Performance Comparison of low-latency Anonymisation Services from a User Perspective

Rolf Wendolsky, Dominik Herrmann, Hannes Federrath

University of Regensburg, 93040 Regensburg, Germany

Abstract. Neither of the two anonymisation services Tor and AN.ON clearly outperforms the other one. AN.ON's user-perceived QoS is generally more consistent over time than Tor's. While AN.ON's network latencies are low compared to Tor, it suffers from limitations in bandwidth. Interestingly, Tor's performance seems to depend on the time of day: it increases in the European morning hours. Utilising AN.ON's reporting of concurrently logged-in users, we show a correlation between load and performance. The reported number of users should be adjusted, though, so that it serves as a better indicator for security and performance. Finally, the results indicate the existence of an overall tolerance level for acceptable latencies of approximately 4 seconds, which should be kept in mind when designing low-latency anonymisation services.

1 Introduction and motivation

Several anonymisation services for low-latency communication have grown up from research projects recently: among them are the well-known systems AN.ON [3] and Tor [17]. This paper focuses on the performance of the services for web surfing from a user perspective.

Although AN.ON and Tor are based on common building blocks (e. g. so called *mixes* [6], which relay multiply encrypted traffic from a client to a server), they differ in various technical attributes such as structure, threat model and application range. AN.ON uses a limited set of *cascades*, each consisting of predefined mixing nodes. In contrast, Tor relies on a large amount of nodes from which random *circuits* are constructed in real-time. As the user base is usually hundreds or thousands of times bigger than the amount of nodes used for relaying traffic, performance issues may arise.

It has been shown that performance, especially latency, is an anonymity-relevant parameter [11]. We can assume that many users are not able to evaluate the real security of an anonymisation service [5]. Therefore, their decision to use a specific service may highly depend on its overall performance: Only few people are willing to use a slow service, and, regardless of any sophisticated cryptographical techniques, such a service might not

provide any anonymity at all. Consequently, the performance from a user perspective might serve as an important indicator for the overall quality. Moreover, performance evaluations can be used to identify characteristics of the different approaches, and – obviously – they allow the evaluation of tuning measures.

In this paper, we will provide an empirical study regarding the relation between performance and the number of concurrent users. Based on that we will present the results of a comparison of AN.ON and Tor from a user perspective and try to explain the source of any differences found. We will show that a naïve comparison of average throughputs and delays is hardly sufficient, but conclusions can be drawn with the help of inferential statistics nevertheless. Our results indicate the existence of an overall *performance threshold*. This means that users are not willing to use a service which fails to meet this threshold.

We will introduce the evaluation scenarios for our performance tests in section 2 and present our methodology for data collection in section 3. Section 4 contains a short description of the statistical methods used during analysis. The results of our evaluation of AN.ON and Tor are presented in section 5. We suggest areas for future research in section 6, while section 7 summarizes our findings.

2 Performance indicators and evaluation scenarios

In this section we will present the relevant performance indicators and our evaluation scenarios. For the performance evaluation of the anonymisation services, we simulate the behaviour of a typical WWW-user who (1) requests web sites and (2) downloads files. We identified two performance indicators, namely *latency* and *bandwidth*.

The *bandwidth* (KBytes/s) indicates how fast data packets may be transmitted on a communication channel. The *latency* (milliseconds) corresponds to the roundtrip time of a network packet. Ideally, the latency is independent from the bandwidth. For large files it is almost irrelevant, whereas retrieving a web site (with many small objects) can be slowed down by high latencies substantially.

In order to profile the aforementioned indicators, we set up different *scenarios*. A scenario is characterised by two parameters: *type of simulation* and *URL language*. The *type of simulation* is either (1) a test with different web sites containing a (large) number of small objects {WEB}, or (2) a test with fixed-size downloads {DL}. The separation into different *URL languages* is a heuristic method to measure system performance in

a local area, e. g. Germany, or world-wide. For our research, we split the tests into German {DE} and English {EN} content language. While the English pages can be used for a fair comparison of different anonymisers, the German sites allow profiling the AN.ON service from a local perspective.[1] The URLs were chosen from the most popular web sites according to Alexa [2] and the downloads according to downloads.de/downloads.com respectively (cf. table 7). Table 1 lists the basic scenarios.

Table 1. General attributes of the basic scenarios

Simulation	WEB		DL	
Language	DE	EN	DE	EN
Total URLs / scenario	11	14	3	3
Average requests / scenario	398	309	3	3
Average requests / URL	33.17	20.6	1	1
Average KBytes / scenario	1267	987	1520	1702
Average KBytes / URL	105.58	65.8	506.67	567.33

3 Data collection methodology

In this section we will describe our methodology for collecting performance data from anonymisation services based on an example of the Tor network and AN.ON. We will start off with an overview of our evaluation setup and the evaluated services. The major part of this section will present our data quality measures.

3.1 Test suite overview

There are some free tools available to measure proxy or server performance [10, 15]. Unfortunately, they proved not suitable for the evaluation of anonymisation services. They focus on other applications and consequently lack important features such as failure tolerance. In the end, we decided to write a test suite specifically designed to meet our needs.

As we evaluate the services from a user perspective, the two performance parameters mentioned, *bandwidth* and *latency*, cannot be determined exactly: There are too many influences not under our control. Therefore, we approximate the performance of the services with the

[1] All current AN.ON servers reside in Germany, whereas Tor is distributed throughout the world.

help of the two observable parameters *throughput* and *initial delay*. The throughput is calculated by dividing the amount of received bytes by the time needed for the data transmission. The initial delay is the time difference between sending the HTTP request and receiving the first chunk of the response.

Our test suite `perfeval`[2] is written in Perl (about 2.500 lines of code)[3]. The scripts retrieve a set of URLs via HTTP (non-recursively) and calculate throughput and initial delay for each HTTP request. All recorded data of a session is aggregated into a *test case*.

We utilise the Perl library *LWP::ParallelUA* [12] which can handle simultaneous connections. Thus, we are able to simulate the behaviour of a web browser: First, `perfeval` downloads the HTML page, and then it fetches all the embedded objects in parallel. In order to prevent proxies or web caches from influencing the results we send a *Cache-Control:no-cache* HTTP header [14] along with the request.

3.2 Scope of the Evaluation

Table 2 lists the three services we evaluated with `perfeval`. In the rest of this paper we will refer to them with the presented acronyms. We also use a control connection (DIRECT) for assessing the performance of the Internet connection used during testing.

Table 2. Evaluated systems

DIRECT	Direct web access without any proxy
TOR	Tor client v0.1.0.16, Privoxy v3.0.3
DD	AN.ON cascade *Dresden-Dresden* (JAP v00.05.078)
CCC	AN.ON cascade *Regensburg-CCC* (JAP v00.05.078)

Privoxy was configured with the option *toggle 0* in order to disable all of its filtering rules. The two mentioned AN.ON cascades were chosen because of high stability and high number of users at the time when we started the test.[4] The test run started on February 15 2006, 6:00 p. m.,

[2] We were running the test suite on two WindowsXP workstations with ActivePerl v5.8.7.815 [1]. The workstations were connected to the Internet directly and had public IP addresses.

[3] http://www.jondos.de/downloads/perfeval.zip

[4] At that time the remaining two AN.ON cascades were used for testing purposes only, and were neither stable in structure nor in code.

and ended on February 26 2006, 11:59 a. m. (both Berlin local time [5]) by manual interruption. Thus, we got test data for 10 complete days and 18 hours, that corresponds to 258 hour-based test cases for each combination of scenario parameters and tested systems. We therefore have 4128 test cases altogether.

For the scope of this article an individual web site or a file download is represented by its URL. Each URL may lead to a number of HTTP requests: Typically, a web sit causes additional requests (for the HTML page and all its embedded objects), whose number typically differs over time, whereas a download causes exactly one HTTP request.

3.3 Data quality measures

In order to get statistically utilisable results for measuring the tested services, the collected data should not be considerably influenced by

(a) external factors jeopardizing the validity of the test cases like downtimes of the network, downtimes and failures of services, HTTP errors reported by web sites, and errors in the evaluation software itself,
(b) bias introduced by the observation itself like concurrent tests on the same anonymisation service, concurrent test requests of the same resource, and performance fluctuations on the computer where the test software runs,
(c) influences through fluctuations during the test like performance fluctuations of requested resources and fluctuations of the total amount of requested data,
(d) performance tampering through HTTP redirects,
(e) performance limit introduced by the Internet conection,
(f) varying performance throughout the day.

These influences have to be mitigated before and during the test. After that, the collected data must be examined for influences by the aforementioned factors. If at least one of those has a *non-negligible* influence, the corresponding data is probably not usable for any statistical analysis. We assume an influence as *non-negligible* if the ratio of (possibly influencing) "critical" cases to "good" cases is higher than 5%.[6]

In short, we found that our test data is of high quality regarding these measures. A more detailed description of our approach to measure data quality is presented in the following sections.

[5] Note that Germany has one single time zone.
[6] Note that this is a heuristic approach. The quality measures are **ratios** and not probabilities as in statistical tests.

External factors Single erroneous test cases resulting from a bad implementation of the test software may be discovered by looking for extreme values in the number of HTTP requests (which should be the same for each test case), the initial delay and the throughput.

HTTP errors, service failures, network and service downtimes may lead to missing or unintentionally influenced cases. For each unsuccessful HTTP request (i. e., the status code of the HTTP indicates a failure), we have to determine whether the source of the problem is the webserver or the network (i. e., the anonymisation service or the Internet connection). We will refer to the former as *errors*, to the latter as *failures*. This differentiation is important to measure the "quality" of an anonymisation service. Our software implements a sophisticated algorithm to differentiate errors from failures:

An unsuccessful HTTP request will be flagged as an *error*, if all of the following conditions apply immediately after the HTTP response has been received:

- a connection to the webserver/proxy can be established successfully
- a HTTP test request can be sent over the network
- a corresponding HTTP response is received
- the HTTP status code is not *200 OK* (or something similar)
- the HTTP status code is not *502 Service temporarily overloaded*, or *503 Gateway timeout*

Otherwise, the unsuccessful request is probably a *failure*, but further examinations are necessary. This is especially true for responses with status codes *502* and *503*, which can be issued by the webserver as well as by the proxy server. If the webserver is the originator, the request should be flagged as *error*, otherwise as *failure*. Timeouts, i. e., delays exceeding 60 seconds, are the most common type of *failures*.

Table 8 lists the number of cases missing either due to software errors or because of network or service downtimes. Compared to the total number in the sample, they are negligible. It also shows that almost all *failures* occur for DD, but as less than 5% of all requests are affected, we still treat external influences as negligible. This finding indicates hardware or network problems on the AN.ON DD cascade, though. Its operators have not been aware of that until now.

The number of *errors* is uncritical for all but one case: the error ratio on the CCC cascade for English downloads is about 9%. That means that a lot of downloads were skipped, probably due to service-specific blockings by the web site operators (e. g. by blacklisting the IP of the last

mix of the cascade). Nevertheless, this influence is limited to reducing the sample size for this service.

Bias introduced by the observation itself The tests for web surfing / downloads together were composed to be completed in less than 30 minutes for each language. In order to force comparable and periodic hour-of-day-based time intervals from 0 to 23 (Berlin local time), we put a hard limit of 60 minutes on the total duration of a language test. For each test case, all URLs were processed sequentially so that no interference between them was possible[7]. As the DE and EN tests should not interfere with each other, we performed these test cases on two separate machines, the latter one starting with a time offset of 30 minutes. Figure 1 shows the course of events during the performance evaluation.

Table 8 shows that the hard limit of one hour was never reached in our experiment and that a 30-minute-overlapping did not occur more often than in 5% of the test cases. These influences are therefore not seen as critical.

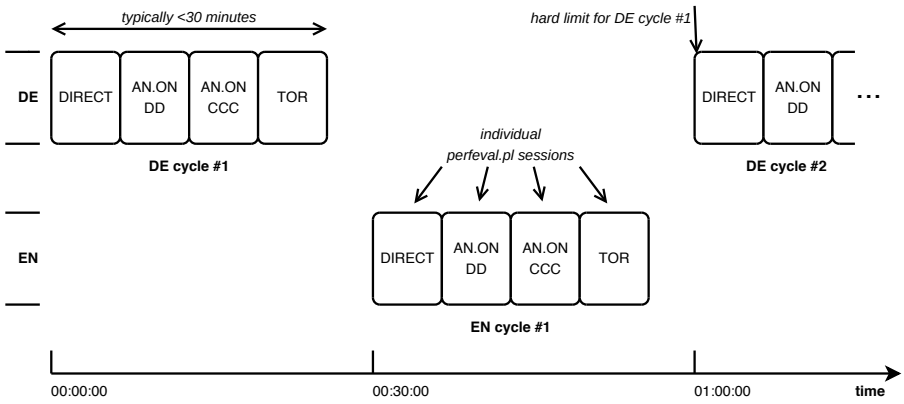


Fig. 1. Test sequence for performance evaluation

Influences through performance fluctuations In order to avoid performance influences from slow web servers that could lead to wrong conclusions in the analysis, the measurements of the individual URLs are

[7] Note that HTTP requests for each requested web site are done concurrently, but this is what a typical web browser would do as well.

aggregated into one *test case* for each scenario. Accordingly, we do not try to evaluate the service performance regarding single URLs (although this would be possible with our result files, of course).

Another possible influence is related to the amount of data received in each test case. To make the cases of one scenario comparable, they should be of equal size. We compared the median and the interquartile range[8] (IQR) of the downloaded bytes for each service with the median and IQR of all services to analyse this influence.

Table 8 shows that the ratios of all medians are negligible. Although there are some problems with English downloads (causing a huge IQR ratio for the CCC cascade), they do not affect the median. Therefore, our analysis suggests that we have indeed collected similar amounts of data for the different services.

Note that measuring performance fluctuations within the infrastructure of the anonymisation service is beyond the scope of this paper. In particular, we are not trying to measure the performance of individual nodes or one anonymity service as a whole. For Tor, we have to trust the node selection algorithm of its client software – we are looking at performance from a user perspective after all.

HTTP redirects Our evaluation software honours HTTP *301* and *302* redirect status codes. Although this behaviour is necessary for the imitation of a web browser, it introduces a new challenge: Our test might be influenced by server-side redirects *(geolocation)*, which would undermine the geographic separation introduced by the *URL language* scenario parameter.

It is rather difficult to rule out this influence completely as we cannot control the behaviour of the web servers. Of course, our software does not send any *Accept-Language* headers which would give away any information about its location or preferred language, nor does it interpret JavaScript code in the HTML pages which could be used to query language-specific browser attributes. But there are still more sophisticated ways for geolocation, for example by querying the WHOIS database for the IP address of the sender of the HTTP request. Obviously, it is impossible to fully prevent a webserver from delivering adapted versions of the requested content to the client. It has been observed that Tor (with its world-wide network of exit nodes) is subject to this phenomenon [18].

[8] The interquartile range is the difference of the upper 75% and the lower 25% quartile of the bandwidth. It is a robust measure for the standard deviation of frequencies.

We screened the evaluation data to make sure that no geolocation was employed, though.

Note that language adaption is not as big a problem as it seems. HTTP requests which are automatically being redirected to a server located in close vicinity of the client are a far more intriguing threat. We have examined the URLs for the [EN] scenario and could not find any indication that this form of redirection was employed by any web site. Of course, some sites utilise *round robin* DNS entries in order to distribute the load on several webservers (e. g. google.com). But such procedures shouldn't affect the performance evaluation because their influence is averaged by the large amount of test cases.

Performance limit introduced by the Internet connection If the local area network suffers from performance fluctuations, it may influence the observed data as well. Network-caused performance breaks in all systems could be mis-interpreted as a common attribute. For example, if the network is not faster than the slowest anonymisation service, all systems would look the same. There is no influence if the local area netwok offers better performance than the fastest system at all times.

The basic idea to estimate the possible influence of the network (DIRECT) is to analyse all single test cases of all tested systems for this possible influence. We call the ratio of the number of all cases with a non-negligible influence to the total number of cases *critical influence ratio*. If this ratio is, for a scenario, higher than 5%, we call the influence of the network on the scenario *non-negligible*. Otherwise, we assume that there is no influence of the network on this scenario.

To calculate a *level of non-negligibility*, we suggest to evaluate all test cases by their throughput, separately for each scenario, by the formula presented in figure 2. This approach basically calculates the difference between the throughput measured for the network at a given hour and the throughput of a given test case in this hour. As a measure for the standard deviation of the network's bandwidth, we also provide the interquartile range for its througput. We subtract half of its value, as only the diminishment of the network's bandwidth is critical, and call the resulting value *critical throughput* for this test case. If the *critical throughput* is greater than zero, we assume a low possibility for network interference. Otherwise, the network influence is assumed to be non-negligible for this test case. As shown in table 8 (critical throughput influence ratio), we found a non-negligible network influence for 5 out of 12 scenarios. This

means that care must be taken when these scenarios are analysed, as at least some clipping phenomena[9] are expected.

Fig. 2. Evaluating performance influences of the network connection

$$
\begin{aligned}
I(S_t) &= \begin{cases} 0 & if\ Th_{\text{crit}}(S_t) \geq 0, \text{ small or no influence} \\ 1 & if\ Th_{\text{crit}}(S_t) < 0, \text{ possible high influence} \end{cases} \\
&\text{w.r.t} \\
Th_{\text{crit}}(S_t) &= Th(\text{DIRECT}_t) - Th(S_t) - \frac{IQR_B(\text{DIRECT})}{2} \\
&\text{where} \\
S &\in \{\text{DIRECT, TOR, DD, CCC}\} \\
t &:= \text{time (day and hour)} \\
S_t &:= \text{test case of } S \text{ at the time } t \\
IQR_B(\text{DIRECT}) &:= \text{Interquartile range of throughput of DIRECT} \\
Th(S_t) &:= \text{measured throughput of } S_t \\
Th_{\text{crit}}(S_t) &:= \text{critical throughput of } S_t \\
I(S_t) &:= \text{possible influence of DIRECT on } S_t
\end{aligned}
$$

Varying performance throughout the day An anonymity service saturated with a big and distributed user group is expected to show a normal distribution in user numbers, bandwidth and latency for each hour and day. In reality, though, the user groups may be heterogenous and therefore have a strong influence on performance over time. Before statistically analysing and comparing services, it is therefore useful to exploratively identify time-dependend trends in the user behaviour.

During the performance evaluation we retrieved the real-time number of concurrent users provided by the AN.ON services for further analysis. We identified two major trends:

1. The user numbers seem to follow a sinusoidal curve with vertex at 11 a. m. (cf. figure 5). Given that most users of AN.ON are located in Europe [8], this means that the majority of them is using the service during the day and not during the night.
2. The variables *throughput* and *delay* seem to be normally distributed between 1 p. m. and 9 p. m. Therefore, the influence of varying loads on the AN.ON services is expected to be minimal in that time period.

[9] Clipping means that some performance curves will have a hard break in the peaks.

Therefore, we decided to introduce a new scenario parameter *daytime* to simplify the comparison of AN.ON with Tor, which is more equally distributed over the whole day. *daytime* has the values *morning* (M) and *afternoon* (A), defined as the hour-of-day intervals 1-9 a. m. and 1-9 p. m. (Berlin local time). All test data from the remaining time periods was discarded.

4 Statistical methodology for analysis and comparison

For distinguishing differences in our sample from "random noise", we performed thorough statistical analyses. This section provides a short explanation of the statistical background needed to understand the results presented in section 5.

4.1 t-tests

In order to compare two samples, we use *Student's t-test*, which is very robust against violations of the *normality assumption*. In this paper we will use t-tests to compare the mean value of a given parameter (i. e., throughput or delay) of two samples (i. e., two anonymisation services). The t-test checks whether the means of the tested parameters *differ significantly* (hypothesis H_1).

t-tests can only be applied under the following assumptions [16]:

1. normal distribution of data
2. homogeneity of variances
3. independent, randomly selected samples

The last assumption is already addressed by the data quality measures mentioned in section 3.3. As we cannot expect the data in our samples to be normally distributed, we employ the *Kolmogorov-Smirnov test*. If the result of this test is significant, the data of the sample is not normally distributed and the t-test may draw incorrect conclusions. Similarly, the equality of the variances is proven with the *Levene test*. Even if the Levene test shows significantly differing standard deviations, the t-test can still be applied. In this case a modified version of the t-test has to be applied, though.

In the following sections the results of the the t-tests are shown in the column labelled "Sides". The higher the number of asterisks (*, **, ***), the more significant is the evaluated difference of mean values. A dash (-) indicates that the test found no significant difference (e. g. table 3).

4.2 Regression analysis

We analyse possible correlations of two or more metric parameters by a *Linear Regression Analysis*. It tests the assumption of a linear correlation between the dependend parameter y_i and the independent parameters x_i of the form

$$\hat{y_i} = b_0 + \sum_{j=1}^{m} b_j x_{ij}$$

for all test cases $i = 1, 2, \ldots, n$ and the independent parameters $j = 1, 2, \ldots, m$. In the following sections the confidence in the regression analysis is shown in the row labelled "Terms". The higher the number of asterisks (*, **, ***), the more significant is the estimated influence of the parameter (cf. table 6).

In order to be able to perform a regression analysis, the basic assumptions of *linearity*, *independence*, *homoscedasticity* and *normality* must be fulfilled for the data [7].

5 Evaluation

As mentioned earlier we decided to split the gathered data points into two data sets according to the time of day. The graphs in figures 5 and 6 show that user numbers, delay, and throughput follow a typical course for the two AN.ON services: between 1 p. m. and 9 p. m. the curves are approximately at the same level, whereas they resemble a quadratic function with a minimum at about 5 a. m. between 1 a. m. and 9 a. m. For the comparison of the services, we focus on the first of these periods which we call 'afternoon', as the AN.ON cascades are obviously not under full load during the latter one – most users are asleep during the 'morning' hours (cf. figure 6). Combining both the morning (M) and afternoon (A) data of the AN.ON services and comparing that with the results of Tor would unduly favor the AN.ON services, as Tor seems to be much less dependent on daytime.

Anyway, splitting the samples offers another benefit: As described in section 4.1 t-tests operate under the assumption of normally distributed data.[10] We found that *within each of the two periods* the samples are either normally distributed or closely resemble a normal distribution. This is not the case if the samples include data of the whole day, though.

Note that we will only provide results on latencies for the WEB scenarios as they are irrelevant for downloads.

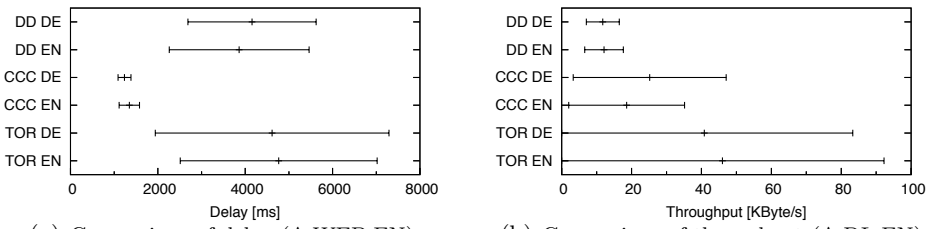
[10] Following common practices we use logarithmically transformed values for this purpose.

5.1 Descriptive statistics for DD, CCC and Tor

Descriptive statistics can provide some first hints regarding the characteristics of a sample. Our results show that the evaluated systems *differ* in offered bandwidth and latency. We suspect that the differences are partly due to varying *loads* (amount of concurrent users) on the anonymisation services. In the rush hours of the afternoon period, DD has very high user numbers (about 1,700 concurrent users on average). In contrast, CCC, which had to be selected manually in order to use it, is used by only 650 users on average. Figure 3 shows the mean values of delay (a) and throughput (b) together with the observed standard deviations for the individual services.

In terms of average delays, CCC offers best performance. The mean values for DD and Tor are considerably worse, but they are too close together for a meaningful graphical comparison. We will provide more concrete results utilising t-tests in section 5.3 and 5.4.

On the other hand, Tor might outperform the AN.ON services in terms of bandwidth. Due to the comparably high standard deviations a comparison without thorough analysis is difficult, though. The AN.ON services tend to offer a more constant QoS. From the user perspective, this may be an advantage, as users might not be interested in performance peaks, but rather in adequate performance every time they use the service.



(a) Comparison of delay (A,WEB,EN) (b) Comparison of throughout (A,DL,EN)

Fig. 3. Comparison of latency in the afternoon

5.2 Tor over daytime

While performance differences between the morning and afternoon periods are rather obvious for the AN.ON services (cf. figure 6), this is not that clearly visible for Tor. As Tor has a global network of nodes and a distributed user base, this is very reasonable. Looking at the descriptive

statistics, though, we found that the mean values of delay and throughput differed a lot between the morning and the afternoon period.

The results of the t-test suggest that there is indeed a difference between the two time periods (cf. table 3 and statistical remarks). Local time may therefore have a significant influence on local measurements, and Tor might not only prefer nodes with the highest bandwidth as found in a recent study [4], but also the nearest (low-latency) nodes. This may be due to an implicit attribute of its implementation, although there is no sign of such a strategy in the source code. If so, Tor's practical anonymity would be affected: The difficulty of mounting a collusion attack to capture the connections of specific local user groups would be substantially reduced. Another reason for the observed pattern might be that the initial assumption of a distributed user community is false. This is difficult to prove, though, as the Tor network does not provide information about the location and the number of its users. Accordingly, further research is needed to explain our observations.

Table 3. Tor: Performance differences morning/afternoon

Scenario			Means (exp)		Kol.-Smir.			T-Test	Sides	
Sim	Lang	Measure	M	A	M	A	Levene	T (df)	2	1
WEB	DE	Log(Delay)	3472	4097	-	-	3.2(177)	-2.3(177)	*	*
WEB	EN	Log(Delay)	3790	4231	-	*	1.0(178)	-1.8(178)	-	*
WEB	DE	Log(Thr)	8.7	6.3	*	-	6.9(177)*	2.4(170)	**	**
WEB	EN	Log(Thr)	5.9	4.9	-	-	0.6(178)	2.1(178)	*	*
DL	DE	Log(Thr)	43.9	34.7	-	-	0.0(176)	1.9(176)	-	*
DL	EN	Log(Thr)	45.7	39.1	-	-	1.5(176)	1.6(176)	-	-

Significance codes: *** p<0.001, ** p<0.01, * p<0.05

Remarks on statistical evaluation According to the results (cf. rightmost columns of table 3) we have to keep the null hypothesis for half of the scenarios. On the other hand, according to the 1-sided[11] t-test, all scenarios but {DL,EN} are significant. As the Kolmogorov-Smirnov test

[11] If there is a good reason – not concluded from the collected data – that one of the means should be higher or lower than the other one, the p-value (not shown in the tables) of the t-test may be halved, as only one side of the test is of interest, and the test returns a higher significance.

is only slightly significant in only two cases, there is a high confidence in the correctness of the test result.

5.3 Comparison of Tor and DD in the afternoon

The DD cascade is the common entry point to the AN.ON system for JAP users. As there is (at the time of measurement) no automatic switching function between different AN.ON cascades, most unexperienced users (who do not know how to switch cascades) use the DD cascade. In terms of latency the statistical results from table 4 show that there is little difference between DD and Tor in the afternoon period. This may indicate that there is a tolerance level for this kind of unexperienced users regarding latency of approximately 4 seconds. A constant latency above this level seems to deter from using the system.[12] This supplements the results of [11] who found that there is a linear relation between user numbers and latency by altering the internal delay of the DD service.

Remarks on statistical evaluation Looking at table 4 we observe that DD seems to have a slight advantage over Tor in regard to latency, but the difference is only significant for the {WEB,EN} scenario. But then, Tor obviously offers higher channel capacities by far (as shown by the {DL} scenarios) and thus is able to outrun DD in the {WEB} scenarios. The significant difference in bandwidth shows up in the {WEB,EN} scenario once again: Here, the difference in bandwidth is not as clear as in the {WEB,DE} scenario however.

Table 4. Comparison of Tor and DD on afternoon

Scenario			Means (exp)		Kol.-Smir.			T-Test	
Sim	Lang	Measure	Tor	DD	Tor	DD	Levene	T(df)	Sig
WEB	DE	Log(Delay)	4032	3689	-	**	31.7(178)***	-0.3(131)	-
WEB	EN	Log(Delay)	4238	3427	*	*	19.4(178)***	2.1(153)	*
WEB	DE	Log(Thr)	6.30	4.30	-	*	22.2(178)***	3.8(140)	***
WEB	EN	Log(Thr)	4.92	3.75	-	-	17.6(178)***	2.7(150)	**
DL	DE	Log(Thr)	34.71	10.31	-	*	46.1(176)***	7.7(119)	***
DL	EN	Log(Thr)	39.13	10.25	-	***	53.8(176)***	6.7(122)	***

Significance codes: *** p<0.001, ** p<0.01, * p<0.05.

Units: throughput [KBytes/s], delay [msecs]

[12] Note that using the system and being connected to it are two different perspectives.

5.4 Comparison of Tor and CCC on afternoon

While the DD cascade is the default in AN.ON's client software (JAP), the CCC cascade has to be explicitly selected by the user. Obviously, most users stay with the default (cf. figure 5). Consequently, this situation leads to lower latencies on CCC than on DD. Nevertheless, compared to Tor the bandwith of the CCC cascade is still lagging behind as shown in the {DL} scenarios in table 5. This is true even for the German downloads, where CCC presumably has an implicit advantage. Nevertheless, CCC outperforms Tor in the {WEB} scenarios, which is quite interesting. Apparently, for web surfing extremely low latencies (CCC) are more critical than sheer bandwith (Tor).

Table 5. Comparison of Tor and CCC on afternoon

Scenario			Means (exp)		Kol.-Smir.			T-Test	
Sim	Lang	Measure	Tor	CCC	Tor	CCC	Levene	T (df)	Sig
WEB	DE	Log(Delay)	4032	1091	-	-	98.4(178)***	17.5(96)	***
WEB	EN	Log(Delay)	4238	1191	*	-	91.2(178)***	18.9(105)	***
WEB	DE	Log(Thr)	6.30	10.07	-	-	25.0(178)***	-9.1(137)	***
WEB	EN	Log(Thr)	4.92	9.15	-	-	23.3(178)***	-11.4(143)	***
DL	DE	Log(Thr)	34.71	21.40	-	-	8.4(177)**	2.4(161)	*
DL	EN	Log(Thr)	39.13	15.84	-	**	25.0(176)***	4.3(142)	***

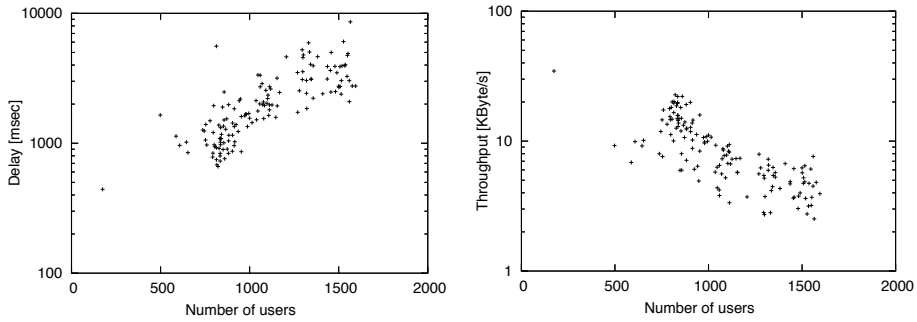
Significance codes: *** p<0.001, ** p<0.01, * p<0.05.

Units: throughput [KBytes/s], delay [msecs]

5.5 Correlations of user numbers and performance

In this section we will evaluate the influence of *load* on performance. AN.ON cascades provide the number of concurrent users at a given time. We will use this information to investigate the correlation between *user number* of both AN.ON cascades and the performance parameters. We expect a strong positive correlation between user numbers and latency and a strong negative correlation between user numbers and throughput. Figure 4 shows this graphically in two scatter plots.

The performance parameters have been scaled logarithmically as we expect an exponential influence of the load. The correlation is especially explicit in the *selected* morning period which contains data points with



(a) Scatter plot for delay (b) Scatter plot for throughput

Fig. 4. Influence of number of users on performance (M,WEB,DE,DD)

widely varying user numbers, whereas the afternoon period consists of fairly uniform data that is not suitable for further analysis.

The results of a regression analysis confirm the graphical observations. While both cascades are similar in terms of delay, their characteristics differ a lot in terms of throughput. Apparently, user numbers have a much greater effect on the performance of CCC than on DD. This observation cannot be explained by a gerenally inferior infrastructure (i. e., less capacity) of CCC, which still has plenty of unused resources (cf. figure 3). Instead, we assume that users on DD are considerably less active than those on CCC. A constant and inactive user base would correspond to the findings in [11] where still some hundred users were counted on DD even when the service had been made unusably slow.

According to these findings raw user numbers are no suitable predictor for load and expected performance on a cascade. We therefore suggest that AN.ON services should only report the number of *active users*. Otherwise, users might be deceived in terms of the provided anonymity, which is shown in JAP's *anonymeter*. As adjusted user numbers would correspond to the actual load they could serve as suitable performance measure. Due to their different characteristics finding a uniform regression model for multiple cascades can be a daunting task, though.

Remarks on statistical evaluation As we assume exponential correlations, all performance parameters are transformed by log_{10}. For the DE scenarios, we could clearly identify normally distributed (transformed) residuals, while this is not the case for the EN scenarios, though. As shown in table 6 the exponential correlation is highly significant and explains most of the spread of the performance parameters ($R^2 > 0.5$).

Table 6. Regression model for performance and user numbers for language DE

	Scenario					
	WEB				DL	
	DD	CCC	DD	CCC	DD	CCC
Param. ($\hat{y}_i$)	Log(Delay)	Log(Delay)	Log(Thr)	Log(Thr)	Log(Thr)	Log(Thr)
Terms						
Const. (B_0)	2.708***	2.655***	1.49***	1.66***	2.11***	2.871***
	(0.04)	(0.02)	(0.04)	(0.03)	(0.05)	(0.05)
Users	5.075***	5.802***	-5.215***	-10.235***	-6.781***	-25.170***
Model						
N	258	256	258	256	256	256
R^2	0.612	0.522	0.623	0.626	0.631	0.774
F	404.2***	277.1***	422.8***	426.0***	433.4***	870.8***
df	1/256	1/254	1/256	1/255	1/255	1/255

Significance codes: *** p<0.001, ** p<0.01, * p<0.05
Standard errors in brackets (). Users: $B_1 * 10^4$

6 Future work

Maybe our methodology for collecting performance data can be further improved concerning the robustness of the collected data. As measurements took place always in the same interval, this might give rise to inherent biases due to repeated network phenomena being in time with the test cycles. A simple solution might involve randomly changing session time slots or delays (cf. section 3.3), e. g. using a Poisson distribution as proposed in [13].

Moreover, extending the measured time frame would allow for interesting long term analyses and could help the developer community to understand the impact of newly introduced features. Besides, more AN.ON cascades with high load should be investigated in order to confirm the findings about a user tolerance level, and for building a common regression model for the cascade performance depending on user numbers. This will be more promising in the future, as AN.ON now has a client-based load balancing, and may take this study as a reason for only counting active users.

Finally, the time-dependent performance differences of Tor should be further analysed.

7 Conclusion

Evaluating the performance of Tor and two AN.ON cascades, we have shown that Tor, a large scale implementation of a free-route mixing protocol, is subject to unpredictable performance, while AN.ON, implementing typically more central mix cascades, is able to offer more consistent performance in general.

The suggestions of the Tor community regarding tuning the connection handling policy of the web browser to mitigate Tor's rather high network latencies [19] are a reasonable approach. Anyway, the overall performance of Tor is already sufficient for fast web surfing and downloads. The reason for the performance differences between morning and afternoon periods remains unclear for now. If Tor's routing strategy was really lured into selecting close-by nodes, this would have considerable implications for the anonymity provided.

In contrast, AN.ON's advantage in latency is restrained by its limited bandwidth and its lack of a load balancing mechanism. Apparently, the DD cascade of AN.ON suffers from high loads (up to 2,000 concurrent users observed). Therefore it cannot deliver satisfying performance during the busy afternoon period where it behaves comparable to Tor regarding latency. The less frequently used CCC cascade is able to offer low-latency web surfing, but at the price of a smaller user base and therefore less anonymity.

An important finding is the supposed user tolerance level for latency: Tor, as a distributed network with many entry points, may automatically adapt to user expectations regarding latency, and therefore pick up as many users as possible with the given network structure. Its performance is not expected to suffer noticeably from sinlge new users connecting to the system. AN.ON, on the other hand, deters a lot of users by offering a single entry point for new users right at the tolerance level, as the performance of this entry point is much more affected by new users than that of Tor.

As this relatively high latency seems to be tolerated by most privacy-aware users, i. e., the ones using Tor or AN.ON, this level may serve as a foundation for a new definition of *low-latency* in the context of anonymity services. Accordingly, this observation might be useful for designing new and more secure anonymity protocols. Further experiments should verify this level and whether it changes over time.

Acknowledgement

We thank Rainer Boehme for his priceless help concerning our statistical analysis, the reviewers for their valuable hints and remarks, and Simson Garfinkel for helping us as shepherd to give the article the final cut.

References

1. ActiveState ActivePerl. http://www.activestate.com/Products/ActivePerl/ (2006)
2. Alexa Top Sites. http://www.alexa.com/site/ds/top_sites (2006-02-06)
3. AN.ON: Protection of Privacy on the Internet. http://www.anon-online.de (2006)
4. Bauer, K. et al.: Low-Resource Routing Attacks Against Anonymous Systems. Technical Report. http://www.cs.colorado.edu/department/publications/reports/docs/CU-CS-1025-07.pdf (2007)
5. Boehme, R. et al.: On the PET Workshop Panel "Mix Cascades vs. Peer-to-Peer: Is One Concept Superior?" In: Lecture Notes in Computer Science, Proceedings for PET 2004 **3424** (2005) 243–255
6. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. In: Communications of the ACM **4(2)** (1981)
7. Draper, Norman R. et al.: Applied Regression Analysis. New York: Wiley (1966) 17.
8. Federrath, Hannes: Privacy Enhanced Technologies: Methods - Markets - Misuse. In: Lecture Notes in Computer Science, Proceedings of 2nd International Conference on Trust, Privacy, and Security in Digital Business (TrustBus '05), **3592** (2005) 1–9.
9. I2P. http://www.i2p.net (2006)
10. JMeter. http://jakarta.apache.org/jmeter/ (2006)
11. Köpsell, Stefan: Low Latency Anonymous Communication - How long are users willing to wait? In: Lecture Notes in Computer Science, Proceedings of Emerging Trends in Information and Communication Security (ETRICS '06), **3995** (2006) 221–237.
12. LWP::ParallelUA 2.57. http://search.cpan.org/~marclang/ParallelUserAgent-2.57/ (2006)
13. Paxson, Vern: End-to-end routing behavior in the internet. Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (1996) 25–38.
14. RFC2616 Hypertext Transfer Protocol - HTTP/1.1. Section 14.9 (2006)
15. Servertest: http://softwaregarden.com/products/servertest/index.html (2006)
16. Sheskin, David J.: Handbook of parametric and nonparametric statistical procedures, 2nd edition. Boca Raton: Chapman & Hall/CRC (2000) 247.
17. Tor: An anonymous Internet communication system. http://tor.eff.org (2006)
18. Tor FAQ: Why does Google show up in foreign languages? http://wiki.noreply.org/noreply/TheOnionRouter/TorFAQ#GoogleLanguage (2006)
19. Tor Wiki http://wiki.noreply.org/noreply/TheOnionRouter/FireFoxTorPerf (2006)

Appendix

Table 7. Domains chosen from Alexa's[2] top 20 and Downloads.de/.com top 200

Simulation	Language	Domains
WEB	DE	google.de spiegel.de amazon.de t-online.de msn.de mobile.de leo.org freenet.de arcor.de heise.de
WEB	EN	yahoo.com msn.com google.com passport.net amazon.com myspace.com microsoft.com bbc.co.uk aol.com blogger.com go.com alibaba.com cnn.com craigslist.org
DL	DE	virenschutz.info gratisgames24.de neuesvon.de
DL	EN	morpheus.com freewarefiles.com macromedia.com

The criteria for our choices of URLs were

- server performance much better than performance offered by anonymisation service, so that the results are not biased by slow servers[13]
- comparable number of URLs and downloaded bytes within the same scenario
- low number of HTTP errors produced by the requested web servers
- average total download time for web sites plus downloads of one language is much smaller than 30 minutes
- for web site URLs: plausibility of ranking in the Alexa top list

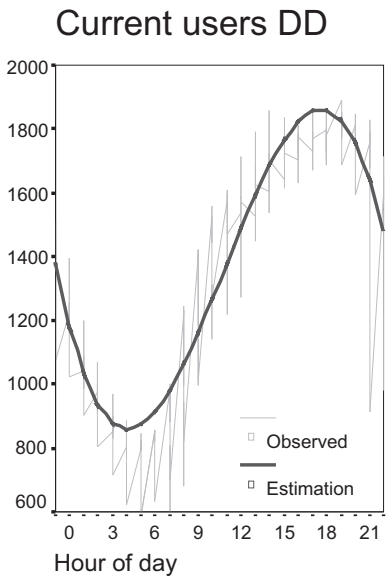
Geolocation detection As stated in section 3.3, the separation of the EN/DE scenarios might be jeopardised through geolocation of the client based on its IP address. Geolocation is performed by the webserver in order to (1) provide a localised version of a web site, or to (2) enhance the user-view performance by redirecting the request to a "nearer" webserver.

Localised versions of web sites do not influence our tests unduly, because latency and bandwidth are not affected. However, if requests are re-routed to another server, this will change. We applied the following checks to check whether any form of request re-routing took place:

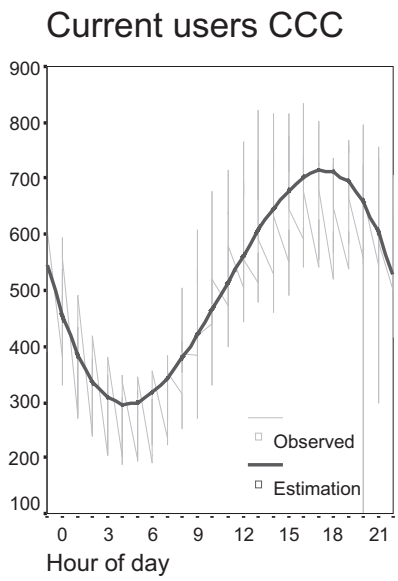
[12] To minimize space requirements, the domains are listed here only, not the downloaded files or the protocol identifier. Files were requested by HTTP only.

[13] As we can never be sure that all servers have an adequate speed during the measurement, we aggregate the download performance of a set of URLs to a *test case* in order to mitigate possible influences.

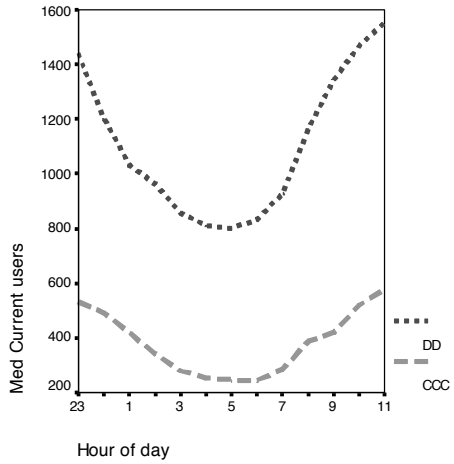
– We utilised the Unix *dig* utility and examined the DNS records for the individual hosts. We found multiple IPs and short TTLs, which indicates that several sites employed *round robin* IP rotation. Typically, web sites under high load use this approach for load balancing, but not for geolocation.
– We requested the individual URLs from our {EN} scenarios with the Unix *wget* utility and looked for HTTP redirects, which the webserver might send during geolocation: No URL used in the scenarios employed HTTP redirects for their homepage.



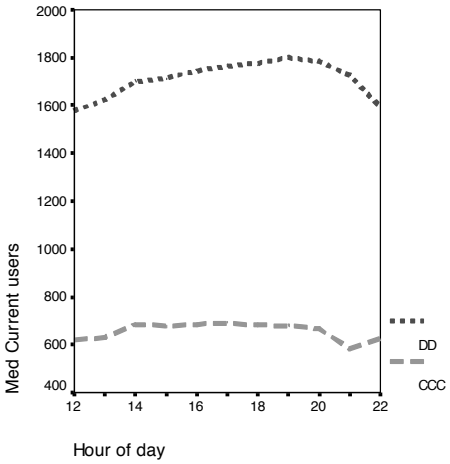
(a) Curve estimation: Avg. users over one day {DD}



(b) Curve estimation: Avg. users over one day {CCC}

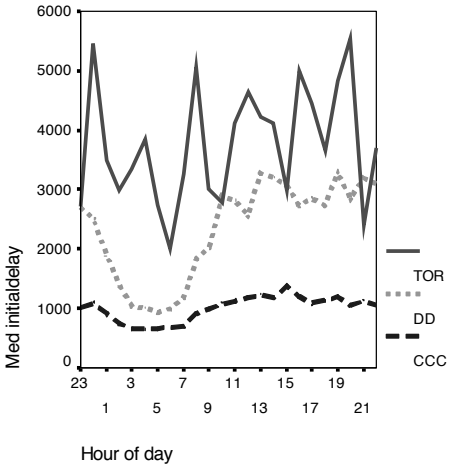


(c) Avg. users from 11 p. m. to 11 a. m.

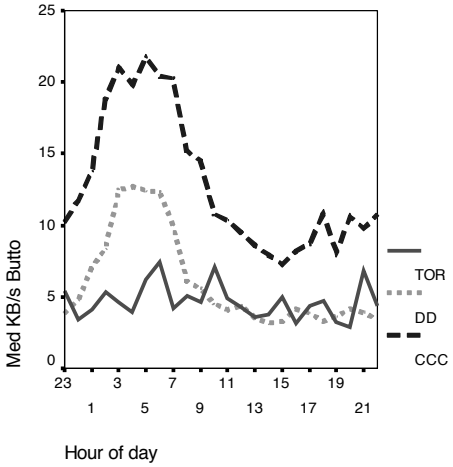


(d) Avg. users from 12 p. m. to 10 p. m.

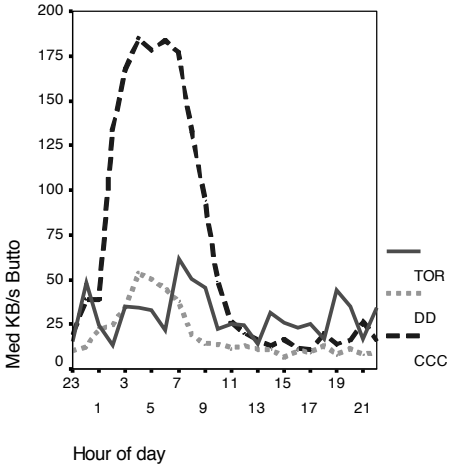
Fig. 5. User behaviour in AN.ON cascades



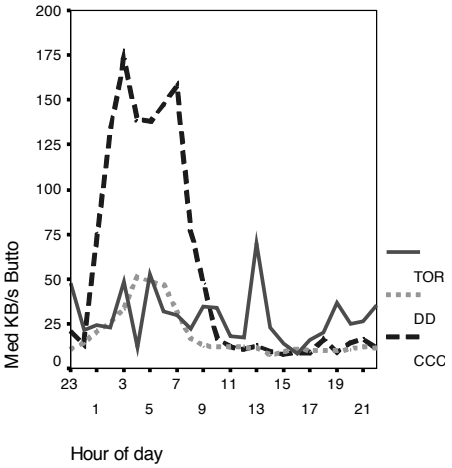
(a) Avg. delay over one day {WEB,EN}



(b) Avg. throughput over one day {WEB,EN}



(c) Avg. throughput over one day {DL,DE}



(d) Avg. throughput over one day {DL,EN}

Fig. 6. Graphical comparison of different anonymity services

Table 8. Data quality measures (cf. section 3.3)

Simulation		Web browsing		Downloads	
Language		DE	EN	DE	EN
Total test cases		258	258	258	258
30min overlap ratio		0.02	0.05	0.02	0.05
1h breaks		0	0	0	0
Critical throughput influence ratio	TOR	0.04	0.02	0.00	**0.18**
	DD	0.03	0.04	0.00	0.05
	CCC	**0.13**	**0.28**	0.01	**0.22**
Missing test cases	DIRECT	0	1	1	0
	TOR	1	0	2	2
	DD	0	1	2	0
	CCC	0	0	1	0
	ALL	1	2	6	2
Median received KBytes	ALL	1274.65	997.1	1529.44	1759.69
IQR received KBytes	ALL	73.64	56.50	82.00	0.00
HTTP Requests w/o failures		103130	79623	771	774
Error ratio		0.00	0.00	0.00	0.00
Failures	DIRECT	0	0	0	0
Median received KBytes ratio		0.00	0.01	0.00	0.00
IQR received KBytes ratio		**0.06**	0.05	0.00	0.00
HTTP requests w/o failures		102199	79264	768	768
Error ratio		0.00	0.00	0.02	0.02
Failures	TOR	0	0	0	**1**
Median received KBytes ratio		0.00	-0.02	0.00	0.00
IQR received KBytes ratio		-0.01	0.02	0.00	0.00
HTTP requests w/o failures		102236	79200	767	772
Error ratio		0.00	0.00	0.00	0.00
Failures	DD	**17**	**11**	**15**	**27**
Median received KBytes ratio		0.00	0.00	0.00	0.00
IQR received KBytes ratio		**0.06**	**0.17**	0.00	0.00
HTTP requests w/o failures		102845	79876	771	774
Error ratio		0.00	0.00	0.00	**0.09**
Failures	CCC	**3**	0	0	0
Median received KBytes ratio		0.00	0.00	0.00	0.00
IQR received KBytes ratio		-0.01	-0.13	0.00	∞