Performance Measurements of Tor Hidden Services in Low-Bandwidth Access Networks

Jörg Lenhard[1], Karsten Loesing[2], and Guido Wirtz[1]

[1] University of Bamberg, Germany
joerg.lenhard@stud.uni-bamberg.de,
guido.wirtz@uni-bamberg.de
[2] The Tor Project
karsten.loesing@gmx.net

Abstract. Being able to access and provide Internet services anonymously is an important mechanism to ensure freedom of speech in vast parts of the world. Offering location-hidden services on the Internet requires complex redirection protocols to obscure the locations and identities of communication partners. The anonymity system Tor supports such a protocol for providing and accessing TCP-based services anonymously. The complexity of the hidden service protocol results in significantly higher response times which is, however, a crucial barrier to user acceptance. This communication overhead becomes even more evident when using limited access networks like cellular phone networks. We provide comprehensive measurements and statistical analysis of the bootstrapping of client processes and different sub-steps of the Tor hidden service protocol under the influence of limited access networks. Thereby, we are able to identify bottlenecks for low-bandwidth access networks and to suggest improvements regarding these networks.

1 Introduction

With the Internet paving its way into more and more areas of life and business, also the need for privacy on the Internet is ever increasing. The greater the number of users and the wider the area of utilization gets, the greater grows also the number of loopholes that can be exploited through the lack of privacy. But privacy is, among other things, the basis for various core values of democratic societies, like freedom of speech. Hence, providing mechanisms for anonymous communication can be considered an important goal. Privacy is not only relevant for those requesting information or using services offered by others in an anonymous manner. It is also important for providers of services. There is no merit in being able to utter one's opinion freely and without fear of harassment if there is no platform on which one could do so.

There are various approaches to address the subject of anonymous communication. One of them is based on the concept of routing traffic through networks of relays, called *anonymity networks*. The Tor network [6] is a widely deployed anonymity network and consists of approximately 1,300 relays in March 2009. A user of such a network builds a chain of several relays, a *circuit*, to prevent others

M. Abdalla et al. (Eds.): ACNS 2009, LNCS 5536, pp. 324–341, 2009.

from linking her identity or location with her actions. All connections between relays are secured using cryptographic mechanisms and none of the relays knows both initiator and responder of a communication session. The user then routes her traffic over the circuit. So, for an outside observer, it looks as if all requests are performed by the last relay in the circuit and not by the actual user. The assumption is that an adversary does not control all relays in a circuit, or more precisely, at least not the first and last relay in a circuit.

Tor permits requesting information from public servers anonymously, as well as providing services pseudonymously, without revealing the IP address of the server. The former functionality is given by attaching application-level streams to circuits which are built as described above. The latter functionality is called *hidden services* and works by connecting circuits of both client and hidden server on a common rendezvous point, again a relay in the network, to grant location privacy to both communication parties. It is obvious that this process is inevitably more complex than connecting to a non-anonymized service.

The usual assumption nowadays is that clients or service providers use broadband access networks of some kind, like cable, DSL, or UMTS. But access networks with lower bandwidth, like second-generation cellular wireless or fixed-line networks are still in wide-spread use. These networks generally provide lower data rates and higher latencies. In many regions of the world, especially less industrialized countries, users are dependent on older, and therefore inferior networks. However, these regions might have an even higher demand for privacy than well-connected areas, as it happens to be the case that they are also less politically stable. The question to be answered here is to what extent the access network of a user influences her capability to use an anonymity network. Studies on the influence of low-bandwidth access networks on the usage of anonymity networks are rare. The present study is the first one to consider the access of location-hidden services using such access networks.

The approach we are taking here is to measure the performance of Tor processes over low-bandwidth access networks, in particular mobile phone and fixed-line telephone networks. We created a measurement setup, involving several Tor processes using these networks, as well as broadband networks. We focus on the evaluation of clients bootstrapping in the low-bandwidth environments and the sub-steps of connecting to hidden services. Both accessing and providing hidden services over low-bandwidth access networks is considered. By these measurements, we identify specific bottlenecks in the process that need to be improved.

In the next section we give a brief overview over previous work on the performance of anonymity networks, especially Tor. Section 3 describes the Tor bootstrapping phase and the Tor hidden service protocol, being the focus of this paper. Section 4 contains an analysis of the proportion of low-bandwidth clients in the Tor network and a description of the environment we created to gather the data. In Section 5 we present statistical analysis of the data from the bootstrapping phase, discuss the implications of this data and suggest performance improvements based on our evaluations. Section 6 contains a similar analysis for hidden services with special focus on circuit building times. Section 7 concludes the paper.

2 Related Work

Work on the performance of anonymity networks is not only motivated by improving usability for its own sake. The level of anonymity provided by the network is dependent on the number of users, forming the underlying *anonymity set*. Networks that offer a high performance will attract more users, resulting in a higher degree of anonymity provided for all participants [5].

Recently, there is a growing interest in measuring performance of anonymous communication. Köpsell [10] observed the influence of the performance provided by a network on the number of users. Wendolsky et al. [18] measured the performance of anonymous communication from the client's point of view. They observed connection latencies to be on average approximately 4 seconds for the Tor network. Utilizing the work of Köpsell, they conclude that these 4 seconds are the overall tolerance level users are willing to take. It is important to note that these 4 seconds cannot be directly compared to this study. Here, we observe accessing and providing hidden services which is necessarily more complex than accessing public services anonymously.

Panchenko et al. [14] focus on the examination of possible reasons for the delay of the Tor network. Their special interest concerns the building of circuits and the geographical diversity of the relays in a circuit. With the help of empirical measurements, they advertise a new path selection algorithm to improve the performance of anonymous communication via Tor.

Øverlier and Syverson [12] suggested changes to the protocol for establishing connections to hidden services. Their suggestions include the reduction of the number of relays involved in the process, which should lead to a decrease in connection establishment times. In earlier studies [11], we measured the latencies during connection establishment to hidden services with special focus on the overall response times. We found that connection establishment, when using a broadband access network, took on average 24 seconds. It is important to mention that these numbers are lower than those presented in this paper. Here, we also consider the time a client needs to build a circuit to a directory server.

However, all studies discussed in this section only consider broadband access networks, neglecting the influence of low-bandwidth access networks as discussed here.

3 Tor Background

Before going into the details of the measurements and their results, some background on the measurement setup is in order. In this section, we describe the Tor bootstrapping process and the Tor hidden service protocol, being the focus of our measurements.

When connecting to the Tor network for the first time, a client needs to download and verify information about the status of the network and single relays in it. [16] describes document formats and [3] outlines the process in detail. As the documents reflect the state of the network, their size can vary

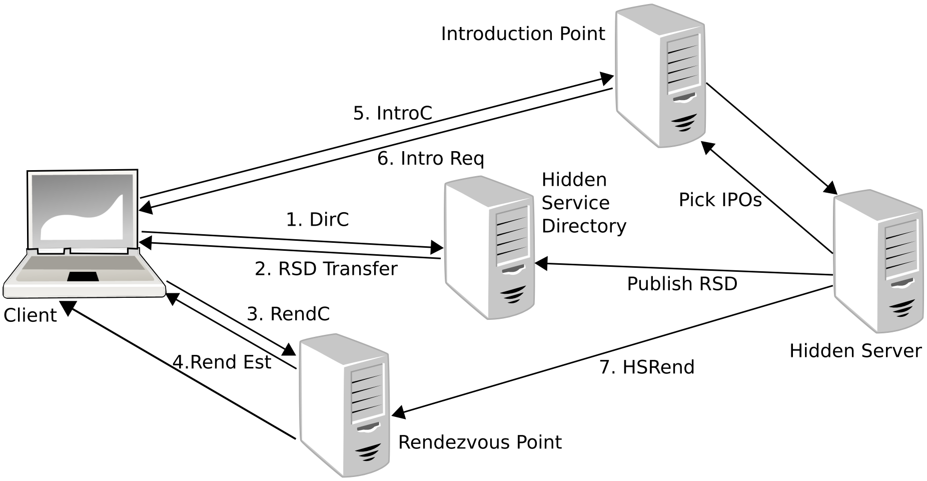

Fig. 1. Establishment and access of a hidden service

strongly, depending on the size of the network. The initial action of a newly started Tor process is to choose a directory authority, establish a TCP connection to it, perform a TLS handshake, and establish a one-hop circuit (bootstrapping phase 0–15%). Next, the Tor process opens a stream to load a network consensus document (15–25%). The process retrieves the document which currently (March 2009) has a size of approximately 90 kilobytes, checks its signatures, and starts loading relay descriptors (25–50%). The process continues loading descriptors until at least one fourth of the total amount is fetched (50–80%). All server descriptors of the network currently add up to approximately 1.6 megabytes of data. Then, the process chooses relays and starts building circuits. For this, again a TCP connection to a relay is built and a TLS handshake is performed. The process then keeps on adding relays to the circuit until it has finished the first circuit consisting of three hops, concluding the bootstrapping process (80–100%). So, all in all about 500 kilobytes of data need to be downloaded by a newly started process to successfully connect to the Tor network. The rest of the data will also have to be downloaded during runtime for ensuring anonymity.

Tor can be used for accessing public services in an anonymous way, but it can also be used to provide services anonymously. The actions described above are independent of hidden services and also apply to normal Tor usage. To be able to communicate with each other anonymously, the provider of the service as well as the client have to perform various steps of the hidden service protocol. Figure 1 visualizes the process of establishing and accessing a hidden service and outlines which steps are measured.

The first step in the hidden service protocol [17] is the establishment of a hidden service in the network by its provider, Bob. For this, Bob configures a Tor process to act as a proxy for his service. The Tor process then builds circuits to three arbitrarily chosen relays in the network and establishes *introduction points* on them for his service. Introduction points work as medium-time contact points for clients trying to access the hidden service. Furthermore, Bob generates a public and a private key for the service and derives a unique identifier from it, the *onion address*. This address consists of sixteen characters ending in *.onion*. As the onion address is derived from the public key, anyone possessing the key can verify that they are communicating with the respective service. In the next step, Bob constructs a *rendezvous service descriptor* (RSD) with the contact information of the introduction points and his public key. He signs the descriptor with his private key and publishes it to a directory server, normally an ordinary relay that provides additional functionality for storing RSDs. Now the hidden service is ready to be accessed by a client, Alice.

First, Alice learns about the hidden service and its onion address and decides to access that service. She needs a Tor process to work as a proxy for her request. She builds a circuit to a directory server (*DirC*) and asks for Bob's RSD. Not all circuits needed during the connection establishment are newly built. If possible, the process tries to pick an existing pre-built circuit. This technique is called *cannibalization* and means that the purpose of a previously built circuit can be changed to whatever purpose is required. This operation can be done without delay. The cannibalized circuit only needs to be extended by a single hop to the directory node. If the RSD is found, Alice downloads it (*RSD Transfer*). She then finds the introduction points' addresses along with Bob's public key in it.

As soon as the RSD is loaded, Alice tries to cannibalize two more circuits. The first one is the circuit to the *rendezvous point* (*RendC*) which is a randomly chosen relay in the network, Alice wants to use for later message exchange with Bob. This circuit has to be a three-hop circuit that can simply be cannibalized, without further operations needed. If no circuit is available for cannibalization, a new three-hop circuit is built from scratch. After completing the circuit to the rendezvous point, Alice establishes it as such (*Rend Est*). This establishment consists of the transmission of two cells. The first cell is sent from Alice to the rendezvous point and requests the rendezvous. The second cell is sent from the rendezvous point to Alice and acknowledges the request. When requesting the rendezvous, Alice also hands over a one-time secret, serving as her identification. The second circuit built after the reception of the RSD is a circuit to one of the introduction points of the hidden service (*IntroC*). As soon as a circuit to an introduction point is completed and a rendezvous point is established, Alice requests this relay to introduce herself to the service (*Intro Req*). She does this by handing over the rendezvous point's address and her one-time secret, encrypted with Bob's public key. The introduction point answers with an acknowledgment message and forwards Alice's request and her secret to the hidden service. Bob decrypts this message using his private key and obtains the address of the rendezvous point and Alice's one-time secret. Bob can now decide if he wants to contact Alice, and if so, he builds a circuit to the rendezvous point (*HSRend*). When this circuit is completed, Bob asks the rendezvous point to connect his circuit to Alice's. The rendezvous point matches their circuits with the help of the one-time secret and establishes a connection between the two. Then, the rendezvous point sends Alice a notification about the connection establishment.

Now, Alice and Bob can start exchanging messages via the rendezvous point. We denote the period from request start to reception of the connection establishment message, sent from the rendezvous point to Alice, as total round-trip time (*Total RTT*) and the period from reception of the RSD to reception of the same connection establishment message as small round-trip time (*Small RTT*).

4 Measurement Setup

The measurement setup consists of a few Tor processes connected to the public Tor network over either broadband or low-bandwidth access networks. In this section, we give some information about the low-bandwidth access networks and describe the utilized Tor versions and process distribution.

The access networks we observed are analog modulation via the telephone network, the mobile network *Enhanced Data Rates for GSM Evolution* (EDGE) [15], and a broadband network. The modem we used was of standard *V92*, thus offering a data rate of 56 kilobits per second downstream and 44 kilobits per second upstream [9]. EDGE provides a data rate of up to 230 kilobits per second, depending on radio conditions. The broadband connection was represented by the university network, consisting of fiber optics and offering a data rate of up to 100 megabits per second. For the Tor processes a minimal fraction of this rate would have been sufficient, so the broadband access network can be considered unlimited for the measurements.

These access networks can be seen as representatives of the prevailing types of access networks nowadays. Analog modulation formed the most important access network in the early times of the Internet. Although it is losing its importance more and more, it is still widely in use, especially in developing countries and rural areas where the establishment of broadband networks is not yet lucrative for ISPs. Moreover, many desktop and laptop computers are also equipped with V92 modems per default. EDGE is an enhancement of the GSM standard for mobile communication which was established in 1982. As of September 2008, GSM makes up eighty percent of the world's subscriber connections [7]. In contrast to broadband mobile access networks, such as UMTS, EDGE and its predecessor enhancement of GSM, GPRS [15], are widely distributed in industrialized countries and also available in less developed areas. Data transmission via optical fibers provides the highest possible data rates today. It is not yet forming a major access network, due to its expense. Instead, it is generally combined with other fixed line access networks. Fiber optics connects main centers, whereas the final connections to households are built using, for example, DSL.

When conducting the measurements, we assumed that low-bandwidth access networks are used by a major share of the networks' clients. Based on a suggestion from one of the reviewers, we investigated this assumption more closely. We observed the bandwidth of clients downloading the network consensus document from one of the six directory authorities for one week between March 14–21, 2009. We analyzed the size and duration of every consensus document download to conclude which bandwidth clients have. We excluded relays to observe only

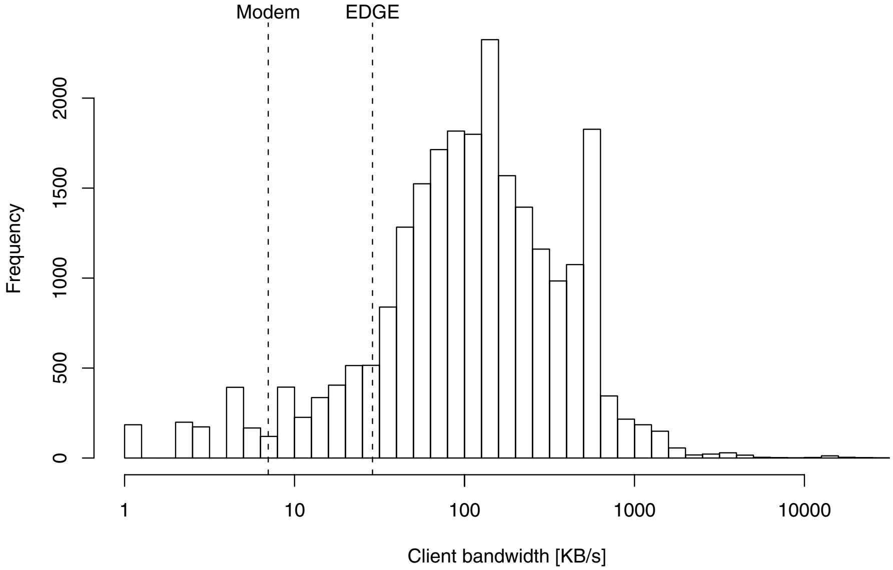

Fig. 2. Download speed of client connections loading the network consensus in log scale

the bandwidth of clients. Results are shown in Figure 2. Roughly 7% of these connections might have been performed by clients using a V92 modem, and 16% by clients using EDGE or a modem. So, a total of 16% of the network's clients can be considered low-bandwidth in the terms of this study. Our measurements only include successful downloads, so that the number of low-bandwidth clients might be even higher. This is a sufficiently large share to demand special interest. Furthermore, if the network had lower bandwidth requirements, the number of low-bandwidth clients might increase even more.

We observed the log events of the Tor processes indicating the sending and reception of messages, the opening of circuits and the progress in the bootstrapping phase. Client and hidden service processes used Tor version 0.2.1.5-alpha as code base. We had to patch this version to resolve two bugs that would otherwise have had an impact on the measurements.[1]. The first bug involved failures when loading router descriptors, and the second bug lead to erroneous behavior when loading rendezvous service descriptors. Both bugfixes are contained in Tor version 0.2.1.6-alpha which was not available at the time of performing measurements. So, we patched the Tor versions of clients and hidden services with all revisions that were necessary to fix these bugs.[2]

We further implemented a few changes to the Tor source code in order to perform measurements: The first change forces the Tor process offering a hidden

[1] Detailed descriptions can be found in Tor's bug tracker: `http://bugs.noreply.org/flyspray/index.php?do=details&id=767` and `http://bugs.noreply.org/flyspray/index.php?do=details&id=814`

[2] These were the revisions r16808, r16810, r16817, and r16915.

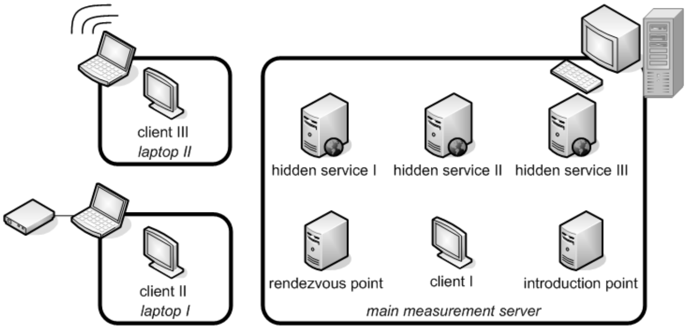

Fig. 3. Process distribution with clients using low-bandwidth access networks

service to select a specific relay as introduction point which can be controlled by us. The second change is to make clients pre-build a three-hop circuit to a specific relay which is also controlled by us, so that it can be chosen as rendezvous point later on. As a third source code change, the client selects the introduction point that is controlled by us, given that Bob had chosen it in the first place. For the measurements we set up a Tor relay, acting as introduction and rendezvous point. This Tor process was running an unpatched Tor 0.2.1.4-alpha version, as neither introduction nor rendezvous point were affected by the previously mentioned bugs. The measurements were then divided into two phases. During all measurements, three hidden services, one for each access network type, and the relay were running continuously.

The Tor processes for the hidden services as well as the introduction and rendezvous point were started some time prior to the measurements. Clients accessing the services were created in regular intervals. We did not use the same Tor processes for the clients, but created new ones in each interval, to avoid any influences on the results by caching directory information. The distribution of the processes on the different physical machines is shown in Figures 3 and 4, respectively. In the first measurement phase, the clients used low-bandwidth access networks, while the hidden services had broadband access. In the next measurements the configuration was turned around and the clients used the broadband access network, while the hidden services were offered over low-bandwidth access networks. Every access network was used by a different physical machine and the low-bandwidth access networks were only used by one Tor client or hidden service at a time to not overcharge them. So, all in all, we used three computers, the main measurement server and two laptops. All other processes were running on the machine using the broadband access network. However, this was not a problem, because all processes communicated over circuits in the real Tor network and never directly.

The interval in which client processes were created, and thus the time they had to bootstrap and perform the hidden service request, was capped at 6 minutes for the client-side low-bandwidth measurements and 5 minutes for the server-side low-bandwidth measurements. During both measurements, as soon as the client process finished bootstrapping, the hidden service request was initiated.

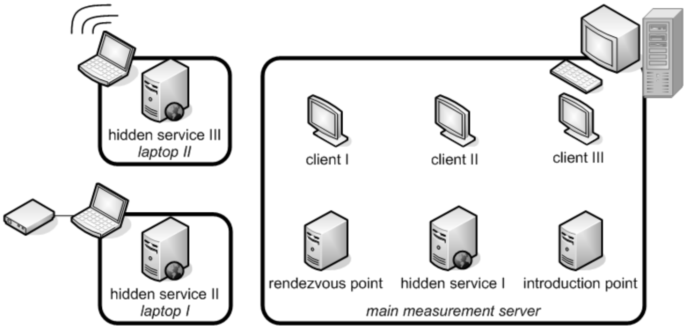

Fig. 4. Process distribution with hidden services using low-bandwidth access networks

We chose different intervals for both measurements due to the bootstrapping phase. During the server-side low-bandwidth measurements, hidden service processes were running over low-bandwidth networks. These processes needed to bootstrap only once in advance to the measurement period. During the other measurements, bootstrapping by the client processes needed to take place in every interval, also consuming more time. Client-side low-bandwidth measurements then lasted for 134 hours between 23–29 September 2008 and server-side low-bandwidth measurements for 114 hours between 6–11 October 2008.

5 Bootstrapping

As a first step in analysis, we investigate the total bootstrapping time as visualized in Figure 5. It becomes clear that bootstrapping over limited access networks is a major problem in comparison to the broadband access network. For the limited networks, the total bootstrapping time is approximately five times that of the broadband network, with median values of 232.9 seconds for EDGE and 249.0 seconds for modem and an interquartile range of 91.9 seconds for EDGE and 45.6 seconds for modem. The broadband median lies at 22.9 seconds and the interquartile range at 39.3 seconds. It is important to note that descriptive values are likely to be even higher in the population, especially for maximum values. In the measurements, test runs were stopped after 6 minutes which eliminated records that would have exceeded this value.

It has turned out that some sub-steps of the bootstrapping process contribute more to these differences than others, as can be seen in Figure 6. It is obvious that the most time-consuming sub-step lies between fifty and eighty percent, where relay descriptors are loaded. At the time of measurement, at least 325 relay descriptors had to be loaded during this phase to initiate the building of circuits. These descriptors make up the largest share of the data that needs to be downloaded during bootstrapping. The median duration of this sub-step ranges from 127.0 seconds for EDGE to 155.6 seconds for modem which is more than forty times as long as the broadband network with 3.3 seconds. This is a

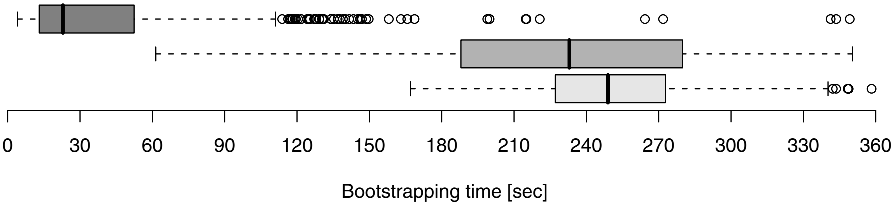

Fig. 5. Durations of total bootstrapping time [sec] in broadband (dark gray), EDGE (medium gray), and modem (light gray) access networks

considerable barrier for the usage of the anonymity network over limited access networks.

A reduction of the initial amount of descriptors that need to be downloaded is not an option, as it would pose a threat to anonymity which is of course more important than performance. The smaller the initial set of relays a client is able to choose from, the more the client is prone to *intersection attacks* [2]. For these attacks, it is necessary that the users of the network are not continuously active and some messages might be linkable. If an attacker knows the initial set of relays a client might use, she could observe messages sent via these relays at a given point of time and intersect the sets of possible active senders, thus cutting out the non-active users at this point of time and reducing the sets of possible senders. The smaller the initial set of relays, the easier this operation gets. By systematically reducing the sets of possible senders, an attacker could correlate messages to certain clients.

The problem of slow bootstrapping is also addressed by several Tor proposals. One approach is to drop the requirement to download server descriptors while bootstrapping [13] and download them on demand while building circuits. In this approach, clients would still be able to use all relays for circuit building. The idea is to add all information that is required for path selection into the network summary, so that server descriptors are only required for building circuits. This approach reduces the download size of directory information during bootstrapping from at least 500 kilobytes to 100 kilobytes. The disadvantage of the described approach is, however, that all circuit extensions require an additional message to download the required server descriptor. Clients must not cache received server descriptors for future extensions, because this would leak the information that a client has used a relay before from not having to ask for its descriptor. As a result, the improvement in bootstrapping leads to deterioration in circuit establishment. A subsequent proposal [4] introduces microdescriptors containing only the onion key as the minimum information for building circuits. Clients would download microdescriptors instead of router descriptors, reducing the total size of directory information during bootstrapping to around 300 kilobytes. It is yet uncertain which variant will be implemented in future Tor versions. But the discussion shows that there is a need to find better solutions to accelerate the bootstrapping process and support clients on low-bandwidth access networks.

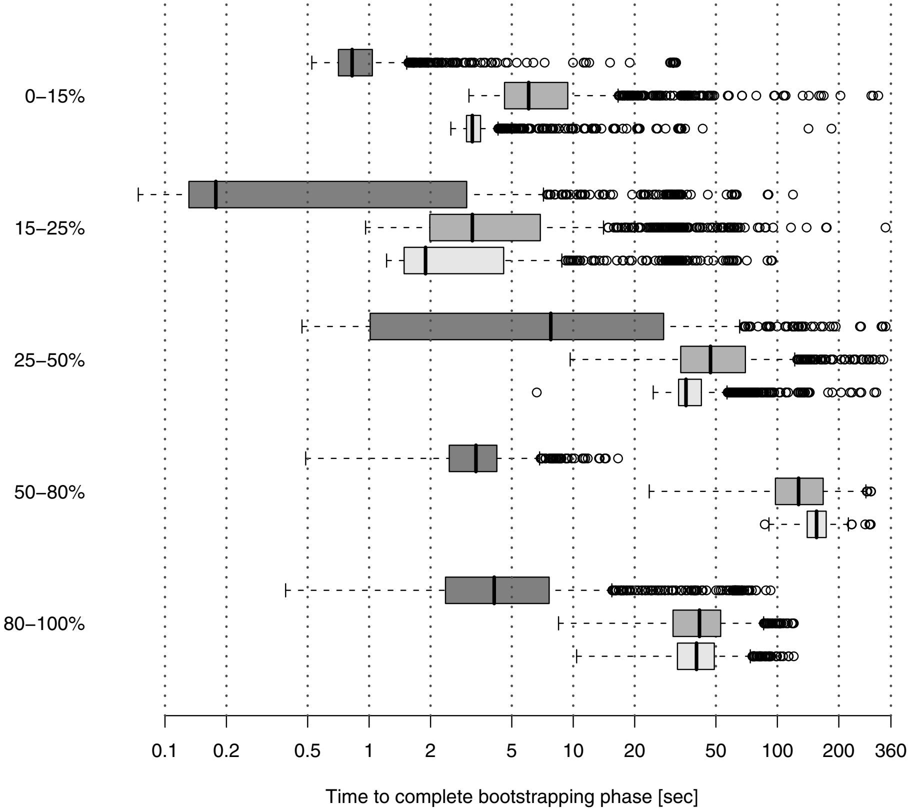

Fig. 6. Durations of bootstrapping substeps [sec] in broadband (dark gray), EDGE (medium gray), and modem (light gray) access networks on a logarithmic scale

6 Hidden Service Access

The second focus of this paper lies on hidden service access times. It has to be stated that bootstrapping took far longer than we had expected. This had an effect on the measurements of hidden service connection establishment. If the bootstrapping phase took up most of the whole measurement interval, there was no time to perform the actual hidden service request. This lack of time resulted in a cut-off and missing values at some point during the process. For the evaluation of the hidden service requests, we limited the data set to those requests that were not influenced by the bootstrapping phase. That is to say, only requests are considered that had at least 2 minutes of the measurement interval left. However, these restrictions only affect the data of the client-side low-bandwidth measurements, as only here bootstrapping was a major issue. Instead of 1,350 hidden service requests performed, we limited the data set of the low-bandwidth access networks to around 500 records. For these records, independence from the bootstrapping phase is guaranteed.

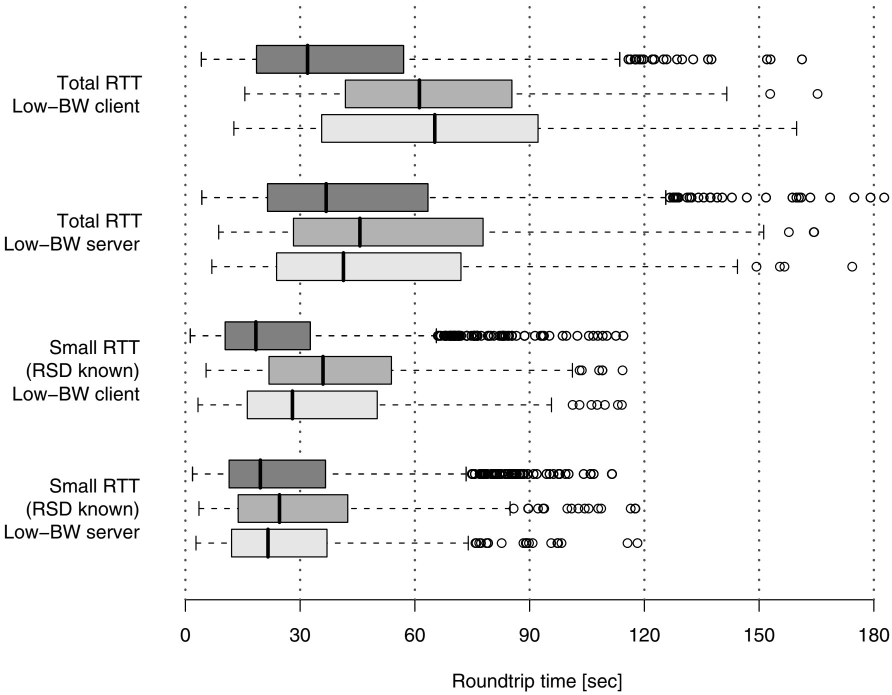

Fig. 7. Durations of round-trip times [sec] in broadband (dark gray), EDGE (medium gray), and modem (light gray) access networks

Figure 7 outlines round-trip times for clients and servers on low-bandwidth access networks. For the client-side measurements, the differences between the access networks are obvious. When considering the total RTT, median values range at up to more than twice as high for the limited access networks, with a total RTT of 61.2 seconds for EDGE and 65.2 seconds for modem in comparison to 32.0 seconds for broadband. The interquartile range lies at 43.5 seconds for EDGE, 56.6 seconds for modem and 38.4 seconds for broadband. For the small RTT, the differences between broadband and limited networks shrinks to 17.6 seconds for EDGE and 9.6 seconds for modem, when comparing the median. Absolute median values and interquartile range amount to 36.0 seconds and 31.7 seconds for EDGE, 28.0 seconds and 34 seconds for modem as well as 18.4 seconds and 22.2 seconds for broadband. For the server-side low-bandwidth measurements, the difference is less obvious. When looking at the total RTT median, it shrinks to 8 seconds between EDGE and broadband and only 1 second between modem and broadband, with absolute values of 44.2 seconds for EDGE, 37.7 seconds for modem and 36.4 seconds for broadband. The small RTT shows similar results with a difference of 8.6 and 2.5 seconds, respectively, when comparing the broadband network to EDGE, respectively modem. Absolute median values range at 25.8 seconds for EDGE, 19.7 seconds for modem and 17.2 seconds for broadband.

When looking at the round-trip times in the server-side low-bandwidth measurements, we can observe that values for the low-bandwidth access networks do not differ strongly from those of the broadband network. Especially the values of the modem network range at a level of only approximately 1 second higher. An analysis of the different sub-steps of the whole protocol unveils the reasons for this. For events where the hidden service access network is involved, broadband shows a better performance. But these events have a much smaller impact on the total access time than events dependent on the client access network. For the client-side events, we observed a slightly better performance of clients accessing services with a low-bandwidth access network, ranging at a level of 0.1 seconds per event. These discrepancies have to be assumed to be random, because all respective processes were running on the same physical machine using the same access network. There is no way in which client processes accessing low-bandwidth services could have been preferred over other processes. Still, these random differences equalize the differences produced by the hidden service access network. This becomes especially obvious for the modem connection where, in the end, there is hardly any difference to the broadband connection. We conclude that the influence of the hidden service access network on the hidden service protocol is of rather minor importance. Hidden services can in principle be offered over low-bandwidth access networks, without enlarging the overall connection establishment time too much. Other factors might be more likely to produce a bottleneck here. These could, for example, be the usage of the access network for something besides offering the hidden service, thus limiting the available bandwidth even further. Also the size of the actual product of the service or the number of clients accessing the service at the same time are relevant.

We concentrated our further analysis of hidden service access on circuit establishment. The building of the various circuits consumes the largest share of time in the whole process, in many cases up to 80% of the total access time. Once circuits are completed, message transfer times only constitute minor delays. Figure 8 shows establishment times for all circuits involved in the process of accessing a hidden service. For the completion of each of the circuits there is a timeout of 60 seconds. If the circuit is not completed within this time, it is abandoned and a new attempt is started. It is important to mention that the presented data constitutes absolute times until a circuit to a respective relay is established. This can involve more than one attempt and thus also more than 60 seconds.

The *client-side circuit to the rendezvous point (RendC)* is built very quickly for all access networks, almost immediately after requesting it. The median values are 0.0 seconds for all access network types. This is the case, because the rendezvous circuit is simply cannibalized and not extended. In very few cases, cannibalization was not possible, and a new circuit had to be built which of course took some more time.

The *client-side circuits to the directory server (DirC) and introduction point (IntroC)* show bigger differences between the access network types. Values for these circuits are very similar for the same network types, as they are built

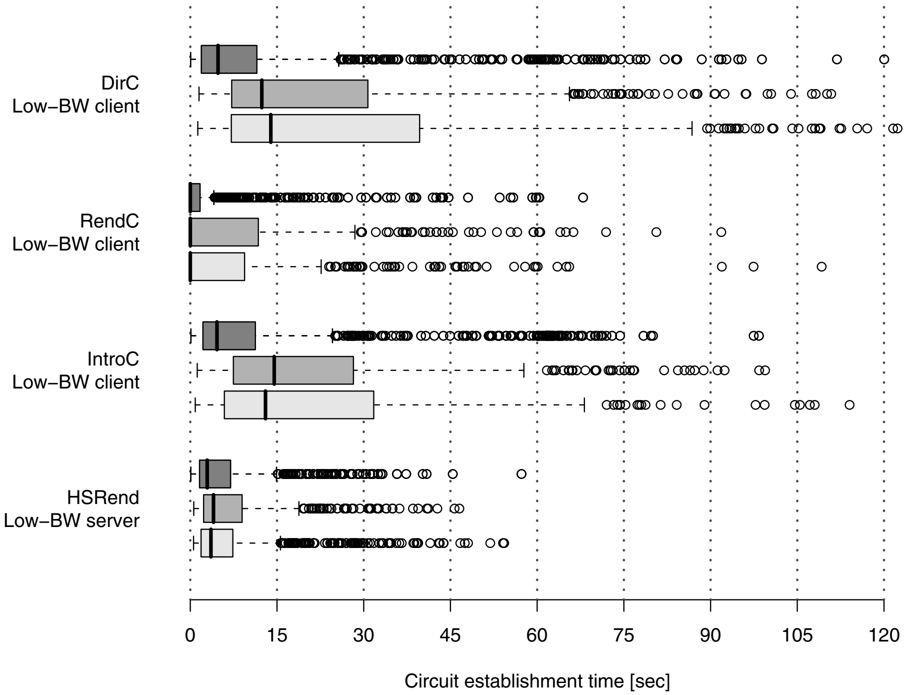

Fig. 8. Durations of circuit building [sec] in broadband (dark gray), EDGE (medium gray), and modem (light gray) access networks

in the same manner. Here, if possible, an existing circuit is cannibalized and extended to the respective relay. This extension has noticeable impact for the different access networks. This can be seen by the high difference, compared to the broadband network, in median values. In median, for both circuits, values range about 8 to 9 seconds higher for the limited access networks. It has to be mentioned that the data for the circuits to the directory server presented here is likely to be slightly higher than in the population. In some cases the time of the completion of this circuit could not be determined unambiguously from the log files among other circuits. We considered a slight over-estimation to be less critical and thus always chose the circuit that finished last.

Finally, the *hidden-service-side circuit to the rendezvous point (HSRend)* is built rather quickly and the broadband network is only slightly faster with around 1 second in difference for median in comparison to the low-bandwidth access networks. This circuit is cannibalized and extended to the rendezvous point. The hidden services in the measurements had fewer operations to perform than the clients. Bootstrapping was done once and fewer circuits had to be built during an attempt. So, the hidden services were more likely to have an existing internal circuit ready for cannibalization which explains why the building time for this circuit is much shorter than the time for building the client circuit to the introduction point or to the directory server.

Table 1. Binomial tests on circuit completion

Phase	Type	p_{30}	p_{40}	p_{45}
DirC	Broadband	$4.1e^{-8}$	$6.9e^{-17}$	$4.5e^{-19}$
	EDGE	1	0.04	$9.2e^{-4}$
	Modem	1	0.52	0.06
IntroC	Broadband	$1.1e^{-4}$	$9.4e^{-16}$	$2.8e^{-21}$
	EDGE	1	0.26	$1.2e^{-3}$
	Modem	1	0.58	0.03
HSRend	Broadband	$2.1e^{-14}$	$7.2e^{-25}$	$1.6e^{-27}$
	EDGE	$2.0e^{-12}$	$6.4e^{-24}$	$1.5e^{-26}$
	Modem	$2.1e^{-14}$	$3.0e^{-27}$	$1.8e^{-33}$

Starting with Tor version 0.2.1.7-alpha, the circuit timeout for the above circuits has been reduced to 30 seconds and in case of the introduction circuit, after 15 seconds a second attempt is started in parallel. It can now be observed with the present data if this new timeout is suitable also when limited access networks are in use. To determine the suitability, we applied binomial tests [8]. This type of test simply requires a binomial distribution of the data set. So, we split the set into two groups: on the one hand those attempts where the building of a certain circuit took less or equal time than for example 30 seconds and on the other hand those where it took more, up to 60 seconds. As the timeout is only relevant for a single attempt, we did not consider the absolute times as represented in Figure 8, but instead analyzed single attempts. We considered all successful attempts, no matter whether they were the first or second or maybe even third try to build a circuit to a certain relay. We set the percentage of completed circuits for considering a timeout as suitable, to 90%. Put in other words, concerning the binomial tests, we set the probability for a success to 0.9. It was important to find a measure for the timeout that guaranteed fault recognition, without cutting off too many attempts that would have finished later. Furthermore, Panchenko et al. showed that subsequent message transmission times over a circuit correlate to its building time [14]. So, cutting off circuit building at a reasonable limit should increase the performance of connection establishment and message transmission. We set the significance level to 5%. As the tests were three-fold, because of three connection types, we applied alpha adjustment which reduced the significance level to 1.66%. We did not perform binomial tests for the client circuit to the rendezvous point. This circuit is built almost immediately in most cases and a timeout reduction would not advance this. The results of the tests can be found in Table 1. It is quite obvious that a timeout of 60 seconds is too high in case of the hidden service circuit to the rendezvous point. Very low and significant p-values are achieved for all access network types. Thus, a timeout reduction to 30 seconds for this circuit is reasonable. But for the other circuits, the timeout reduction cannot be supported with the present data. While the broadband access network shows significant p-values also for 30 seconds, the low-bandwidth networks do not. Even a timeout of 40 seconds does not fit. The p-value of EDGE for the circuit to the directory server with 0.4 approaches a significant level, but other p-values still rank high. Only with a timeout of 45 seconds, p-values of EDGE become significant. The p-values of the modem access network, with 0.06 for the circuit to the directory server and 0.03 for the circuit to the introduction point, are still not significant but close to the significance level. As the timeout should be as convenient as possible for all access network types, a compromise needs to be chosen. On the one hand, a timeout of 45 seconds might still be slightly too low for the modem access network. On the other hand it is provably too high for the broadband access network. Being the convenient middle way, we propose a timeout of 45 seconds for both circuits.

Improving static timeouts may be a good first step. But as our measurements show, no timeout can fit all client environments equally well. A better approach would be to track circuit build times at the client and use these data to adjust a local timeout variable. By doing so, clients could even adapt to changing network environments. One such approach is described in a Tor proposal [1] which is not yet implemented, though.

7 Conclusion

We conducted comprehensive performance measurements of the usage of Tor in limited access networks. Our focus was the evaluation of the bootstrapping phase and sub-steps of hidden service access, especially circuit building and round-trip times. The bootstrapping phase has turned out to take significantly longer than expected over low-bandwidth access networks. The bottleneck in this phase is formed by the download of relay descriptors. We discussed advantages and disadvantages of different approaches to accelerate the bootstrapping process. The analysis of circuit building times showed that building or extending circuits is a major bottleneck in the process of accessing hidden services, especially when using low-bandwidth access networks. We conducted binomial tests to determine adequate timeouts for the circuits involved in hidden service access. We confirmed the usefulness of the timeout for the hidden service circuit to the rendezvous point. For the client circuit to the directory server and to the introduction point, we showed that the timeout is set too small when using low-bandwidth access networks. Instead we proposed a timeout of 45 seconds for these two circuits which would also fit the demands of the limited access networks. Furthermore, we found round-trip times to not differ strongly when using service-side low-bandwidth access networks. For the usage of client-side low-bandwidth access networks, the difference was more obvious.

The contribution of this paper is to compare a few selected uses of anonymity networks in low-bandwidth access networks. Future investigations might focus on other use cases, e.g., anonymous web surfing or downloading of large files. Also, other types of anonymity networks could be taken into consideration. Further future work includes separate measurements of bootstrapping and Tor hidden services in low-bandwidth environments. The limitation of the measurements to 6 minutes reduced the data that could be collected. With significantly shorter bootstrapping, the 6 minutes could be used to measure application-level message latency or throughput. For measurements of the bootstrapping, a higher timeout of 10 to 15 minutes could give more informative results.

Acknowledgements

We thank the reviewers for their helpful comments which gave us the idea to measure and include data on client bandwidths in the Tor network. Part of this work has been generously funded by the NLnet foundation.

References

1. Chen, F., Perry, M.: Improving Tor path selection. Tor Proposal 151, The Tor Project (July 2008), https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/151-path-selection-improvements.txt
2. Danezis, G., Serjantov, A.: Statistical disclosure or intersection attacks on anonymity systems. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 293–308. Springer, Heidelberg (2004)
3. Dingledine, R.: Keep controllers informed as Tor bootstraps. Tor Proposal 137, The Tor Project (July 2008), https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/137-bootstrap-phases.txt
4. Dingledine, R.: Clients download consensus + microdescriptors. Tor Proposal 158, The Tor Project (January 2009), https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/158-microdescriptors.txt
5. Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: Anderson, R. (ed.) Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006), Cambridge, UK (June 2006)
6. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium (August 2004)
7. GSM Assocication. Market Data Summary (2008), http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm
8. Hays, W.L.: Statistics. In: Holt, Rinehart, Winston (eds.), 3rd edn. (1981) ISBN: 0-03-056706-8
9. International Telecommunications Union. V.92: Enhancements to Recommendation V.90 (November 2000), http://www.itu.int/rec/T-REC-V.92-200011-I/en
10. Köpsell, S.: Low latency anonymous communication – how long are users willing to wait? In: Müller, G. (ed.) ETRICS 2006. LNCS, vol. 3995, pp. 221–237. Springer, Heidelberg (2006)
11. Loesing, K., Sandmann, W., Wilms, C., Wirtz, G.: Performance Measurements and Statistics of Tor Hidden Services. In: Proceedings of the 2008 International Symposium on Applications and the Internet (SAINT), Turku, Finland. IEEE CS Press, Los Alamitos (2008)
12. Øverlier, L., Syverson, P.: Improving efficiency and simplicity of Tor circuit establishment and hidden services. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 134–152. Springer, Heidelberg (2007)
13. Palfrader, P.: Download server descriptors on demand. Tor Proposal 141, The Tor Project (June 2008), https://svn.torproject.org/svn/tor/trunk/doc/spec/proposals/141-jit-sd-downloads.txt

14. Panchenko, A., Pimenidis, L., Renner, J.: Performance analysis of anonymous communication channels provided by Tor. In: ARES 2008: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Washington, DC, USA, pp. 221–228. IEEE Computer Society Press, Los Alamitos (2008)
15. Sauter, M.: Communication Systems for the Mobile Information Society. Wiley, Chichester (2006)
16. The Tor Project. Tor directory protocol, version 3 (2008), `https://svn.torproject.org/svn/tor/trunk/doc/spec/dir-spec.txt`
17. The Tor Project. Tor Rendezvous Specification (2008), `https://svn.torproject.org/svn/tor/trunk/doc/spec/rend-spec.txt`
18. Wendolsky, R., Herrmann, D., Federrath, H.: Performance comparison of low-latency anonymisation services from a user perspective. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 233–253. Springer, Heidelberg (2007)