I2P Usability vs. Tor Usability
A Bandwidth and Latency Comparison

Mathias Ehlert

Humboldt University of Berlin

Abstract. This paper analyzes latency and bandwidth drawbacks while using I2P to surf the Internet anonymously. Therefore tests are conducted utilizing scripts previously used to analyse the TOR network. The results gained in the tests for I2P are analysed and in a next step compared to previous results observed in TOR. The results show that I2P can only yield advantages to TOR when issuing simple HTTP-GET-Requests, but is inferior to TOR when browsing complete web sites and downloading files. Also the findings include that with the described set up, theoretical user cancellation rate while browsing the web using I2P is 100%.

Keywords: anonymity,anonymous web browsing, TOR, I2P, comparison, usability, latency, bandwidth

1 Introduction to I2P

I2P is a peer-to-peer (P2P) overlay network which allows participants to interact anonymously within this network. Similar to TOR (The Onion Router) I2P is a *mix network* which relies on routing data trough multiple peers to render tracing virtually impossible [1]. The project was first proposed in 2003, having its roots in the Invisible Internet Project (IIP), an anonymous real time communication project. Development has actively been going on since this time. Today I2P already offers a range of applications which can be used inside the network such as anonymous web browsing, chatting, file-sharing and e-mailing [2] [3].

1.1 Technical Introduction to the I2P Network

I2P's operation is based on the creation of tunnels at each participant (*router*). Since routes in I2P can only be used in one direction two types of tunnels need to be in place for communicating: *inbound* and *outbound* tunnels. In the process of setting up a tunnel for a specific protocol (HTTP, IRC, ...), users can determine how many *hops* (additional steps) should be used. This enables balancing between security and latency as desired. As soon as the inbound tunnel is established, information about it has to be made public in I2P's "*network database*" (NetDB). The NetDB is used to store information about how to contact a specific user. Therefore the "Inbound Gateway", the entry point of the inbound tunnel leading to the actual user, is stored for each participant. The typical process of a client establishing connections in I2P consists of the following steps:

1. Build own tunnels (inbound and outbound)
2. Query NetDB to find destination's inbound tunnel gateway
3. Use own outbound tunnel to send message
4. Message is routed through the outbound tunnel to the gateway node (last node of the tunnel)
5. Outbound tunnel gateway forwards message to the destinations's inbound tunnel gateway
6. Inbound tunnel gateway node forwards the message to the actual recipient

Currently there is no distributed DNS in place inside I2P. A `hosts.txt`-like approach is used to store mappings of domain names (`*.i2p`) to destinations in users' *addressbooks*. Layered encryption of the messages is realized through *garlic encryption*. The sender encrypts its message multiple times using the public keys of each node on the message's route. Each node then decrypts one layer of encryption using its own private key, revealing where to forward the message. For a more detailled description of I2P's inner workings see I2P's tech intro ([3]).

1.2 Differences to TOR

The main characteristic in which I2P differs from TOR is the focus of the project. While TOR was designed with the intention to enable anonymous internet browsing, I2P's focus is to provide an anonymous network, isolated inside the internet, offering various protocols and applications within. It is however possible to utilize I2P *outproxies* to reach the internet and thus enable anonymous web browsing. Due to the focus of the project, there was only one public outproxy in place at the time of this writing (`false.i2p`). Another difference to the TOR network is, that I2P tunnels can only be used unidirectional compared to TOR's bidirectional tunnels. Also, as mentioned in Section 1.1 I2P uses garlic encryption where TOR uses onion encryption. Both implementations are based on layered encryption, garlic encryption offering the possibility to store multiple messages inside the innermost layer [3].

2 Test Scenario

Naturally the technique used to anonymize traffic through I2P yields additional latency and bandwith loss with every hop taken. The main task of this work is to quantify average latency and bandwith when accessing websites through an I2P outproxy and to compare the results with previously measured values for TOR. To achieve this, a testing environment has been set up consisting of two machines connected to the I2P network. One machine configured to function as a dedicated outproxy and one client set up to use the two outproxies (the public one and the dedicaded one).

2.1 Outproxy

The first task while setting up the environment was to put an own outproxy into the I2P network for dedicated use in these tests. This was done to be independent from the public outproxy. Since there is only one, chances are it might be under heavy load if multiple users access it at the same time, thus distorting measurement results. Used was a Virtual Machine inside the Humboldt University network, Berlin Germany. On this VM, a Squid proxy was set up with caching disabled in order to measure actual bandwidth to destinations. Also the machine was connected to I2P and a local tunnel was created, routing requests from inside I2P to the local squid, thus accessing the internet. The fact that the machine's address can be used as an outproxy was not published, to make sure access to it was exclusive. When setting up the HTTP tunnel, default values were used for tunnle length (2 hops), variance of tunnel length (0), and all other values.

2.2 Client

On the other end, a common asymmetric 16 MBit/s downstream, 1 MBit/s upstream DSL has been used. On a Windows 7 host , Ubuntu was running inside a Virtual Machine. The host system had an Intel i7 M620 (2,67GHz) processor and 8 GB of RAM. The VM was assigned 1 CPU core and 2 GB of RAM. I2P was installed on the VM and the outbound HTTP proxy was set up with standard values concerning tunnel lenght and variance via the I2P web interface. The web server is automatically launched when `i2prouter`, I2P's main program, is started. Both outproxies were specified, the public one defined by its hostname `false.i2p`, and the dedicated one by using its unique *base32* name (52 characters + .i2p) [4]. That way, I2P cycles through both proxies in steps of about 7 minutes.

2.3 Scripts

In order to quantify latency and bandwidth objectively, a set of perl scripts was used to measure:

1. HTTP GET Request durations (CORE Latency)
2. Duration of downloading whole webpages including images and external ressources (AVERAGE Latency)
3. Speeds when downloading certain amounts of data (Bandwidth)

The perl scripts were written and previously used by Fabian et al. [5] to analyse latency and bandwidth inside the TOR network. The scripts simulate typical user requests while browsing websites and store measured values. As example destinations, a list of the 500 most visited web sites according to `http://www.seomoz.org` was used. The targets were accessed directly and via I2P alternately.

3 Results

The following section will present the results of the measurements. A box-diagram design was chosen, in order to show the distribution of recorded values. The graphs are based on minimum and maximum values, (whiskers) as well as .25 and .75-quantiles (lower and upper bounds of boxes).

3.1 CORE Latency: HTTP-GET-Requests

The first measurements which were done, aimed at CORE latency of visiting web sites, meaning only a HTTP-GET-Request to the specified target. Figure 1 displays CORE latency when accessing ressources directly. Data has been limited to 20 web sites for better visibility. With a direct connection an average core latency of 3.31 seconds was measured on the set of 20 web sites.

Figure 2 contrasts CORE latency when anonymously browsing via I2P. It is

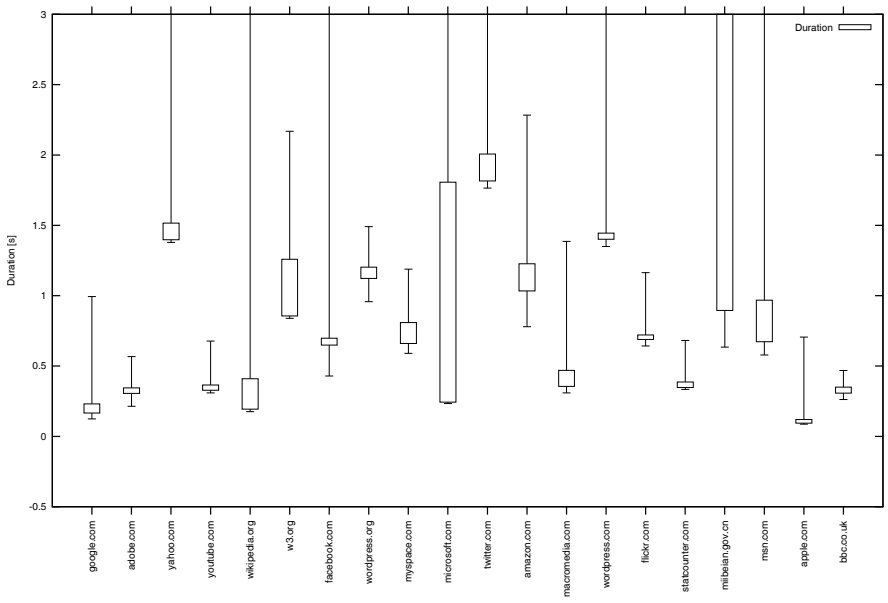

Fig. 1. CORE Latency Direct (Top 20)

observable that latency increases sharply by a factor of around 3 to an average of 10.07 seconds. Finally, the complete set of data from accessing the Top 500 list was used to render Figure 3. Here the overall level of additional latency of I2P to HTTP-Requests becomes apparent. When 25% of all direct requests took less than 0.64 seconds, the .25-quantile of I2P requests lies at 2.16 seconds. Furthermore the .75-quantiles were at 1.45 seconds (direct) and 6.04 seconds (I2P).

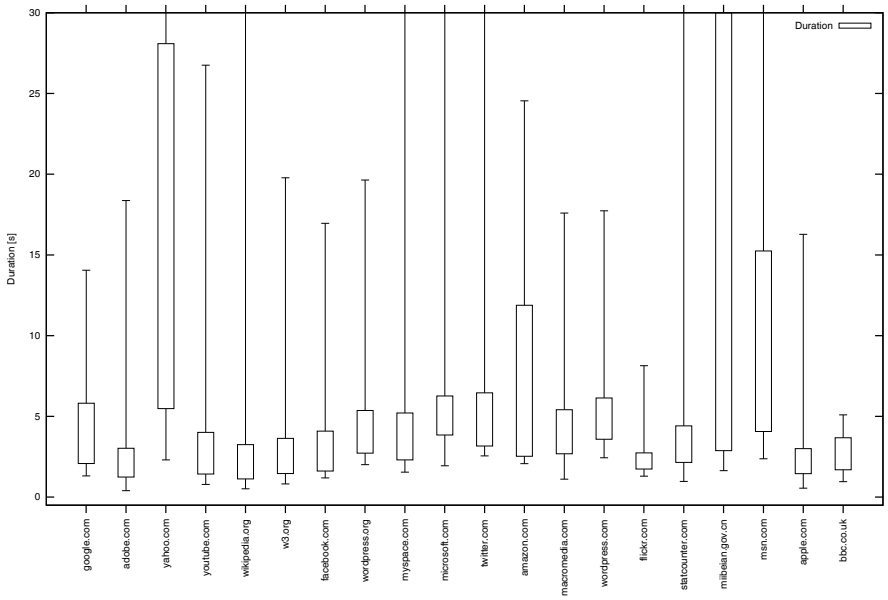

Fig. 2. CORE Latency I2P (Top 20)

Also, when evaluated individually, only a slight deviation can be identified between the publicly available outproxy (`false.i2p`) and the outproxy set up for this work. The latter was slightly faster with an average of 9.33 seconds compared to the public proxy with 12.07 seconds latency on average.

3.2 AVERAGE Latency: Complete Webpages

During the next tests the duration of accessing whole web pages was analyzed, including neccessary images and other ressources. So called AVERAGE Latency for direct access of the first 20 sites is displayed in Figure 4. A clear variation between simpler and more complex websites (google.com vs. adobe.com) can be observed in the graph. Conducting the same tests through an I2P-tunneled connection yields the results shown in Figure 5. The scale of the Y-axis has been expanded by a factor greater than 10 to properly hold the bigger latency values. To contrast both direct and tunneled latency values, Figure 6 displays minimum and maximum values, as wel as .25 and .75 quantiles. The graph points out, that while 75% of all direct visits took less than 13.818 seconds seconds, I2P increased this value to 226.854 seconds, yielding a factor of more than 16. In addition, 25% of websites accessed via I2P were retrieved only faster than 14.39 seconds while with direct access' .25 quantile was measured at 2.957 seconds.
Comparing average values of the two I2P outproxies to each other, a slight difference can be observed: the public one seems to be a little faster with a median of 157.51 seconds, while the dedicated outproxy average amounts to

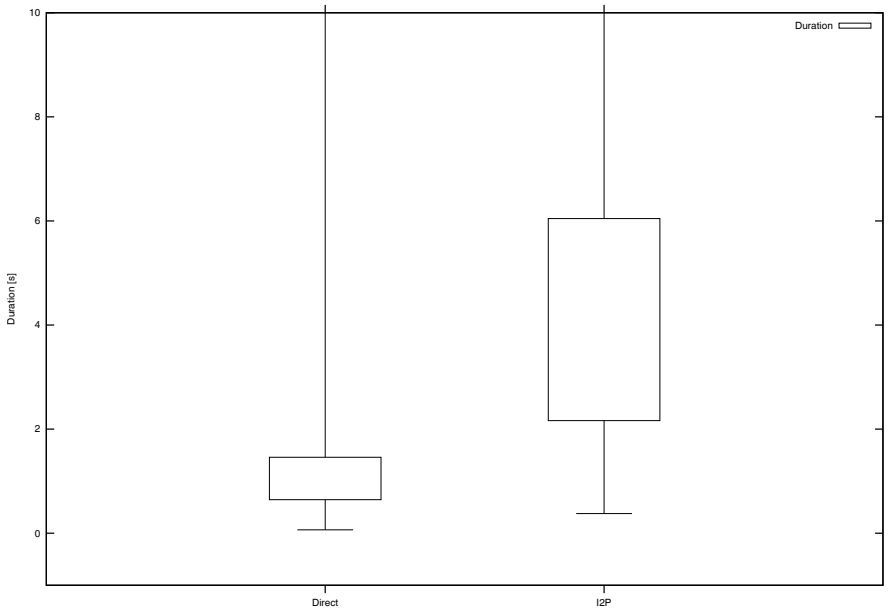

Fig. 3. Comparison: CORE Latency, Direct - I2P

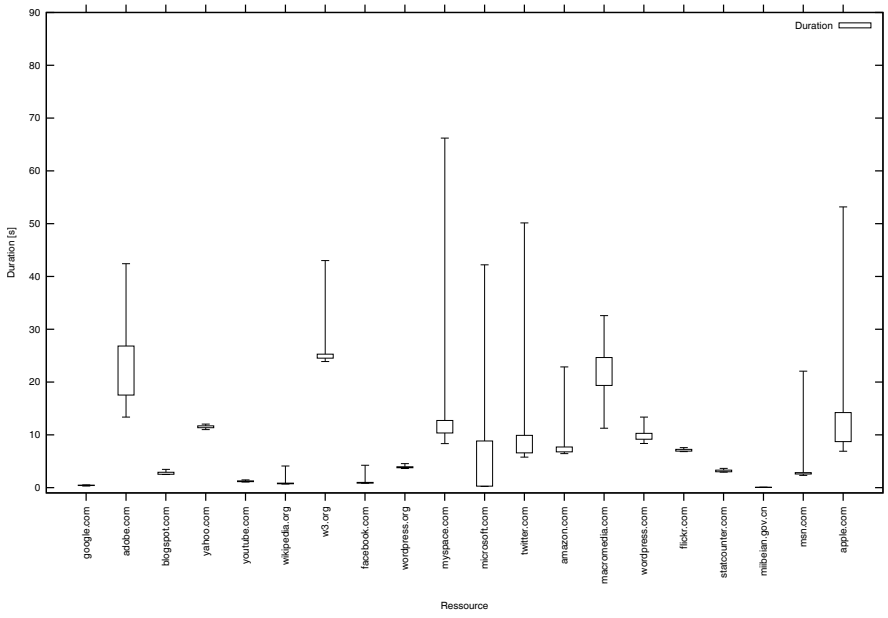

Fig. 4. AVERAGE Latency Direct (Top 20)

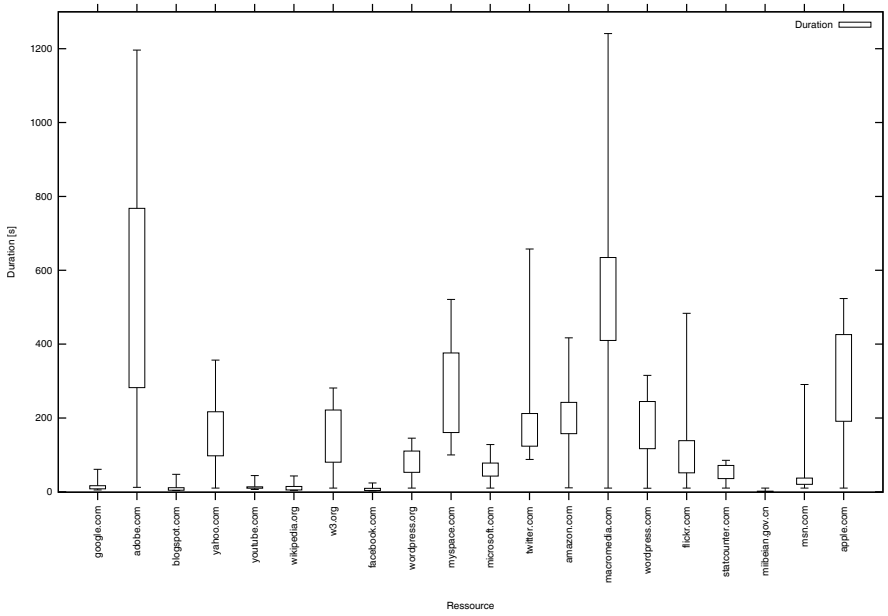

Fig. 5. AVERAGE Latency I2P (Top 20)

215.54 seconds. This could be an effect caused by caching at the former but was not analysed in more deeply in this work.

3.3 Download Speeds

In a last test, download bandwidth was measured. Parts of a file in a fixed location were downloaded repeatedly with increasing part sizes (50-1000 kB), each time recording download speed. As expected the differences depicted in Figure 7 are significant. Average download speeds on the direct connection were observed at 937.90 kB/s while the average bandwidth via I2P dropped to 31.28 kB/s. It is still worth noting that the maximum download speed reached through I2P was 277.82 kB/s. Furthermore when analysing details about the distibution of bandwidth values it becomes apparent that the .75 quantile amounts to 38.39 kB/s via I2P in contrast to 1224.18 kB/s achieved directly.
It is also worth noting, that here as well, no significant discrepancies could be identified when analysing results grouped by outproxy. The average download speed via the public outproxy was 33.35 kB/s, while the dedicated outproxy yielded 29.63 kB/s on average.

4 Comparison to TOR Measurements

Comparing usability factors such as latency and bandwidth of I2P to TOR is the objective of the next section. This shall be done by relating TOR measurements,

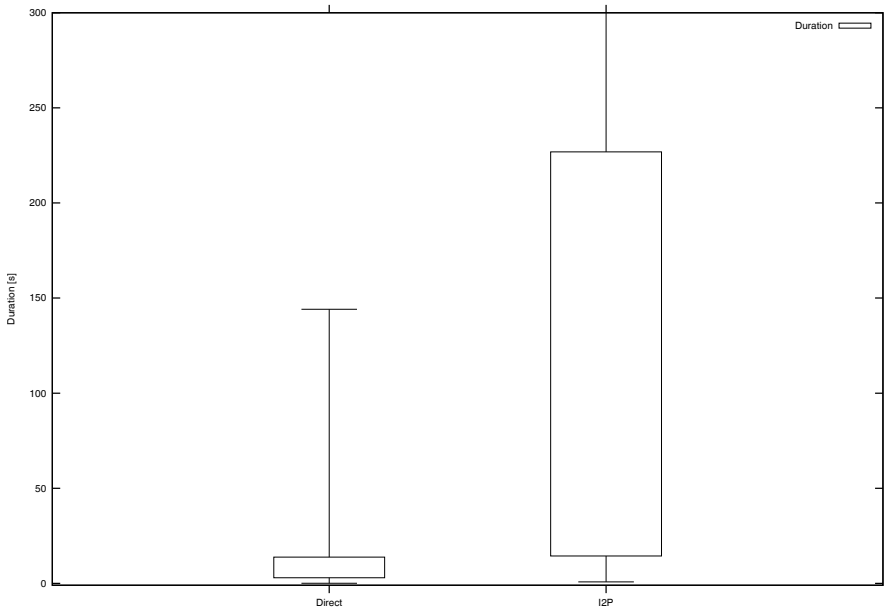

Fig. 6. Comparison: AVERAGE Latency, Direct - I2P

previously presented by Fabian et al. [5], to the newly acquired I2P results described here in Section 3.

4.1 Comparison of CORE Latency

The candlestick plots of Figure 8 show how on average CORE latency in I2P is considerably below the TOR value. The median latency values for HTTP-Requests lie at 3.49 (I2P) and 7.08 (TOR) seconds. For I2P .25 and .75 quantiles are 2.16 and 6.04 seconds respectively, whereas Fabian et al. previous observed values of 4.00 and 12.61 seconds for TOR.

4.2 Comparison of AVERAGE Latency

Observed AVERAGE latency figures of both I2P and TOR are contrasted in Figure 9. Standing out is the fact that unlike with CORE latency, AVERAGE latency in I2P is higher than TOR values. The .25 quantiles lie close to each other at 9.60 (TOR) and 14.39 (I2P) seconds, but medians and .75 quantiles show a notable difference. While 50% of TOR requests are completed in less than 16.99 seconds, half of the I2P requests take 103.19 seconds, which amounts to around 500% of additional latency. A similar tendency can be observed for the .75 quantiles of 30.26 (TOR) and 226.85 (I2P) seconds.

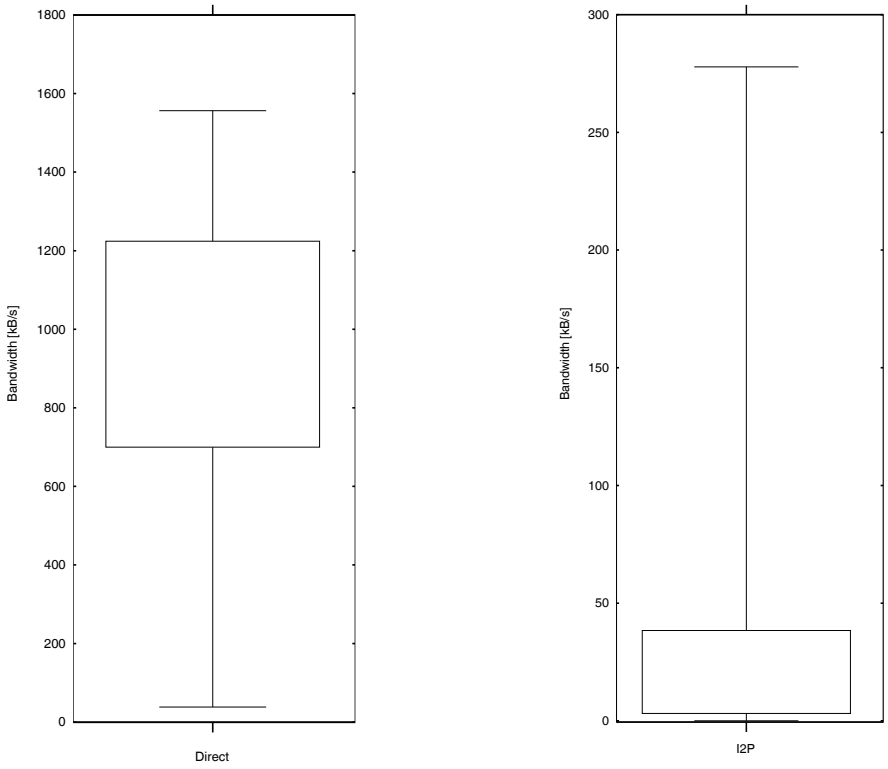

Fig. 7. Comparison: Download Bandwidth, Direct - I2P

4.3 Comparison of Download Speeds

The last comparison of this chapter focuses on bandwidth while downloading a single file. The results are visualized in Figure 10. Again average download bandwidth of TOR is above I2P's. While TOR's median lies at 51.62 kB/s, I2P's median value only amounts to 12.91 kB/s.

5 Conclusion

The comparisons in the last chapter pointed out, that while using I2P one can achieve better results for CORE latency (HTTP-GET-Requests) than via TOR. Still, AVERAGE latency and download bandwidth provided favorable results via TOR.
The reasons for the discrepancy between the findings could be explained by the different structures of the two services. TOR is a widely known tool, thus having a large number of users, distributed all around the globe, being the de-facto standard anonymity tool. In contrast, I2P is less known and has a smaller user base, which might be geographically more centered around Central Europe.

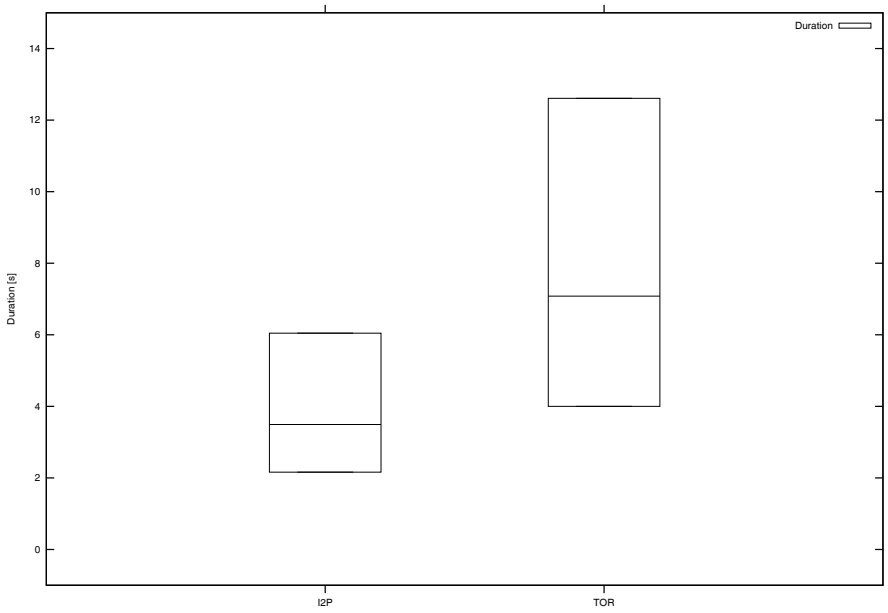

Fig. 8. Comparison: CORE Latency, I2P - TOR

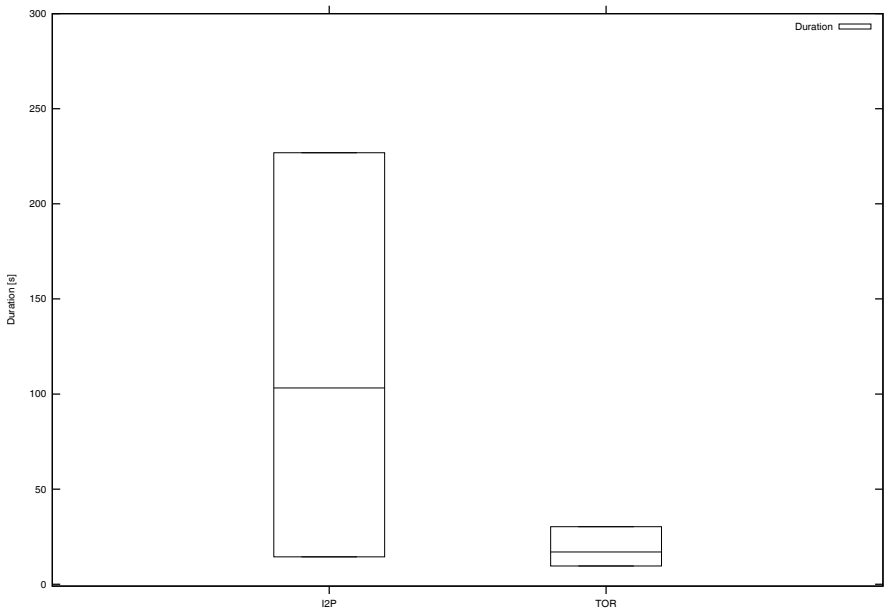

Fig. 9. Comparison: AVERAGE Latency, I2P - TOR

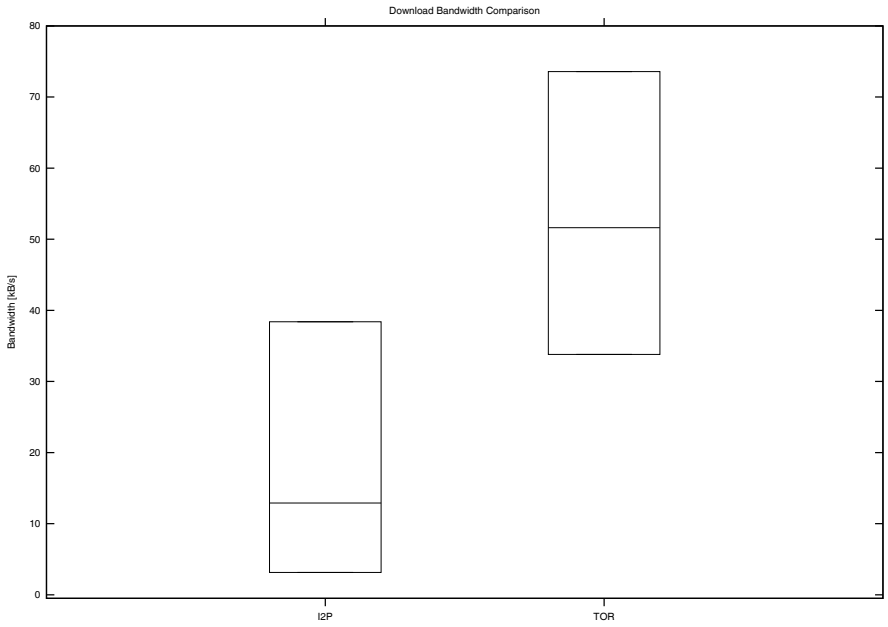

Fig. 10. Comparison: Download Bandwidth, I2P - TOR

This would mean requests in TOR could potentially have to travel longer routes, although only 3 hops are taken in between a TOR user and a TOR exit node. When most users of I2P are located in, or around germany and Central Europe, routes could be shorter on average, although usually 5 hops are typical in I2P (2 for the own outbound tunnel, plus 3 for the outproxies's inbound tunnel).
In cases where bandwidth plays a larger role, like AVERAGE latency (complete webpages) and download measurements, TOR could benefit from the fact that only 3 additional hops are taken. In both settings the peer with the lowest bandwidth determines the overall throughtput. In a shorther route chances might be lower to encounter a very slow peer. Another reason for the higher throughtput in TOR might be differences in encryption overhead in the two services.
Furthermore, taking into account the sudies of Nah [6] and Fabian et al. [5], the median value of I2P's AVERAGE latency of 184.59 seconds can be mapped to a cancellation rate of 100% (TOR: 88%) rendering I2P not suitable for anonymous web browsing offering decent usability under the given conditions at the time of this writing.

6 Future Work

With the results of this work, several possibilities emerge to expand research about the topic of latency benchmarking in I2P. While it is unlikely that the team of researchers and programmers behind I2P will focus further development

on improvement of browsing through an I2P outproxy there are ways in which results could potentially be improved. As noted in Section 2.2, default values were used when setting up the required tunnels. Part of future research could include analysis and quantification of the effects, different tunnel lengths have on latency.
Also, latency measurements for this work were done over a period of several days, at random days of the week and times of the day. This approach could also be improved by systematizing the starting times and durations of test runs. In order to improve comparatibility, a larger set up, measuring I2P and TOR at the same time, is also imaginable.
Since DNS-Requests are also part of browsing web pages, Fabian et al. [5] included additional latency of tunneled DNS requests in their TOR analysis. Setting up a DNS tunnel in I2P and including the according measurements in future works would also be a desirable addition.

References

1. zzz (Pseudonym), and L. Schimmer: Peer Profiling and Selection in the I2P Anonymous Network PET-CON 2009.1.
2. www.invisiblenet.net (expired): Invisible Internet Project (I2P) Project Overview `http://www.i2p2.de/_static/pdf/i2p_philosophy.pdf`
3. I2P Staff Introducing I2P: A scalable framework dor anonymous communication `http://www.i2p2.de/techintro.html` (last accessed Feb. 13th 2011)
4. I2P Staff Naming In I2P `http://www.i2p2.de/naming.html` (last accessed Feb. 13th 2011)
5. Benjamin Fabian, Florian Goertz, Steffen Kunz, Sebastian Mller, Mathias Nitzsche Privately Waiting A Usability Analysis of the Tor Anonymity Network 16th Americas Conference on Information Systems
6. Nah, F. F. A study on Tolerable Waiting Time: How Long are Web Users Willing to Wait (2004) Behaviour & Information Technology