Performance Measurements and Statistics of Tor Hidden Services

Karsten Loesing, Werner Sandmann, Christian Wilms, and Guido Wirtz
University of Bamberg
Faculty Information Systems and Applied Computer Science
Feldkirchenstr. 21, D-96045 Bamberg, Germany
{karsten.loesing, werner.sandmann, guido.wirtz}@uni-bamberg.de

Abstract

Tor (The Onion Routing) provides a secure mechanism for offering TCP-based services while concealing the hidden server's IP address. In general the acceptance of services strongly relies on its QoS properties. For potential Tor users, provided the anonymity is secured, probably the most important QoS parameter is the time until they finally get response by such a hidden service. Internally, overall response times are constituted by several steps invisible for the user. We provide comprehensive measurements of all relevant latencies and a detailed statistical analysis with special focus on the overall response times. Thereby, we gain valuable insights that enable us to give certain statistical assertions and to suggest improvements in the hidden service protocol and its implementation.

1. Introduction

Tor hidden services [2] constitute a convenient way for providing a TCP-based service to clients without revealing the hidden server's IP address. Typical applications are hidden web servers or hidden IRC servers. There are often good reasons for people who provide potentially controversial services or content to others to hide their identity. Otherwise these people could be faced with personal consequences, ranging from job-related disadvantages up to prosecution and personal harm.

Tor's anonymity originates from relaying traffic over a network of about 2,200 publicly deployed relays (April 2008). A client that wants to communicate anonymously creates a *circuit* consisting of three randomly selected Tor relays. The last relay in a circuit connects to the public server that the client actually wants to talk to. All messages between client and the last relay are encrypted in multiple layers, which is the reason for Tor's name: *The Onion Routing*. The idea is that no single entity but the client can learn where the circuit starts and where it ends.

The hidden service design is based on connecting two circuits—one of them created by the client, the other by the hidden server—on a commonly agreed Tor relay. This so-called *rendezvous point* acts as message relay by forwarding outgoing messages from the client-side circuit to the server-side circuit and vice versa. In order to protect rendezvous points from attacks, a hidden service picks a set of Tor relays as *introduction points* which work similar to the rendezvous points. Introduction points are only used for transferring a single message containing the location of the selected rendezvous point. In order to accept client requests, the hidden service publishes a *hidden service descriptor* containing a signed list of introduction points to directory servers from which it can be downloaded by clients.

The design of hidden services is inevitably more complex than the design for anonymizing a connection between a client and a public service. Constructing a circuit requires three Tor relays, whereas accessing a hidden service involves more than a dozen. This makes accessing a hidden service outstandingly slow and connections fragile, which may be considered the major problems of hidden services.

We study the performance and QoS properties of hidden services in the public Tor network via measurements and statistical analysis. We focus on latencies rather than on bandwidth, because previous studies have shown that latencies of connection establishment are the major problem in terms of usability [8].

In the next section we briefly review previous work on the performance of Tor and hidden services. Section 3 describes our setting to measure client access times to a publicly deployed hidden service. Overall response times and times of a number of sub-steps are considered. In Section 4 we statistically analyze the measurements and fit them to probability distributions in order to detect and investigate bottlenecks or irregularities and to be able to predict response within a given time with a certain probability. In Section 5 we propose some performance improvements based on our observations. Finally, Section 6 concludes the paper and outlines further research directions.

2. Related Work

Wendolsky et al. [12] measured the performance of client-anonymous connections in Tor. They found that latencies of connections averaged to 4 seconds. They concluded from the studies by Köpsell [7] that these 4 seconds were the acceptable time that users are willing to wait: Whenever the number of users increases, so that the network load goes up, the average latency in the network increases too, and the less anonymity-aware users are deterred from using the system, so that the user base and with it the average latency stabilizes. However, we want to emphasize that these 4 seconds cannot be directly compared with the latencies of hidden services, because of the inherent complexity and higher number of involved nodes.

In earlier measurements we found that connection establishment to a hidden service took in average 5.39 seconds [8]. Those numbers are significantly lower than those to be presented in the present paper, because we then excluded the times for descriptor download and data exchange. In [8] we also found that subsequent message exchanges only took in average 2.32 seconds, which is quite fast compared to connection establishment. Therefore, we did not consider message exchange times in this paper.

Øverlier and Syverson [9] proposed a new connection establishment protocol for hidden services. Their revised protocol reduces the number of involved Tor relays compared to the original design [2] and therefore should lead to reduced latencies. An implementation might lead to significant performance improvements for hidden services.

3. Environment and Measurement Setup

Assume, Bob wants to offer a hidden service. The first step is to configure his Tor client accordingly. The Tor client generates a long-term public key pair to identify the service. Further the Tor client selects three nodes within the Tor network to act as the server's introduction points and opens a circuit to each of them. We configured the Tor client to select a specific node, that was also controlled by us, as first introduction point. The information to access the service, i.e. the identifier and the addresses of the three introduction points forming the hidden service descriptor, is published on a lookup directory server within the network. The process of setting up a hidden service only needs to be performed once and was consequently done prior to the actual measurements. It has no influence on the service user perspective. All steps necessary to establish and access the hidden service are shown in Figure 1 and described below.

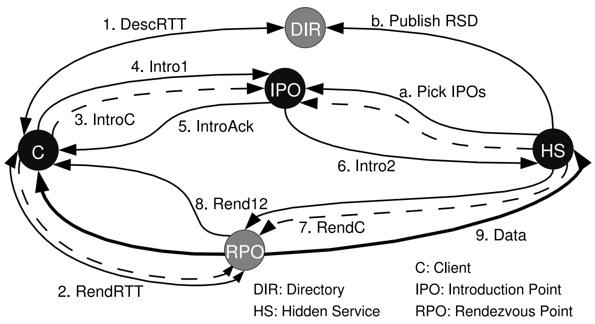

Figure 1. Establishing and accessing a hidden service

Alice wants to access Bob's website and therefore learns the identifier from a website or another source. First her Tor client contacts the directory server and receives the hidden service descriptor (the time between opening a circuit to send the request and receiving the reply is referred to as *DescRTT* in the next section). Regularly, it would randomly pick one of the introduction points, but we implemented a minor code change to force Alice's Tor client to always pick the first introduction point from the hidden service descriptor. As mentioned above this introduction point was controlled by us. After fetching the hidden service descriptor Alice's Tor client chooses another node in the Tor network as rendezvous point and builds a circuit to it. If available, the Tor client may pick the third Tor relay in an existing 3-hop circuit instead of building a new circuit to a random Tor relay, which is called *cannibalization* and which can be done without delay. Alice's Tor client requests the chosen rendezvous point to act as such and the latter acknowledges (*RendRTT*). At the same time the Tor client builds a circuit to the introduction point (*IntroC*). If a suitable prebuilt 3-hop circuit is available, this circuit is extended to the rendezvous point with a fourth hop. This is another form of cannibalization, which is faster than building all hops on demand. When the introduction circuit is built and the rendezvous point has acknowledged the request, Alice's Tor client informs the introduction point that Alice wants to access the hidden service (*Intro1*), handing over the rendezvous point's address. The introduction point forwards the message to the hidden server (*Intro2*) and sends an acknowledgment back (*IntroAck*). Now the hidden server also builds a circuit to the rendezvous point (*RendC*), that connects it to the circuit built by Alice's Tor client and informs her client of the successful connection (*Rend12*). This circuit can also be built by cannibalization, so only the fourth hop to the rendezvous point needs to be built on demand. Now a connection between Alice and Bob is established and Alice can send a request, e.g. an HTTP GET message to retrieve a website (*Data*). Figure 2 shows the sequence of all values measured to emphasize parallel steps and to display a *critical path* when trying to reduce the complete response time of hidden services.

For the measurements we used two instances of Tor version 0.2.0.6 alpha (r11276) running on a virtual root server

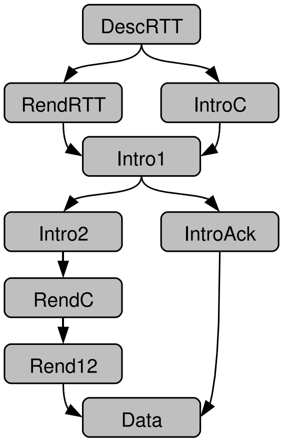

Figure 2. Sequence of measured values

located in Frankfurt on the Main, Germany. We observed log events indicating the sending and receiving of messages as well as opening new circuits to other nodes. In order to gather the data we started the Tor relay acting as introduction point first and then the Tor client providing the hidden service prior to the actual tests. We created new Tor clients for the test every five minutes for 72 hours between 27–29 August 2007, and let them perform a single access attempt on our hidden service. We chose not to re-use the same onion proxy on client side to avoid the effects of caching information. Further we did not control all roles involved in the process. We used the official directory servers of the Tor network and therefore had no access to their log events. We did also not control the relay chosen as rendezvous point by Alice, because the configuration option to choose a specific rendezvous point does not work in case of cannibalization.

4. Statistical Data Analysis

The first step in statistical data analysis consists of computing meaningful empirical characteristics of the measured data to get an impression of the basic properties. Overviews of the empirical statistics and percentiles for all measured times are given in Table 1 and Table 2, respectively.

In order to find appropriate probability distributions that well represent the measured data, we performed a parametric fit. Some candidate distributions were selected and for each of these distributions a maximum likelihood estimation (MLE) was carried out to find those parameters that fit best to the data. The goodness of fit was tested by the χ^2 test, the Kolmogorov-Smirnov (KS) test, and the Anderson-Darling (AD) test. In a nutshell, given a parametric distribution and measurement data, MLE determines the parameter values that are most likely with respect to the data. Goodness-of-fit tests check the assumption that the data is according to a specific distribution and compute test statistics that indicate how well justified this assumption actually is. Roughly speaking, the goodness of fit is evaluated by the rule that the smaller the test statistic, the better the fit. For the details of the statistical methods see, e.g., [11, 13].

Due to space limitations it is not possible to present plots and advanced investigations for all measurements. We restrict our detailed presentation to response times. These are most important from the users' point of view as they are what users really care about and perceive. Hence, they are particularly well suited as a measure of user-perceived quality of service (QoS) or quality of experience (QoE).

We selected candidate distributions based on the empirical statistics and percentiles as well as on the shape of histogram plots. We also considered some widespread distributions often appearing in network models. For instance, the most common one is the exponential distribution and thus we included it. However, from the data it is obvious that any distribution whose support is the whole set of positive numbers will probably not well fit the data for small values since the minimum response time is significantly larger than zero. Consequently, distributions like the shifted exponential distribution should be considered. Another reasoning takes the occurrence of some very large data values into account which may indicate heavy-tailedness. In fact, there is much evidence in the presence of heavy-tailed distributions on the Internet [1, 10], where the Pareto distribution has become particularly prominent. However, even in cases where the Pareto distribution seems most appropriate, from a practical point of view it may be more convenient to work with other distributions, e.g., when the goal is to build stochastic models involving the distribution and further investigate these models. For instance, the Pareto distribution causes serious problems in simulation [4]. Besides, Downey [3] found that the lognormal distribution is often more appropriate. As a distribution with a similar shape to that of the lognormal distribution where in particular the mode is not at the lower boundary, the loglogistic distribution is also taken into account. Particularly reasonable with respect to the measured data are extreme value distributions where also the mode is not at the lower boundary. We considered the Frechet distribution also known as extreme value distribution of type 2, and the generalized extreme value distribution. Detailed descriptions of all the mentioned (and many more) distributions can be found in [5, 6]. Table 3 contains the densities of the candidate distributions. The parameters obtained by MLE are given in Table 4 and the goodness-of-fit test results in Table 5.

From the goodness-of-fit tests we can obtain a clear ranking for the fitted distributions. For all tests, sorting with respect to the test statistics yields the same order among the distributions. We can see that the exponential distribution, either shifted or not, and the Pareto distribution seem to be completely inappropriate. The Frechet distribution unam-

Table 1. Empirical statistics of measured times, times are given in seconds

Statistics	DescRTT	RendRTT	IntroC	Intro1	IntroAck	Intro2	RendC	Rend12	Data	RespTime
Min	0.073	0.020	0.063	0.010	0.008	0.136	0.050	0.032	0.690	2.118
Max	82.022	55.006	106.560	32.427	7.648	27.472	48.386	56.531	99.653	151.850
Mean	5.333	1.842	5.657	0.718	0.597	0.512	2.336	1.586	6.530	24.052
Var	94.767	24.819	235.320	3.393	0.731	2.381	17.933	10.986	72.766	555.340
StdDev	9.735	4.982	14.340	1.842	0.855	1.543	4.235	3.315	8.530	23.566
CoeffVar	1.825	2.705	2.712	2.565	1.431	3.015	1.813	2.090	1.306	0.980
StdErr	0.338	0.173	0.532	0.067	0.031	0.056	0.147	0.115	0.296	0.817
Skewness	4.800	6.433	3.617	11.739	4.267	12.379	5.243	8.979	5.279	2.243
Kurtosis	26.980	51.479	12.072	168.410	26.081	182.880	35.728	112.030	38.485	5.233

Table 2. Empirical percentiles of measured times, times are given in seconds

Perc.	DescRTT	RendRTT	IntroC	Intro1	IntroAck	Intro2	RendC	Rend12	Data	RespTime
5%	0.401	0.072	0.177	0.035	0.033	0.142	0.299	0.219	1.442	5.752
10%	0.510	0.132	0.250	0.054	0.051	0.143	0.377	0.270	1.723	7.324
25%	0.963	0.250	0.493	0.144	0.142	0.143	0.556	0.425	2.463	10.039
50%	2.541	0.491	1.169	0.279	0.258	0.145	1.157	0.992	4.305	15.238
75%	5.495	1.473	2.362	0.894	0.923	0.489	2.141	1.534	7.022	26.624
90%	11.861	2.895	6.123	1.346	1.302	1.005	4.253	2.546	11.324	65.157
95%	17.024	7.958	60.504	1.991	1.806	1.550	8.748	3.945	20.125	77.504

Table 3. Densities of candidate distributions

Distribution	Density $f(x)$
Exponential	$\lambda e^{-\lambda x}$
Shifted Exp.	$\lambda e^{-\lambda(x-c)}$
Extreme	$\frac{\exp\left(-(1+\lambda z)^{-1/\lambda}\right)}{\sigma(1+\lambda z)^{1/\lambda+1}}$
Frechet	$\frac{a}{b}\left(\frac{b}{x}\right)^{a+1} e^{-(b/x)^a}$
Loglogistic	$\frac{a}{b}\left(\frac{x-c}{b}\right)^{a-1}\left(1+\left(\frac{x-c}{b}\right)^a\right)^{-2}$
Lognormal	$\frac{1}{x\sigma\sqrt{2\pi}}\exp\left(\frac{-(\ln x-\mu)^2}{2\sigma^2}\right)$
Pareto	$\frac{ab^a}{x^{a+1}}$

Table 4. MLE parameters

Distribution	MLE parameters
Exponential	$\lambda = 0.042$
Shifted Exp.	$\lambda = 0.046, c = 2.118$
Extreme	$\lambda = 0.414, \mu = 12.92, \sigma = 8.829$
Frechet	$a = 1.624, b = 12.091$
Loglogistic	$a = 1.98, b = 13.994, c = 2.023$
Lognormal	$\mu = 2.849, \sigma = 0.771$
Pareto	$a = 0.477, b = 2.118$

biguously provides the best fit. However, one needs to be careful with such a quick argumentation since the statistical tests only provide an overall view of both data and distributions without any special regard to particular data regions. We shall get more specific insights from visualizations of the data and the fitted distributions.

4.1. Visualization and Interpretation

Figure 3 shows a plot of the densities together with the histogram for the measurements which gives a rough overall impression of the appropriateness of the fitted distributions. It confirms the conclusion from the goodness-of-fit tests for the overall view. Figure 4 and Figure 5 show probability-probability (PP)-plots and probability difference plots, respectively, for the fitted distributions against the measured data. A PP-plot is a graph of the distribution function of the fitted distributions versus the empirical distribution of the measured data where a perfect fit would yield a straight

Table 5. Goodness-of-Fit test statistics

Distribution	χ^2	KS	AD
Exponential	332.1	0.169	39.096
Shifted Exp.	214.8	0.114	23.837
Extreme	45.8	0.083	4.894
Frechet	15.4	0.037	2.147
Loglogistic	36.1	0.051	4.065
Lognormal	76.6	0.088	10.043
Pareto	874.1	0.348	159.280

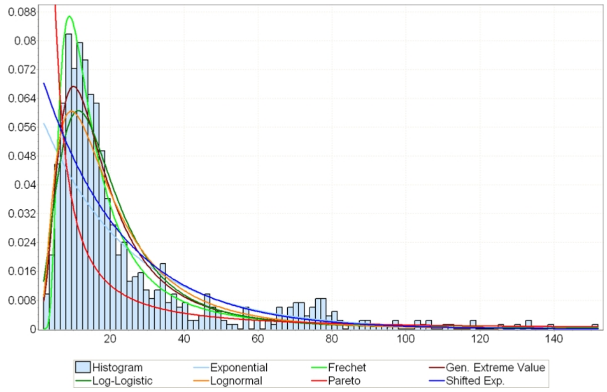

Figure 3. Histogram-density plot for all fitted distributions

diagonal line from (0,0) to (1,1). A probability difference plot, as the name suggests, visualizes the difference between the empirical distribution and the distribution function of the fitted distributions. It should be close to zero.

All plots clearly eliminate the Pareto distribution from any further considerations. It neither reflects the data for small or medium values nor for large values. Taking a deeper look at some peculiarities we can recognize that none of the distributions appropriately covers the irregularities for data values slightly greater than 60 seconds. In practice, there is a simple explanation for this effect. If the connection request is not handled by the server within 60 seconds, then a timeout occurs and the connection is requested again. We performed fitting procedures where the time scale was subdivided into intervals of length 60 seconds, resulting in the same ranking of distributions as for the overall fit. Thus, for the time being with regard to the statistical analysis of response times, we can neglect this effect because neither one of the candidate distributions is ranked better or worse due to this effect and, more important from the applications point of view, end users care about their effective response time regardless whether or not it includes timeouts. Moreover, the following observation will lead us to a separate treatment of response times less than 60 seconds and response times greater than 60 seconds anyway.

An important effect indicated by all plots is the significant difference in the goodness of fit with respect to the distributions' body on the one hand and the tail behavior on the other hand. In fact, the preference for the Frechet distribution due to the statistical tests mainly relies on its excellent ability to reflect the range up to 60 seconds whereas the tails decrease exponentially. This particularly demonstrates that there is no evidence of a heavy-tailed distribution of the response times, but rather the tails almost perfectly follow the exponential distribution. In order to illustrate this, we show comparative plots of the exponential distributions (shifted and non-shifted) versus the Frechet distribution in Figures 6–8.

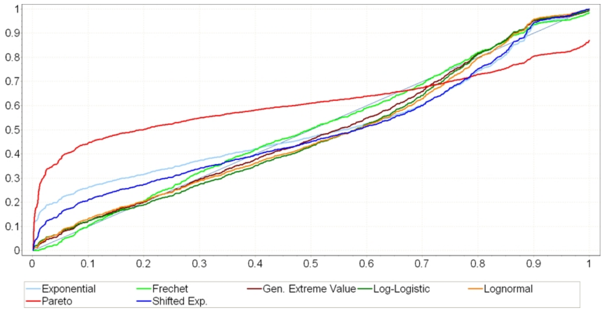

Figure 4. PP-plot for all fitted distributions

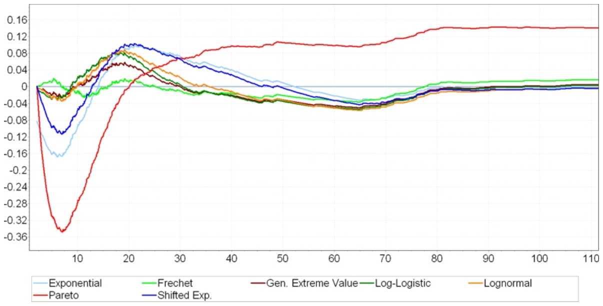

Figure 5. Probability difference plot for all fitted distributions

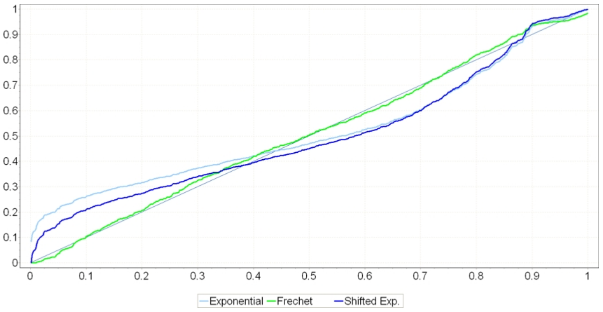

Figure 6. PP-plot for Exp and Frechet

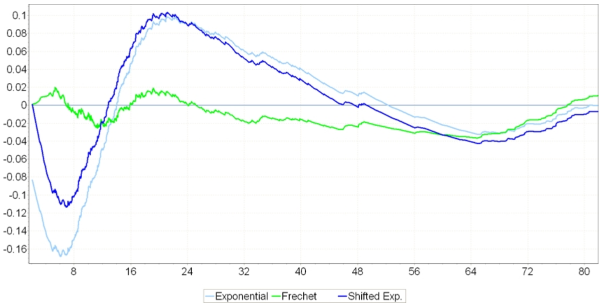

Figure 7. Zoomed probability difference plot Exp and Frechet distribution in range [0,80]

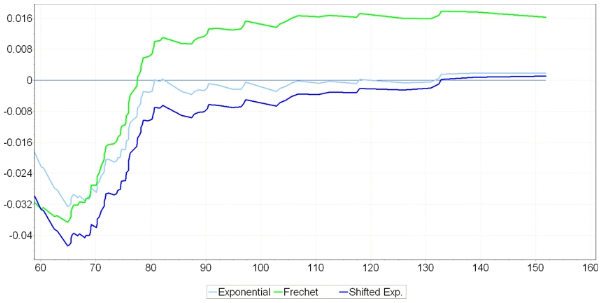

Figure 8. Zoomed probability difference plot Exp and Frechet distribution in range [60,160]

The PP-plot of Figure 6 elucidates the good overall fit of the Frechet distribution. Figure 7 and Figure 8 provide zoomed probability difference plots where we have split the time scale. The range from 60 to 80 seconds is included in both plots in order to show that neither distribution fits well there, but all distributions come back to a quite good fit for values greater than 80 seconds. Most importantly, Figure 7 manifests the superiority of the Frechet distribution for response times less than 60 seconds, whereas Figure 8 shows that the exponential distributions perfectly coincide with the tail of the empirical distribution provided by the measurements.

Another important effect can be observed when plotting the distribution functions as shown in Figure 9. The cumulative distribution functions of the exponential and the Frechet distribution intersect at time 60 seconds. This is particularly remarkable since it means that the probability mass assigned to values less than 60 seconds is equal for both distributions. The same holds for the probability mass assigned to values greater than 60 seconds. In particular, the probability for a response time of more than 60 seconds, one possibly critical QoS parameter, is identical for both distribution types. Evaluating the distribution function (in the following of the exponential distribution) we get $P(R > 60) = 1 - F_{Exp}(60) = 1 - e^{-0.042 \times 60} \approx 8\%$.

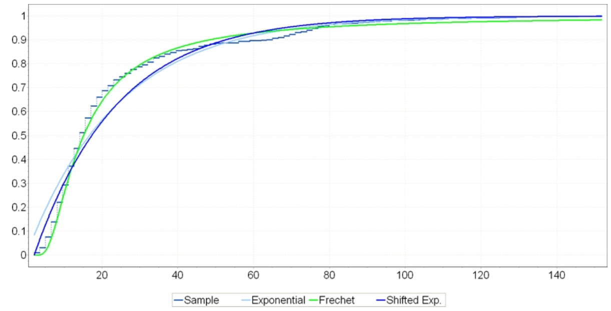

Figure 9. Cumulative distribution functions of Exp and Frechet distribution

4.2. Implications

Our findings directly imply that we can improve the fit of the response time distribution by combining the Frechet and the exponential distribution such that for times less than 60 seconds the Frechet distribution applies and for times greater than 60 seconds the exponential distribution applies. In particular, note that due to the intersection of both distribution functions at time 60, this does not need any further normalization but directly yields that the combined density integrates to one as required to be a density. Hence, altogether we obtain a very accurate fit of the response time distribution that can in turn be used for diverse further modeling and analysis purposes providing insights useful with regard to improvements from a statistical perspective. One immediate result is that, as a generalization of the above evaluation of $P(R > 60)$, the exponential distribution applies to large response times. Hence, far better QoS guarantees follow than it would be in case of applying the Frechet distribution in the tails.

5. Recommended Improvements

After analyzing the current performance of hidden services in Tor we suggest a number of modifications in implementation and protocol to accelerate the access to hidden services and therefore improve usability. The challenge is at the same time not to hurt or reduce the anonymity of either the requesting client or the hidden service provider and not to increase the network load disproportionally.

Comparing the mean values in Table 1, building the introduction (IntroC) and rendezvous (RendC) circuits takes most time, besides fetching the hidden service descriptor (DescRTT) and data transfer (Data). Therefore special focus should be put on circuit creation.

In the current implementation three internal circuits are built before actually needed in order to be cannibalized later. That is the exact number, that is necessary for accessing a hidden service: one circuit to retrieve the rendezvous descriptor, one for the rendezvous circuit and one to contact the introduction point. If one of the circuits is not ready on demand, it cannot be cannibalized and a complete new circuit has to be built. If more general circuits are built in the beginning, the probability of finding a working circuit is increased.

The Tor client could start to cannibalize two 3-hop circuits to contact the introduction point simultaneously. The first circuit that is opened successfully is used to send the introduction message, while the other circuit is discarded. This behavior uses the high variability of circuit cannibalization to decrease the time until an open circuit is available.

The hidden service proxy picks three introduction points and publishes them. In the current protocol version the client proxy randomly chooses one of them and tries to build a circuit to it. We suggest that the client proxy tries to connect to two introduction points simultaneously and when the first circuit is established, discards the other connection attempt. As a disadvantage this would result in increased network traffic. Therefore we need to consider the tradeoff between resulting performance improvements and additional network traffic.

The impact of reducing the number of Tor relays involved in the process of accessing a hidden service, as suggested by Øverlier and Syverson in [9], needs to be evaluated. In one of the proposed scenarios the introduction point also acts as rendezvous point. In another scenario the rendezvous circuit is cannibalized and extended to the introduction point.

Finally, the client-side timeout after which a connection is requested again should be reduced from 60 seconds to a lower value. Though this might occasionally result in unnecessary network traffic, the probability of high response times decreases.

6. Conclusions

We performed comprehensive performance measurements of Tor hidden services and statistically analyzed the data focussing on the response time as an important user-oriented QoS parameter. Our studies resulted in a mathematically well-founded fit of the response time distribution by means of combining Frechet and exponential distribution. Moreover, we obtained insights that led to recommendations on how to improve the hidden service protocol and its implementation. Further research includes the development of a detailed model based on the measurements of sub steps. Such a model could be a foundation for sophisticated mathematical analysis, e.g. via queueing theory or simulation. This would allow us to immediately observe the effects of changes to the protocol which yields huge time savings compared to the effort required for repeated measurements. Besides, the recommended improvements can be efficiently evaluated in advance before actually implementing them.

References

[1] R. J. Adler, R. E. Feldman, and M. S. Taqqu, editors. *A Practical Guide to Heavy Tails*. Birkhäuser, 2000.

[2] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[3] A. B. Downey. Lognormal and Pareto distributions in the Internet. *Computer Communications*, 28:790–801, 2005.

[4] G. S. Fishman and I. Adan. How heavy-tailed distributions affect simulation-generated time averages. *ACM Trans. Modeling and Computer Simulation*, 16(2):1–22, 2006.

[5] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*. vol. 1, Wiley, 2nd edition, 1994.

[6] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*. vol. 2, Wiley, 2nd edition, 1995.

[7] S. Köpsell. Low latency anonymous communication – How long are users willing to wait? In *Emerging Trends in Information and Communication Security*, volume 3995, pages 221–237. Springer, June 2006.

[8] K. Loesing, M. Röglinger, C. Wilms, and G. Wirtz. Implementation of an Instant Messaging System with Focus on Protection of User Presence. In *Proceedings of the Second International Conference on Communication System Software and Middleware*. IEEE CS Press, January 2007.

[9] L. Øverlier and P. Syverson. Improving efficiency and simplicity of Tor circuit establishment and hidden services. In *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies*, pages 134–152, Ottawa, Canada, June 2007. Springer.

[10] K. Park and W. Willinger, editors. *Self-Similar Network Traffic and Performance Evaluation*. Wiley, 2000.

[11] S. M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2004.

[12] R. Wendolsky, D. Herrmann, and H. Federrath. Performance comparison of low-latency anonymisation services from a user perspective. In *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies*, pages 233–253, Ottawa, Canada, June 2007. Springer.

[13] S. S. Wilks. *Mathematical Statistics*. Wiley, 1962.