarding Tor’s usability.

2.4 Usability

A growing field of scientific research in the security area is how usability impacts security and vice versa. [4] [10] [11] [12] The important role of usability was not considered in earlier anonymity network designs, what resulted in hard-to-use systems with very few users. This ongoing conflict between usability and anonymity was well understood by the Tor designers, who saw usability as the most important requirement, beside the main design goal of providing anonymity by frustrating attackers from linking communication partners.

Usability consists of many factors like: [6]

- Deployability: Tor must not be expansive to run (e.g. by requiring too much bandwidth) or difficult to implement or install (by requiring OS-kernel changes or deep network knowledge).
- Portability: Tor has to run on every system, since it cannot be expected that users will change their operating systems.
- Ease-of-use: Tor has to be easy to use to attract many users and so provide more anonymity, since in anonymity networks users hide among users.
- Usefulness: Tor must provide as low latency as possible to enable time critical communication, like web browsing.

Different studies have shown that the first 2 points are fulfilled by Tor, whereas the link between the last 2 points using the technology acceptance model has not been researched so far. Furthermore little data exists regarding the actual performance tradeoff a user has to make to be anonymous with Tor. The question this paper tries to answer with several experiments is: "How low is the low latency actually" and is Tor still usable?

3 Tor Performance Measuring

With the help of experiments the Tor performance is analyzed and measured. These experiments and the surrounding environment will be presented in the coming sections.
To evaluate Tor, the performance is directly compared to the performance of connections to the Internet without using Tor. Important processes of Internet communication are quantified and measured, like:

- Duration of DNS name resolutions
- Duration of HTTP requests to websites
- Throughputs for download and upload via HTTP

In addition the frequency of change of exit nodes and corresponding performance lacks are analyzed.

3.1 Related Research

Several authors have already discussed why Tor is slower and have proposed how to improve the performance. [13] [14] [15] Within the Tor Metrics subproject different users have provided long term data of the performance, shown in Figure 1.

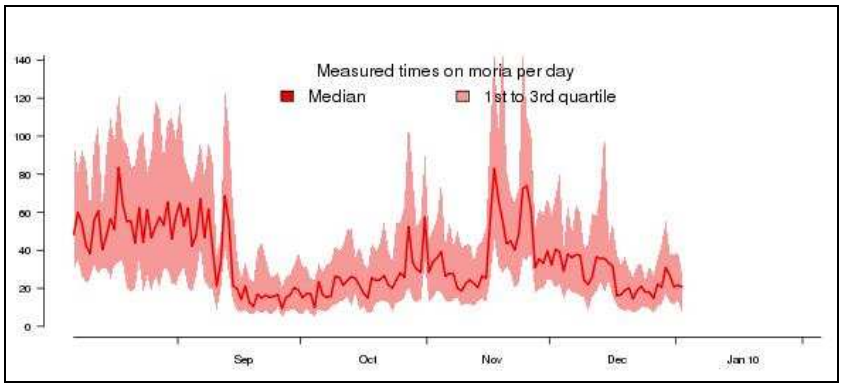

Figure 1: Time needed to download 1MB across the Tor network [16]

These results indicate that the performance of Tor is volatile over time, but do not state anything regarding the actual performance loss of an end user of Tor. The presented paper is aiming to close this gap, as it is focusing on a comparison between using Tor and not using Tor from an end user's perspective.

Other research approaches have conducted demographic studies, regarding the number and country of exit nodes and usual Tor users. [5]

3.2 Environment and Assumptions

Within the scope of this chapter the environment is delineated to ensure the traceability of the experiments.

Topology

The experiment is done from an Internet user's point of view. To emulate a user and its behavior, while performing automated tests the assumptions are:

- that it is not relevant if a user is human or non-human,
- that a user is situated at a specific location,
- that a user utilizes a specific Internet access, hardware, and software,
- that a user is requesting different websites (represented by the Top 500 linked websites) over a given period of time.

Time restrictions

The time frame for the experimental tests is 3 days. Tests take place at different daytimes. Long time observations of Tor's performance are not part of current research.

Hardware and Software

Following hardware, software, and Internet access are used while conducting measurements:

Table 1: Experimental Environment

PC	IBM Lenovo X61s Intel Core 2 Duo 1,60 GHz 2GB memory
Operation System	Windows XP SP3
Software	TorPortable 1.3.1 Privoxy 3.0.12 Vidalia 0.1.15 Strawberry Portable 5.10.0.6 TorDNS v1.7
Internet Access	ADSL 16 Mbit/s Download 1 Mbit/s Upload ISP: Alice (Berlin, Germany)

Tools for measurements

For all measurements scripts (developed with Perl) assist to gain empiric and automate the following experiments. These Perl scripts are executable in Windows and Linux environments.

Test restrictions

Regardless of whether Tor is used or not, performance is very sensitive to competing network traffic. While tests are running, no competing traffic occurs.

3.3 HTTP Request Duration

Objective

This experiment has the objective of comparing the speed of usual web browsing between Tor and direct web access (without Tor). The evaluation of speed of web access is done by measuring the duration of HTTP Requests.

Execution

During one run of the experiment the Perl script traverses the provided list of target websites (Upper 50 of 500 most linked websites). For each website a HTTP request is

performed using the Perl LWP library once with Tor and once without Tor in an alternating sequence.
All Tor requests are directed over the SocksProxy "Privoxy", which then forwards the request over the Tor network. Additional static data on the websites, like images, videos, or Javascript files are not considered or transferred.
Date and time, request duration, and received bytes are logged for each request.
This procedure of requesting 50 websites was repeated altogether 50 times.

Outcome

The results are visualized and presented in the following figures.

Figure 2 directly compares the duration of HTTP requests with Tor and without Tor – aggregated over all 50 runs. The maximum and minimum values, the quartiles, and the median values are marked.
With Tor the middle 50% of all HTTP requests have durations between 3.9s and 12.6s, whereas with direct connection these 50% are situated between 0,79s and 1,95s. Furthermore Tor shows a larger range of values in its durations, what leads to the conclusion that the HTTP performance of Tor is not as stable as HTTP performance without Tor.
To best compare both the median value is chosen, because high deviation and peak values can compromise the average value. Tor's median of 6.98 s compared with 1.37s for direct connection indicates that HTTP requests over Tor are in average by the factor 5 slower. Thus, there have to be factors that impact the performance. Factor of influence might be e.g. the daytime, the current exit node, or even the state of the Tor network.

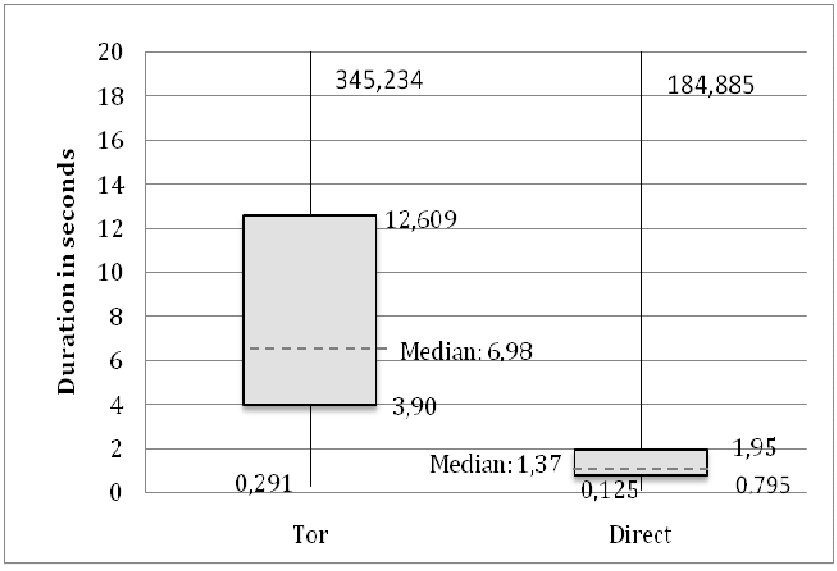

Figure 2: Comparison of HTTP request duration

The results from Figure 2 are proven in Figure 3 and Figure 4, where the results are separated by URL.

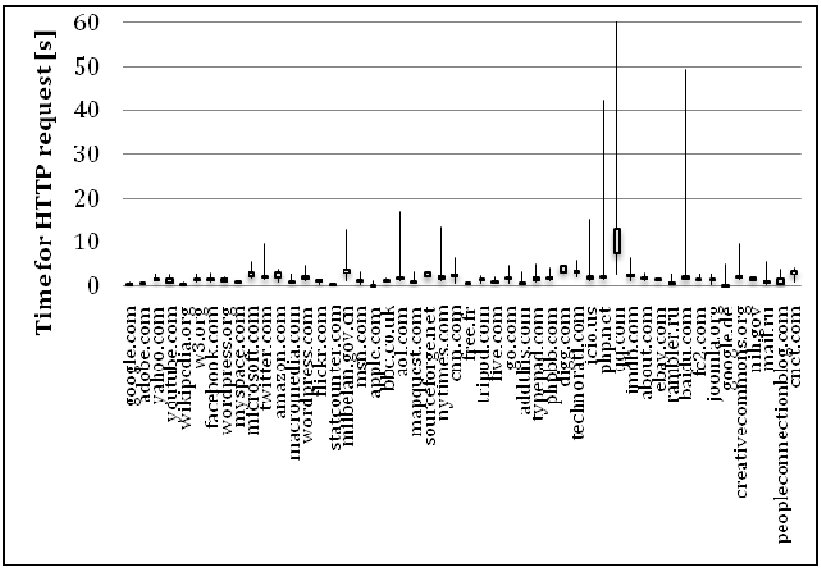

Figure 3: HTTP request duration separated by websites (without TOR)

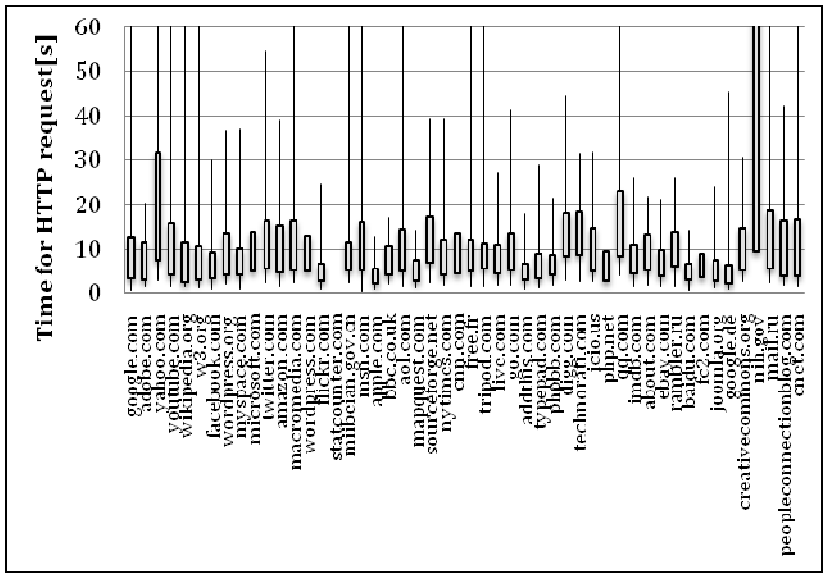

Figure 4: HTTP request duration separated by websites (with Tor)

Figure 5 shows the median of 50 collected values separated by websites. The values are connected (even if mathematically not correct) to show the correlation and emphasis the time distance already shown in the previous graphs.
(Tor has no value for statcounter.com, since this URL is rejected by Privoxy)

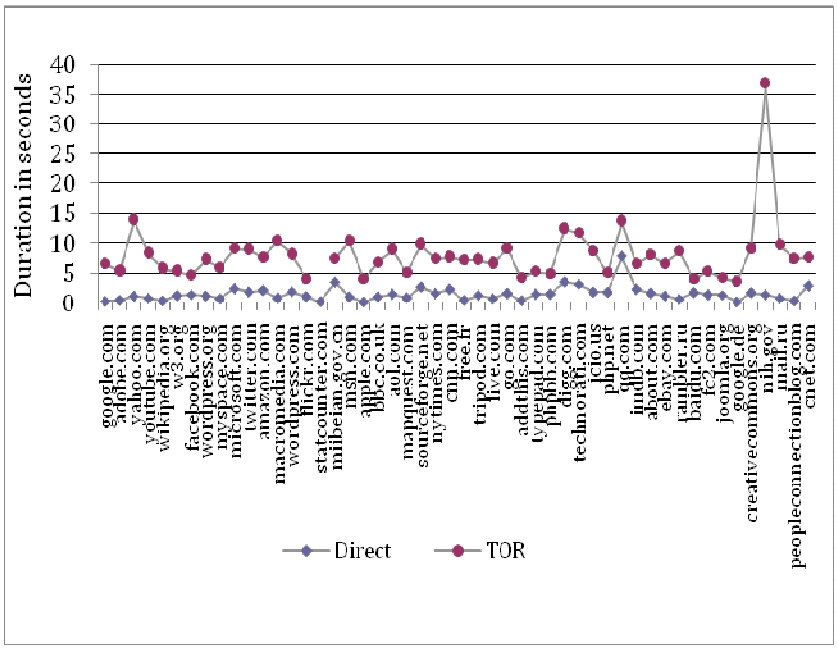

Figure 5: Request duration separated by host

Problems / Further research

While measuring the durations of HTTP requests the vagueness appear, that traffic overhead cannot be separated clearly. DNS name resolution might be involved in results as well as TCP connection establishments. The average time for DNS name resolution is analyzed in a further experiment, which gives an impression of how much DNS requests affect the presented results.
Furthermore, the used Perl script in its current version does not support HTTPS requests. Hence, some websites have been excluded from the result set.

3.4 Throughput

Objective

This experiment has the objective of comparing the throughput with and without applying Tor. The throughput can be simply measured by transferring a file across the network. Throughput is used here in the sense of “goodput”, meaning application

layer throughput and not the actual amount of sent bytes. Overhead like the TCP header is not considered, according to definition of goodput.

Execution

Within this experiment both, the upload and the download will be evaluated separately. In both cases tests are automated with the help of Perl scripts which use the Perl LWP library. All Tor requests are directed over the SocksProxy “Privoxy”, which then forwards requests over the Tor network.

The throughput is measured by logging the time for uploading and downloading files (or better parts of files) from 50KB up to 1MB in steps of 50KB. The whole experiment is repeated 30 times.

Download: The file to download is located on Google's servers: http://dl.google.com/picasa/picasa3-setup.exe. It is assumed that this file is distributed worldwide by Google's distribution content network and the servers are highly available. Based on that assumption the distance from any Tor exit node will be kept at a minimum and so not influence the experiment.

Upload: The upload is done in a very similar way. The Perl script uploads a string composed of repeating "1"s to Google.com, which is chosen, because of the same assumption as for downloads. Uploading a file results in a 403 HTTP Status Code “Forbidden” as the response, however, data is transferred and the error does not influence the results.

Outcome

Download

Figure 6 compares the 2 types of connections by time needed to download certain amounts of KBs. Shown are the aggregated values of all 30 runs. To compare the download speed of Tor and direct connection the average and median values of 30 runs are calculated. Concerning the high fluctuation, the median values are presented besides the average values to gain a more precise impression. The diagram proves ones again the speed difference by the factor of around 5. The 2 data rows are both nearly proportional.

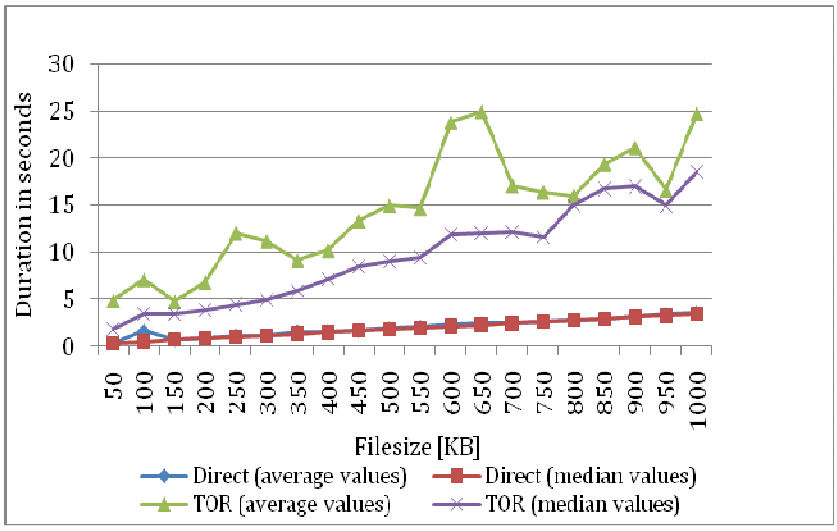

Figure 6: Download via HTTP of different file sizes

In Figure 7 and Figure 8 every run is drawn separately and by KB/s. While direct downloading is very linear (concerning the time) and converges against the maximum speed (concerning KB/s), the download with Tor can only be described as chaotic. It can be concluded, that the experienced download performance of Tor from a users perspective is not as stable and constant as direct connection.

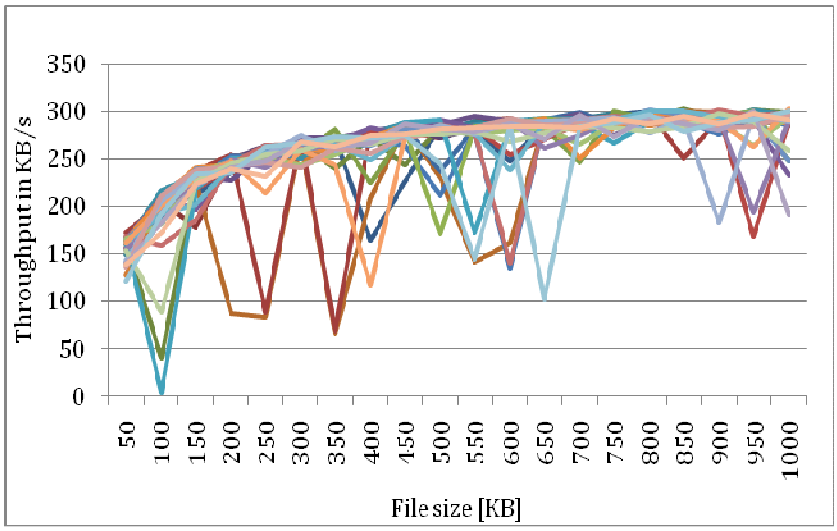

Figure 7: Download throughput without Tor

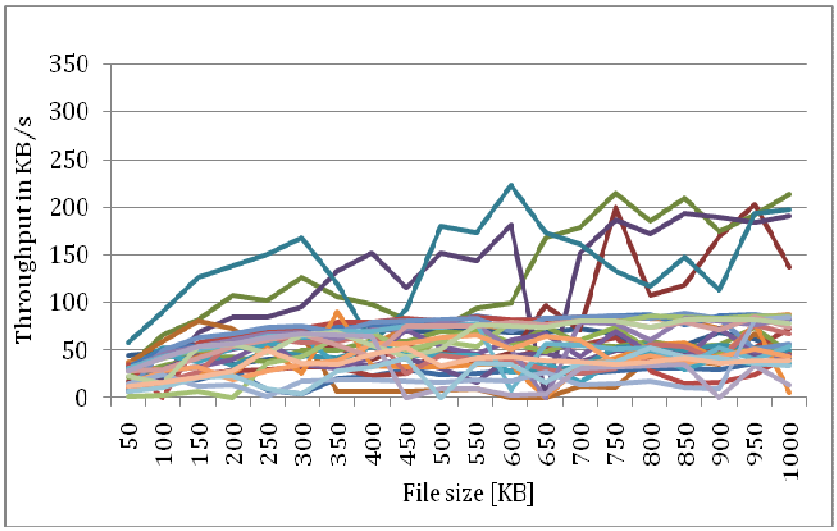

Figure 8: Download throughput with Tor

Upload

In Figure 9 the time which is needed to upload a specific amount of data is documented. The results are very similar to the ones from downloading and therefore not further discussed.

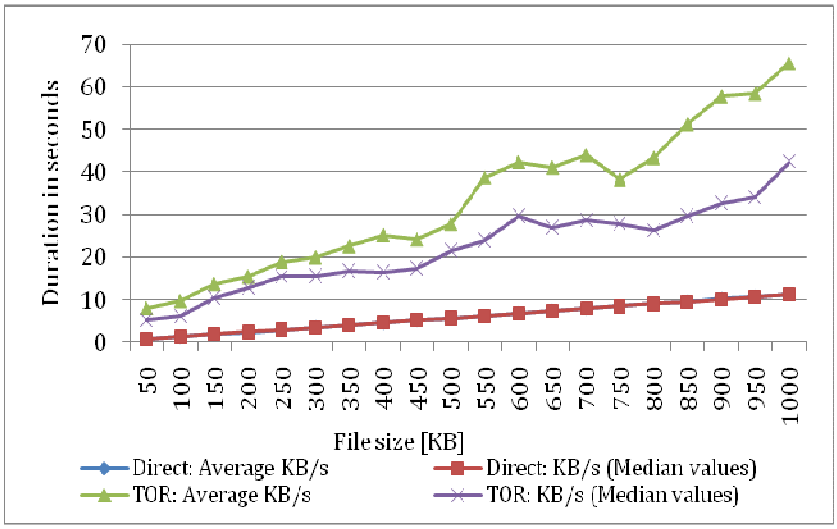

Figure 9: Upload via HTTP of different file sizes

Like before the difference gets clearly, when drawing each run separately – see Figure 10 and Figure 11. Tor again shows a high fluctuation.

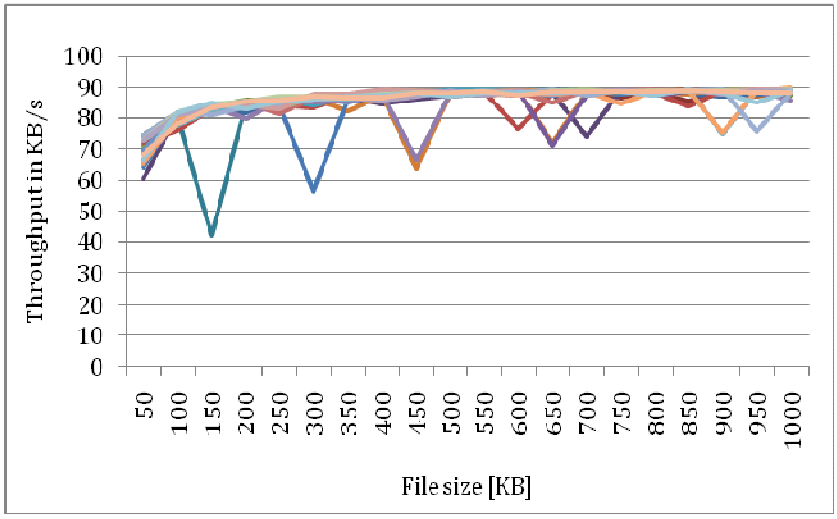

Figure 10: Upload throughput without Tor

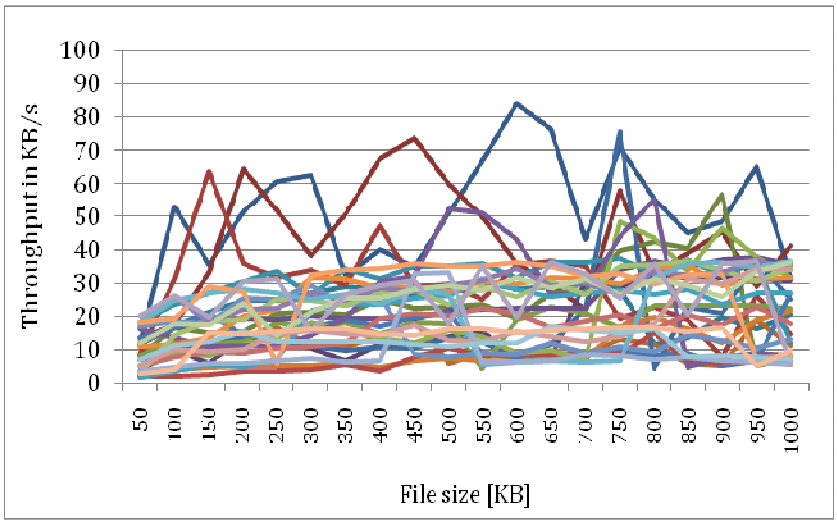

Figure 11: Upload throughput with Tor

Problems / Further research

The results strongly depend on the used Internet connection. Therefore, the results might be different when conducting additional tests in different test environments.
When a larger Internet connection is used it is assumed that the difference of performance will grow. That's because the average performance of all involved Tor nodes will be nearly constant. While using a slower Internet connection it can be

assumed that the ratio will shrink. Summarizing that fact, Tor does not scale with the size of the user's connection.
Further tests should be conducted with different settings of the experimental environment to gain more precise results.

While conducting the upload test, a string composed of repeating "1"s is used. There might be the possibility that routers in the Internet compress that string, because it is transferred in plain text, however, randomly chosen chars produced the same results.

Similar results are expected by using FTP, which was not part of the current experiment, since neither Privoxy nor TSocks can handle FTP. [17] [18]

3.5 TOR Exit Node Changes

Objective

This analysis will focus on the Tor circuit changes, which can be observed by a changed exit node. Especially interesting are the location and the duration of connection with each exit node.

Execution

As the data base for this experiment a log from the HTTP experiment with more than 4000 requests over 22 hours is used. During this time the location and IP-address of all exit nodes were logged. This was done via an additional HTTP request to an IP-test script over the Tor network between each of the test relevant requests. This script is written in PHP, hosted on a private test server and uses the utrace.de service to identify the location of the exit nodes IP-address.

Outcome

Over the time span of the experiment 194 exit node changes, to 54 different exit nodes have been observed. Several exit nodes have been used multiple times, even if during the whole test period around 700 Tor Exit Nodes were online. This shows that some of these nodes are for some reason never considered.

Location

Regarding the location, it was observed that relays in some countries (in this experiment USA; Germany) are used more often then others. Figure 12 shows the percentage of country of the exit node.

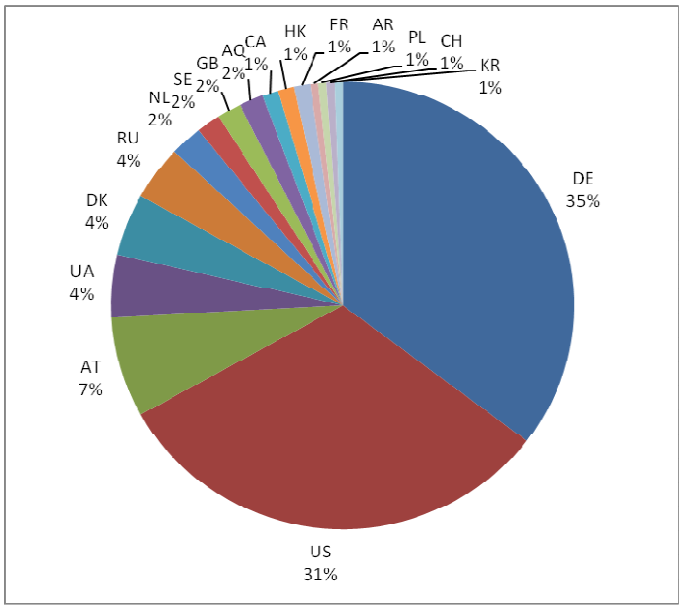

Figure 12: Distribution by Country of Exit Nodes

The reason for the high percentage of observed relays located in the USA and Germany is that these 2 countries are the two countries contributing the most to the TOR network. (Figure 13)

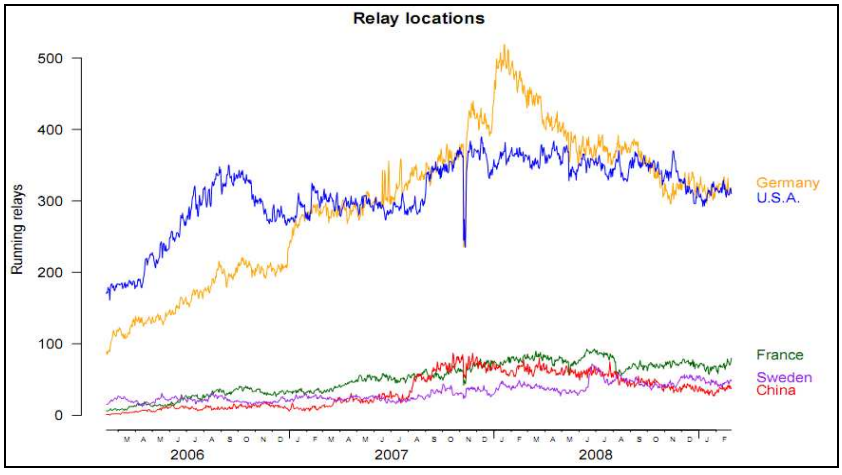

Figure 13: Top contributing countries to the Tor Network [19]

Duration

Regarding the duration a single Tor route was used, it can be confirmed that the Tor network is changing routes like described in the design: "*Tor will reuse the same*

circuit for new TCP streams for 10 minutes, as long as the circuit is working fine. A single TCP stream will stay on the same circuit forever, wihtout rotation, otherwise an adversary with a partial view of the network would be given many chances over time to link you to your destination" [20]

As shown in Figure 14 during the experiment more than 90% of circuits were changed within the first 11 minutes. The high percentage of instable connection at the beginning could be a reason for the poor performance at some points.

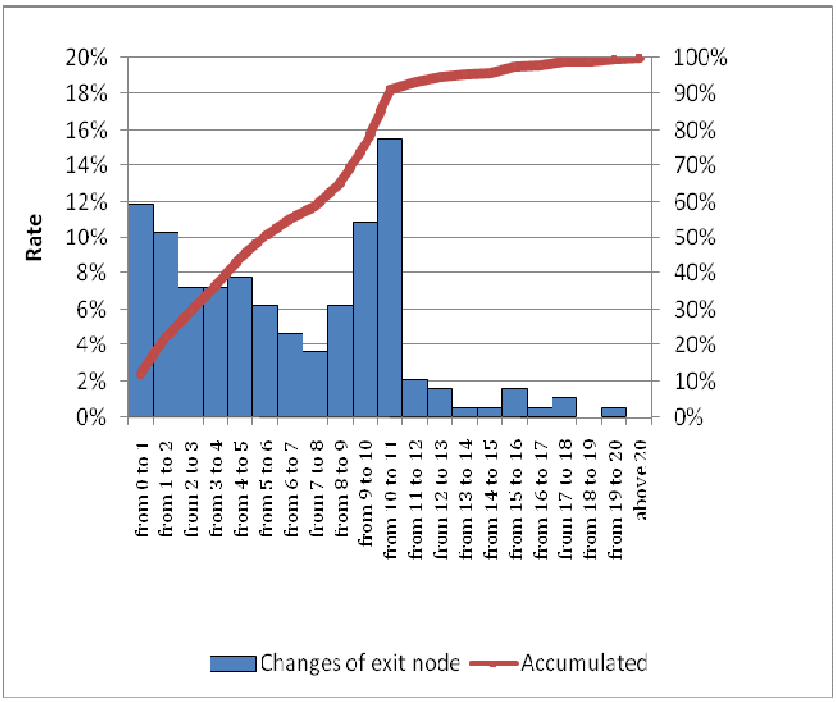

Figure 14: Change interval of Tor exit nodes

Problems / Further Research

Since all experiments have been made from Germany and considering that "*an adversary could break the anonymity, if he can observe both ends of the communication*", it is questionable, if a user should exit the Tor network in the same country or even region he entered it.

A shortcome of the provided data analysis is that it cannot be observed if two times in a row the same exit node is used, even if the route has changed. Additionally the durations are not 100% precise, since the exact moment of a route change could not be observed with the described experimental design. It could only be observed if after a test relevant HTTP request the IP has changed, but not if this happened before or after this request.

3.6 DNS Request Duration

Objective

As mentioned within the HTTP experiment the influence and actual speed of DNS requests could not be measured. This experiment aims to gather empiric data to give a figure regarding the performance losses of DNS requests while using Tor.

Execution

A Perl script was created, which loops over the top 500 websites, doing alternating DNS request with and without Tor using the dig command line tool. Instead of the standard DNS Server of the ISP, for direct requests the GooglePublicDNS server (with IP 8.8.8.8) was used via UDP, to make the experiment outcome reproducible in later experiments. For request over the Tor network a local DNS proxy was needed to forward UDP based DNS requests to Tor as TCP based requests. The TOR-DNS-Tool v1.7, which is recommended on the Tor homepage, was used for this purpose. When using Tor the DNS server is not known and depends only on the exit node's settings.
After one run with 500 requests the DNS cache of the operating system and of TOR-DNS was flushed manually to not distort the results of the next run.
Overall 20 runs have been made, so the data is based on altogether 10000 DNS request with Tor and 10000 direct requests.

Outcome

Tor is in the conducted experiments around 40 times slower with a much larger span of perceived values compared to direct DNS requests. (see Figure 15)

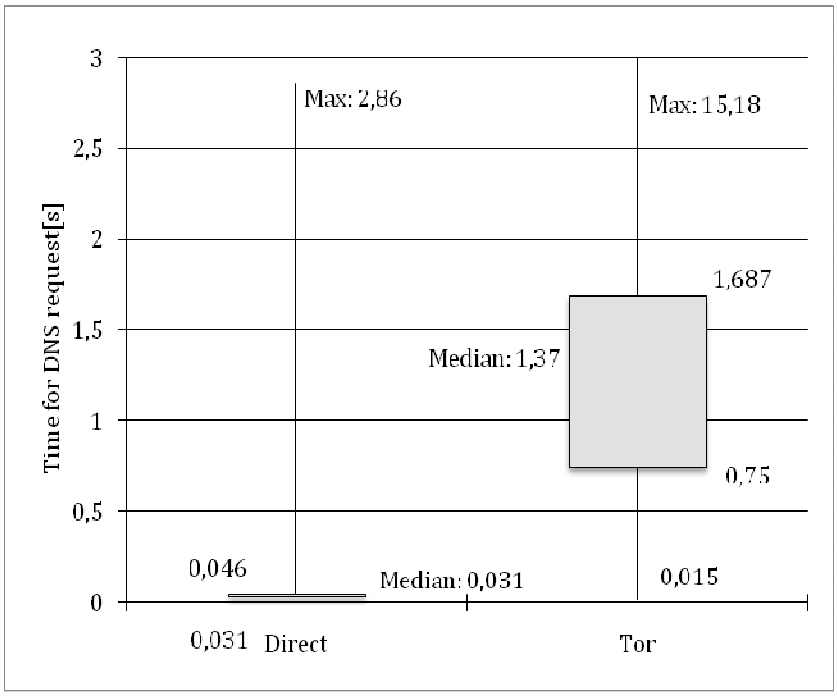

Figure 15: DNS Request Performance when using Tor

Figure 16 shows the average request time per domain, over all 20 runs and so offers a better visualization of the actual distribution of values.

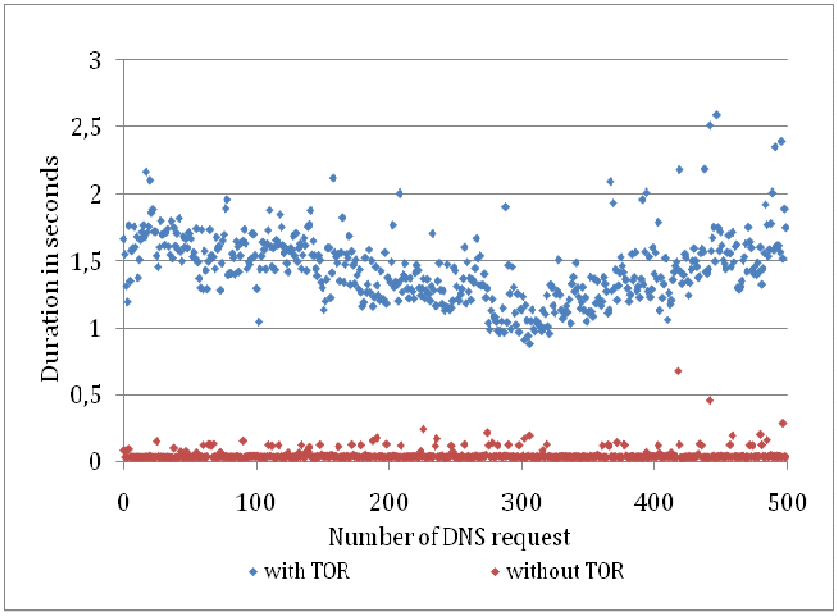

Figure 16: Average durations of 20 DNS requests to 500 different domains

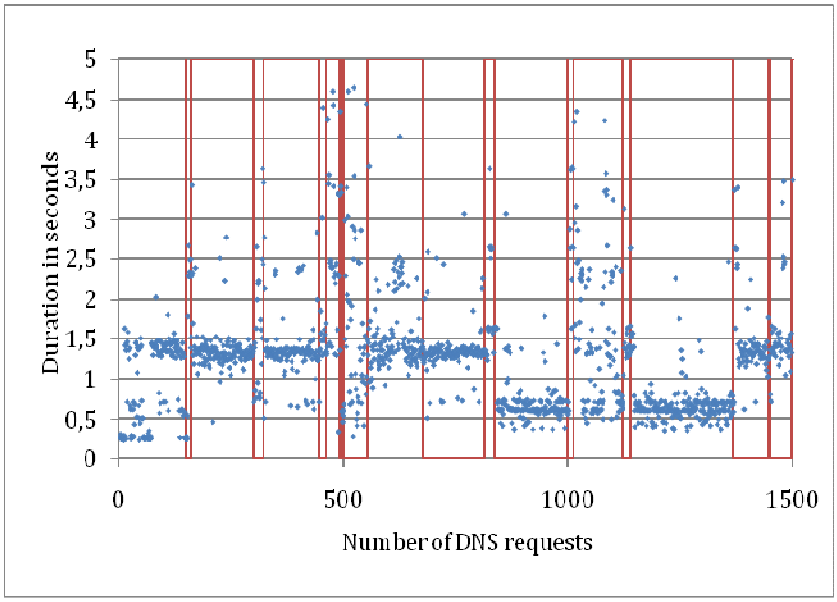

Figure 17: DNS request duration with changes of exit nodes

Furthermore it is noticeable that different Tor relays provide different DNS performances. In Figure 17 changes of exit nodes are marked by red lines (1500 DNS

request over 2 hours are shown). This experiment could not clarify, if this is because the used DNS server of exit node is slow or because of some slow nodes on the Tor route.

Problems / Further research

The same trend could be observed when using nslookup for this experiment. But the actual measured times are higher, because nslookup is not providing the time the DNS request took, why it was only possible to stop the duration from program start to end.

It would not match the reality from a users view, but would maybe improve the comparability to perform DNS requests also without Tor over TCP or repeat this experiment if TOR is one day available over UDP.[22]

4 Conclusion

In this paper the performance of communication while using Tor was evaluated. To classify the costs of anonymity, Tor's performance was directly compared to the performance without Tor. The experiments have proven and quantified the initial assumption that there are performance losses, when anonymizing traffic with the help of the Tor network. DNS und HTTP as two of the most important protocols have been measured deeply and quite precisely over Tor. For HTTP requests a factor of 5 and for DNS requests a factor of 40 could be observed by experienced results.

To gain empiric data it has emerged as a good way to apply Perl scripts which have enabled automated tests. The use of such scripts is recommended for further researches in this area. There is an etiquette to avoid transfers of massive amount of data over the Tor network. This should be considered when doing further tests in the future.

All tests were based on a list of URL's provided by SeoMoz. This URL's represent the 500 most linked websites in the Internet. It can be assumed, that tested target websites have a similar and optimized infrastructure and might be e.g. distributed to multiple locations (via content distribution networks). As a further approach it is conceivable to incorporate additional websites, which are e.g. less known or country-specific. Also a higher number of websites could be considered.

The reason for performance losses were not discussed within the scope of this paper. It needs to be said that gained results have not the intention to classify Tor as unusable. As the strong user base proves, Tor is usable for certain applications, like web browsing, but with costs of anonymity. The goal was only to classify the performance tradeoffs and so provide a base for future researches regarding usability.

References

1. Tor Project Homepage: Overview (2010)
   a) http://www.torproject.org/overview.html.en
   b) http://www.torproject.org/faq-abuse.html.en#WhatAboutCriminals
   (last accessed: 11-Feb-2010)
2. Jones, A.: Anonymous Communication on the Internet (2004)
   http://www10.cs.rose-hulman.edu/Papers/Jones.pdf (last accessed: 11-Feb-2010)
3. Loesing, K.: Messuring the Tor Network from PublicDirectoryInformation (2009)
   http://freehaven.net/~karsten/metrics/measuring-tor-public-dir-info-final.pdf (last accessed: 14-Jan-2010)
4. Vitone, D.: Tor – Anonymous Networks (2008)
   http://blag.cerebralmind.net/wp-content/uploads/2008/05/tor.pdf (last accessed: 11-Feb-2010)
5. Loesing, K., Murdoch, S.J., Dingledine, R.: A Case Study on Measuring Statistical Data in the Tor Anonymity Network (2009)
   www.cl.cam.ac.uk/~sjm217/papers/wecsr10measuring.pdf (last accessed: 11-Feb-2010)
6. D., Roger, Mathewson, N., Syverson, P.: Next generation Tor Release Paper, USENIX Association: Proceedings of the 13th USENIX Security Symposium (2004)
   http://www.usenix.org/events/sec04/tech/full_papers/dingledine/dingledine.pdf (last accessed: 11-Feb-2010)
7. Zetter, K., Nodes, R.: Turn Tor Anonymizer Into Eavesdropper's Paradise (2007)
   http://www.wired.com/politics/security/news/2007/09/embassy_hacks (last accessed: 11-Feb-2010)
8. Murdoch, S.J., Danezis, G.: Low-Cost Traffic Analysis of Tor, 2005 IEEE Symposium on Security and Privacy (2005)
   http://www.cl.cam.ac.uk/~sjm217/papers/oakland05torta.pdf (last accessed: 11-Feb-2010)
9. Murdoch, S.J., Zielinski, P.: Sampled Traffic Analysis by Internet-Exchange-Level Adversaries (2007)
   http://petworkshop.org/2007/papers/PET2007_preproc_Sampled_traffic.pdf (last accessed: 11-Feb-2010)
10. Dingledine, R., Mathewson, N.: Anonymity Loves Company: Usability and the Network Effect, The Workshop on Usable Privacy and Security Software (WUPSS) (2004)
    http://www.freehaven.net/doc/wupss04/usability.pdf (last accessed: 11-Feb-2010)
11. Abou-Tair, D., Pimenidis, L., Schomburg, J., Westermann, B.: Usability Inspection of Anonymity Networks, Proceedings of the 2009 World Congress on Privacy, Security, Trust and the Management of e-Business (2009)
    http://www.pimenidis.org/research/papers/usability-inspection.pdf (last accessed: 11-Feb-2010)
12. CyLab Usable Privacy and Security Laboratory: Symposium On Usable Privacy and Security
    http://cups.cs.cmu.edu/soups/2009/ (last accessed: 11-Feb-2010)
13. Loesing, K.: Measuring the Tor Network from Public Directory Information, 2nd Hot Topics in Privacy Enhancing Technologies (HotPETs), (2009)
    http://metrics.torproject.org/papers/hotpets09.pdf (last accessed: 11-Feb-2010)
14. Dingledine, R., Murdoch, S. J.: Performance Improvements on Tor or, Why Tor is slow and what we're going to do about (2009)
    https://svn.torproject.org/svn/tor/trunk/doc/roadmaps/2009-03-11-performance.pdf (last accessed: 11-Feb-2010)

15. Billen, K.: Der Tor-Speed (2006)
http://blog.kairaven.de/archives/899-Der-Tor-Speed.html (last accessed: 11-Feb-2010)
16. Tor Metrics Portal: Graphs (2010)
http://metrics.torproject.org/torperf-graphs.html (last accessed: 11-Feb-2010)
17. tsocks FAQs: tsocks doesn't seem to be working for ftp, why? (2010)
http://tsocks.sourceforge.net/faq.php#ftp (last accessed: 11-Feb-2010)
18. Privoxy FAQs: I cannot connect to any FTP sites. Privoxy is blocking me (2010)
http://www.privoxy.org/faq/trouble.html (last accessed: 11-Feb-2010)
19. Loesing, K.: Measuringthe Tor Network: Evaluation of Relays from Public Directory Data (2009)
http://freehaven.net/~karsten/metrics/dirarch-2009-03-31.pdf (last accessed: 11-Feb-2010)
20. Tor FAQs: How often does Tor change its paths? (2010)
https://wiki.torproject.org/noreply/TheOnionRouter/TorFAQ#HowoftendoesTorchangeitspaths.3F (last accessed: 11-Feb-2010)
21. Palme, J., Berglund, M.: Anonymity on the Internet (2002)
http://people.dsv.su.se/~jpalme/society/anonymity.html (last accessed: 11-Feb-2010)
22. Jones, A.: UDP-OR A Frair Onion Transport Design: A redesign of onion routing based that focuses on performance fairnes and scalability (2008)
http://www.petsymposium.org/2008/hotpets/udp-tor.pdf (last accessed: 11-Feb-2010)