Waiting for Anonymity: Understanding Delays in Tor

Prithula Dhungel[†], Moritz Steiner[‡],
Ivica Rimac[‡], Volker Hilt[‡], Keith W. Ross[†]
[†]Polytechnic Institute of NYU, Brooklyn, NY 11201
[‡]Bell Labs/Alcatel-Lucent, Holmdel, NJ, 07733
Email: pdhung01@students.poly.edu, {moritz, volkerh, rimac}@bell-labs.com, ross@poly.edu

***Abstract*—Although Tor is the most widely used system for providing anonymity services, its users often experience very high delays. Because much of Tor usage is for Web applications, which are sensitive to latency, it is critical to reduce delays in Tor. To take an important step in this direction, we seek an in-depth understanding of delays in Tor. By taking snapshots of the entire Tor network within a short time window, we are able to study the delay distribution of the entire router population. We also monitor delays introduced by individual Tor routers over extended periods of time. Our results indicate that apart from delays introduced by routers, overlay network latency also plays a significant role in delays in Tor. We have also observed that at any time, there exist huge differences in the delays introduced by different routers. Our results reveal key performance characteristics of Tor system behavior and provide valuable insights for improving the Tor performance.**

I. Introduction

Although Tor [1] [2] is the most widely used system for providing anonymity services, its users often experience very high delays [3]. Because much of Tor usage is for Web applications, which are sensitive to latency, it is critical to reduce delays in Tor.

In this paper, we seek an in-depth understanding of the delays in Tor, which is a pre-requisite for addressing Tor's poor delay performance. Specifically, we address the following important questions: (i) Are the delays in Tor mainly due to delays introduced by Tor routers as a result of heavy Tor traffic, or due to the extra latency each packet has to go through when hopping around multiple Tor routers across the globe? Although it is known to the Tor community that delays due to routers do play a role [3], there still lacks a comprehensive study to evaluate the relative contributions of the router delays and overlay latency. (ii) How much delay does each packet experience in Tor? (iii) Do delays differ significantly across routers? (iv) Is there a correlation between router delays and the available bandwidths the routers advertize? (v) Does a router's delay significantly vary over different time-scales?

We perform a detailed measurement study of delays in the Tor network to address the aforementioned questions. By taking snapshots of the entire Tor network within a short time window, we are able to study the delay distribution of the entire router population. Moreover, we monitor the delays of individual routers over extended periods of time. From the data set and statistics collected using this methodology, we derive new insights into the Tor network and the main factors that determine Tor delay performance.

Our study is the first to comprehensively analyze the relative contributions of overlay latency and router delays in the overall slowness of Tor. Our measurements reveal that there are huge differences in the delays introduced by different routers. Our results also indicate that apart from delays introduced by routers, overlay latency plays a significant role in delays in Tor. This finding should facilitate the analysis of solution space for reducing Tor delays, e.g., modifying Tor's path selection algorithm to prefer nearby routers or modifying the internal operation of the Tor routers.

The remainder of this paper is structured as follows. The next section outlines related work. Section III briefly describes Tor's operation. Section IV presents the measurement results for delays observed by cells passing through the Tor network. Section V presents measurement results for delays observed in routers, with the goal of determining the principal reason behind the large Tor delays. Section VI concludes the paper.

II. Related Work

Apart from studies related to improving its performance [4] [5] [6] [7] [8], there have also been measurement studies related to Tor. [9] presents a performance measurement of the Tor hidden service functionality, measuring the times required for different steps in the process of accessing a hidden service. McCoy et al. [10] performed a measurement study concluding that the web traffic makes up most of the connections in Tor. By analysing the information gathered at Tor directory servers from 2006 to 2009, the author of [11] observed trends in the network, like version update behavior of relay operators and changes in the distribution of relays to countries. [12] presents details of a suite consisting of tools that aid in performing various measurements at Tor nodes. [3] describes the current understanding of the Tor community on the performance issues of Tor.

To the best of our knowledge, this paper is the first in-depth measurement study of delays in the entire Tor network. Our study is also the first to comprehensively analyze the relative contributions of overlay latency and router delays to the overall slowness of Tor. Measuring and evaluating delays in Tor is complementary to earlier throughput measurement studies.

III. Design of Tor

To obtain anonymity, a user installs the Tor application called a Tor onion proxy (OP). The onion proxy selects 3 user-operated servers, called onion routers (ORs), to make a circuit from the OP through the ORs. The first, second, and third routers are respectively known as the entry, middle, and the exit routers. Each application packet is multiply encrypted and routed through these ORs. Each OR peels off a single layer of encryption from the packet and forwards it to the next OR in the circuit. Finally, when the packet reaches the last router, the router peels off the final layer of encryption and forwards the packet to the actual destination for the packet. The response packet from the destination to the OP is then routed via the same 3 routers in the opposite direction. In this manner, each OR in the circuit knows only the OR before and after it in the circuit. Therefore, the communication between the OP and the destination server is anonymous unless the entry and exit routers collude.

Since the ORs are volunteer operated, not all of them permit OPs to use them as exit routers. Throughout this paper, ORs that do not agree to be exit routers are referred to as "non-exit" routers. The ORs that do allow traffic to exit via them are referred to as "exit" routers. The exit routers also specify what kind of traffic is allowed to exit via them in the form of well defined rules. A set of such rules for an exit router form its exit policy. Some routers operators allow only certain types of traffic to exit via them whereas some have very permissive exit policies even allowing bulky streams like file sharing traffic.

The centralized authorities in Tor – the directory servers – keep track of the status of the ORs; for building circuits, the OPs download the list of available ORs from the directory servers. Each OR also reports to the directory servers the peak throughput it has observed for itself in the last 24 hours, monitored over each 10-second interval. This reported throughput is called the advertized bandwidth for the OR. When selecting routers for circuits, OPs select routers with higher advertized bandwidth with higher probabilities compared to ones with lower bandwidths.

Each OP always maintains an ordered list of entry nodes called "guard nodes". When choosing the first hop of a circuit, it chooses a router randomly from among the first 3 usable guard nodes. Each "exit" or "non-exit" router gets flagged as a "guard" node by the directory servers if it is stable (has a high uptime) and has an advertized bandwidth higher than the median of advertized bandwidths of all other routers.

IV. Delay in the Tor Network

In this section, we measure the extra delay faced by a single application packet in the Tor network as compared to sending the packet directly to the destination without using Tor.

A. Experiment Setup

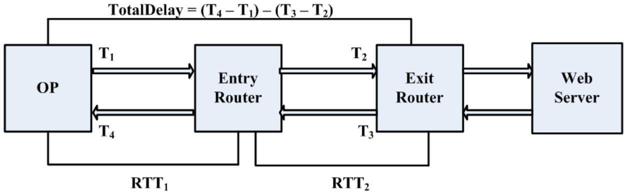

Fig. 1. Experiment Setup

As shown in Figure 1, the experiment setup consists of four entities - an onion proxy (OP), a web server, an exit router, and an entry router[1]. The OP, web server, and exit router are kept fixed whereas the entry router is selected one-by-one from the current list of running routers in the Tor network. The objective of this experimental setup is to a gain a deep understanding of the delay contributions from the different elements in the path.

As shown in the figure, a single cell of 512 bytes is sent from the OP to the web server via the circuit made through the entry router and the exit router. In doing so, we modified the source code in the OP to create 2-hop circuits instead of the 3-hop circuits made by default. As will be apparent later, using 2-hop circuits helps better decompose the delays faced by cells into various types of constituent delays.

Right before the cell is sent out from the OP, the current time (T_1) is noted. After travelling through the entry the router, where it experiences processing and queuing delays, the cell arrives at the exit router. Right after the cell is received at the exit router, the time is noted (T_2). The cell again goes through processing and queuing delays in the exit router before it is received at the web server. The web server then sends the response back to the exit router. Right before the response cell is sent out from the exit router, the time is again noted (T_3).

Finally, after going through new processing and queuing delays at the entry router, the response cell again arrives at the OP when the time (T_4)[2] is noted again. Immediately after this, two TCP SYN pings are sent - one from the OP to the entry router and the other from the exit router to the entry router (The reason for sending out these TCP SYN pings will be apparent shortly.) This procedure is repeated for different routers in the entry position, chosen one-by-one from the current list of running routers in Tor. The entire experiment was completed in a span of 40 minutes. We performed the experiment a number of times in a duration of 5 months between August 2009 and December 2009. In this section, we present results for the experiment we conducted on December 11, 2009 which is a good representative of the results for all other experiments.

The round trip delay between the OP and exit router (excluding the queuing/processing delay in the exit router) is:

$$TotalDelay = (T_4 - T_1) - (T_3 - T_2) \qquad (1)$$

For different entry routers, Figure 2 shows the distribution of *TotalDelay* (as well as other delays to be described subsequently). The data points have been sorted in ascending order of $TotalDelay$. To improve the accuracy of results, for each router, 10 measurements were done back to back and the average values have been plotted. Out of the 1597 router IPs that were collected at the start of the experiment,

[1] The OP and the exit router are running on the same university network

[2] For T_1 and T_4, time is noted right after the entire cell is written into the output buffer of the OP-OR connection and right after the entire cell is read from the input socket of the connection, respectively. For T_2 and T_3, time is noted right after the entire cell is read from the input socket of the OR-OR connection and right after the entire cell is written into the output socket, respectively.

1255 could be successfully pinged 10 times each. For the rest of the routers, either the circuit creation failed or not all 10 measurements were successful. Nevertheless, since the 1255 routers successfully measured were chosen randomly from the list of routers, we argue that the result is a good representative of the delay values in the whole Tor network. It can be observed that 30% of circuits have a *TotalDelay* higher than 1 second, much higher than delays observed by packets in an un-anonymized setting. In actual Tor circuits consisting of 3 routers, the delays would be even higher.

We argue that $TotalDelay$ can be decomposed into two parts - (i) Delay due to latency between OP and entry router plus the latency between entry router and exit router, and (ii) Queuing and processing delays in the entry router. We refer to the latter type of delay as the *RouterDelay*, and the former type of delay as the *latency*. In the following subsection, we investigate the relative contributions of latency and router delay on the $TotalDelay$ faced by each cell.

B. Relative Contributions of Router Delay and Latency

In order to analyze the relative contributions of latency and router delay, we introduce RTT_1 and RTT_2, which denote the RTTs for the TCP SYN messages from OP to entry router and from exit router to entry router, respectively.

The latency observed by each cell (between OP and entry router plus that between entry and exit routers) is:

$$L_D = RTT_1 + RTT_2 \tag{2}$$

The router delay observed by the cell due to the processing and queuing delays in the entry router is:

$$RouterDelay = TotalDelay - L_D \tag{3}$$

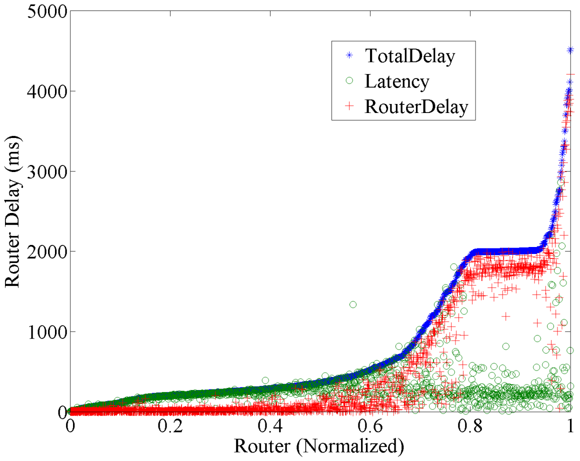

Fig. 2. Relative Contributions of Latency and Router Delay on the Total Delay (Data points sorted in ascending order of TotalDelay)

For each entry router, Figure 2 also shows the relative contributions of router delay and latency in the $TotalDelay$ faced by a Tor cell. It can be observed that for most cases when $TotalDelay$ is high ($>$ 1 sec), the router delay constitutes most of $TotalDelay$. Furthermore, it can be seen that delays introduced by different routers vary from a few milliseconds up to several seconds. Specifically, 57% of routers introduced routers delays less than 100 ms whereas 24% of them had delays of 1 second or more.

We performed the same experiment 8 times within 24 hours on November 14, 2009. The shape of the curve was the same in all the rounds. Furthermore, there were 60 router IPs that were successfully contacted during all 8 rounds. 7% of these routers had consistently high delays throughout ($>$ 1 sec), 17% had low delays throughout, and the rest of the routers had delays fluctuating from a few tens of milliseconds to a few seconds. This means a large fraction of the routers (76% here) have delays that dramatically fluctuate over a 1-day period. The possible reasons for such fluctuations are as follows: (i) The Tor router selection algorithm itself causes fluctuation in the amount of Tor traffic passing through any router. (ii) The machine the router is running on is running other applications and so there exist fluctuations in the network traffic coming in and out of the router; (iii) There are fluctuations in the CPU load in the router due to the non-Tor applications running over it (more on this in later sections).

Cells passing through any circuit that has one or more high-delay routers will face high round trip delays. Furthermore, even though the router delay seems to be the major contributor, 10% of the cases have latency L_D equal to 500 ms or more. For a default circuit length of 3 hops consisting of a middle router, the delay in the third link in the circuit will further add to the latency. Therefore, we conclude that overlay latency can also play a significant role in the delays observed by cells in Tor circuits (although not as much as router delays).

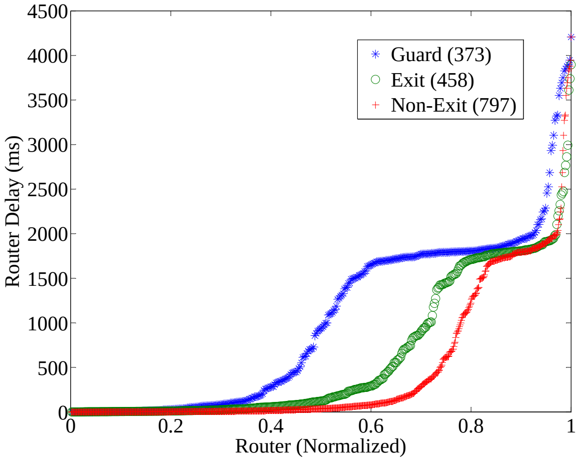

Fig. 3. Router Delay Distribution for “guard”, “exit” and “non-exit” Routers

In Figure 3 we depict the delay distributions of “guard”, “exit”, and “non-exit” routers, extracted from the snapshot experiment conducted on December 11. The total number of routers of each type that were successfully contacted are indicated in the legend. Unlike “non-exit” routers that are only selected for entry and middle positions in circuits, “exit” routers can be selected for all 3 positions. We would expect “exit” routers to have more Tor traffic than “non-exit” routers. Furthermore, we would expect routers that allow bulky traffic like file-sharing traffic to exit via them to introduce significantly higher delays. However, surprisingly, Figure 3 shows that delay distributions of “exit” and “non-exit” routers do not differ significantly. The percentage of “exit” routers above the 1 second delay mark is 28%, only slightly higher than that for “non-exit” routers (22%). In fact, only 14 out of 37 routers having delays higher than 2 seconds are “exit” routers. However, a much higher fraction of “guard” routers have high delays. 49% of “guard” routers have delays of 1 second or more. Also, 28 out of 37 routers that had delays higher than 2 seconds are “guard” routers.

V. Router Delay Analysis

In this section, we perform a more detailed analysis of router delays. Specifically, we check the variation in router delays over time. We also seek to understand the correlation, if any, of router delays with the corresponding router bandwidths.

A. Correlation with Advertized Bandwidth

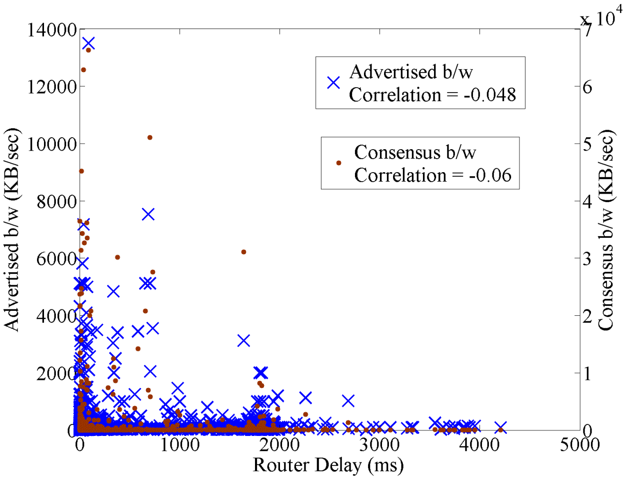

Fig. 4. Router Delay Vs Advertized Bandwidth and Consensus Bandwidth

Figure 4 shows the scatter plot of router delays and the corresponding advertized bandwidths on the left axis and consensus bandwidths[3] on the right axis. There are 3 key observations: (i) 35 (2.8%) of the routers measured have advertized bandwidths of 2 MB/sec or higher. 34 of these relatively high bandwidth routers have delays mostly in the order of a few hundred milliseconds. Figure 5 further shows that delays for a high bandwidth router (8 MB/sec; monitored over an extended period of time) are always in the order of only a few hundred milliseconds. These results are in agreement with the theoretical claim in [3] that the high bandwidth routers are selected with a lower probability compared to an optimal router selection strategy. (ii) 32 (2.5%) out of 37 routers with delay values more than 2 seconds have bandwidths equal to 150 KB/sec or less. This indicates that routers with highest delays are generally those with low bandwidths. (iii) *However, for the majority of the routers (95%), there is very low correlation between the advertized bandwidth of a router and its delay.*

Figure 4 also includes the scatter plot of router delays and consensus bandwidths. Note that consensus bandwidths are much higher than advertized bandwidths for many routers with high advertized bandwidths. Although a larger fraction of routers with high consensus bandwidths have delays in the order of a few hundred milliseconds, the correlation between router delay and consensus bandwidth is very low.

B. Variation in Router Delays

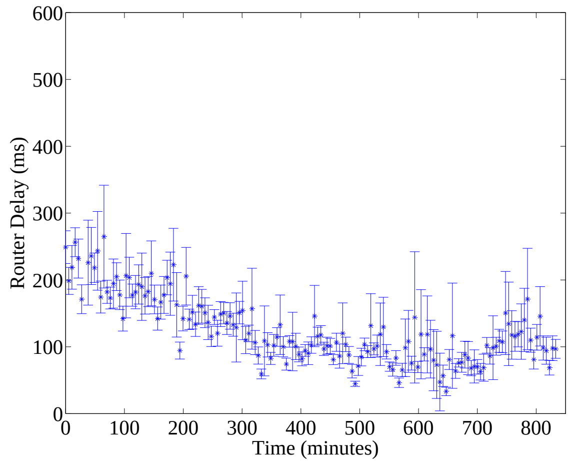

Fig. 5. Router Delay Variation (b/w = 8 MB/sec, 95% confidence interval)

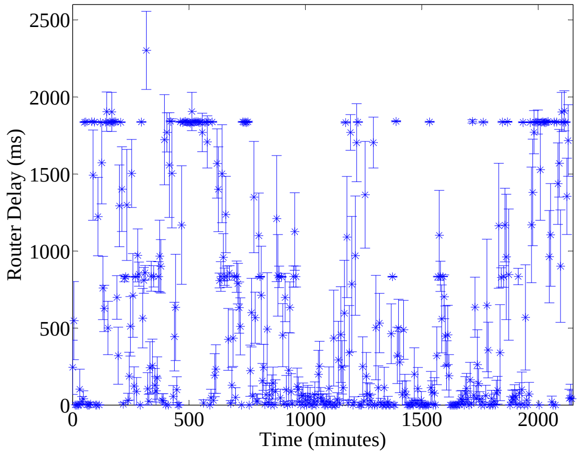

Fig. 6. Average Router Delay over Extended Time Period (b/w = 100 KB/sec, 95% confidence interval)

Figure 6 presents the delays for a low bandwidth router (100 KB/sec) over extended periods of time. The router was set up in our research lab network, configured as a “non-exit” router. In order to avoid loading the router with other applications, there were no extra programs running on the router. In the figure, the average delay faced by 15 measurement cells every 5 minutes has been plotted. The delays in this case

[3] Directory authorities reach a consensus on the actual bandwidth they think each router is capable of providing, based on active measurements [13], [14]

are fluctuating, which is the reason why the delay values for lower bandwidth routers in Figure 2 have no correlation with their advertized bandwidths. These routers sometimes have delays in order of seconds and sometimes as low as a few milliseconds. Since the router showed dramatic fluctuations in delay over time even with no extra interference, it is very likely that the Tor router-selection algorithm itself plays a role in the variation in delays across a given router.

C. Impact of Tor Token Buckets

Notice that a number of routers in Figure 2 have delay values very close to 2 seconds. Also, Figure 6 has a number of points around the 2 second mark. The logs at our research lab network router indicated that for cases when the router delay observed was close to 2 seconds, the Tor cell travelling from OP to the research lab router and the response cell from the exit router to the research lab router had to each wait for almost a second for relay read/write tokens to be available before they could be read from or written to the corresponding *input/output* socket buffers.[4] (Both cells are part of a single measurement out of 15 measurements taken in each round.) Note the narrow confidence intervals around the points close to the 2 seconds mark. This suggests that the router was handling a lot of Tor relay data at that point of time. The narrow confidence intervals for routers close to the 2 second mark in Figure 2 (not shown here due to lack of space) also indicate that these routers were most likely handling large volumes of Tor data and therefore the cells in all 10 measurements for each router were blocked by empty token buckets. The fraction of such routers (14% in Figure 2 with delay equal to 1.7 seconds or more and confidence interval equal to 400 ms or less) gives a lower bound for the fraction of routers overloaded with Tor traffic.

VI. Conclusion

To take an important step in improving the perceived delays in Tor, a thorough understanding of its delays is required. In this paper, we perform a detailed measurement study of delays in the entire Tor network. Our key findings are as follows: (i) Router delays are the principal contributors to delays in Tor. Some routers frequently introduce delays as high as a few seconds. At any instant of time, we observed 14% or more of the routers to be overloaded with Tor traffic. (ii) The router delay is not the only culprit. In almost 10% of circuits the overlay latency contributed more than 500 ms, which is much higher than delays in an un-anonymized setting. (iii) At any point in time, there exist huge differences in the delays introduced by different routers. (iv) Surprisingly, "exit" and "non-exit" routers showed similar delay distributions. (v) In general, "guard" routers introduced higher delay values than "non-guard" routers. (vi) Except for the routers with very high advertized bandwidths, there is no correlation between the delay introduced by a router and its advertized or consensus bandwidths. (vii) Except for the routers with very high advertized bandwidths, the delays for the routers dramatically fluctuate over time, ranging from a few milliseconds up to several seconds. This fluctuation is introduced by the Tor network itself and not due to the fluctuation in load from non-Tor applications that might be running in the Tor routers. $(viii)$ In the current router design, the cells often sit waiting for relay tokens to be available in the next time slot, before they can be read from or written to TCP socket buffers. This phenomenon occurs frequently when the router is handling a large amount of Tor traffic.

Our findings should facilitate the analysis of solution space for reducing Tor delays. For example, modifying OPs to actively keep track of delays introduced by different routers and choosing routers with low delay values for circuits serving delay-sensitive applications can be one approach to improve perceived delays; making the criterion for a router to be promoted into a "guard" to be less stringent might help by distributing loads across more "guards". Similarly, replenishing the token buckets more often than every 1 second should prevent the situation where cells sit waiting for the token buckets to get filled, and therefore improve the incurred delays.

We believe that the observations made in this paper will be useful to the Tor community in their next steps in improving the performance of Tor.

VII. Acknowledgement

We would like to thank Roger Dingledine for his insightful comments on an earlier draft of the paper.

References

[1] "Tor," http://www.torproject.org/index.html.en.

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 303–320.

[3] "Why is Tor slow and what are we going to do about it," https://svn.torproject.org/svn/tor/trunk/doc/roadmaps/2009-03-11-performance.pdf.

[4] L. Øverlier and P. Syverson, "Improving efficiency and simplicity of Tor circuit establishment and hidden services," in *Proceedings PETS 2007*.

[5] R. Snader and N. Borisov, "A tune-up for Tor: Improving security and performance in the Tor network," in *Proceedings of NDSS 2008*.

[6] S. J. Murdoch and R. N. M. Watson, "Metrics for security and performance in low-latency anonymity networks," in *Proceedings of PETS 2008*.

[7] A. Panchenko and J. Renner, "Path selection metrics for performance-improved onion routing," *Applications and the Internet, IEEE/IPSJ International Symposium on*, vol. 0, pp. 114–120, 2009.

[8] J. Reardon and I. Goldberg, "Improving tor using a tcp-over-dtls tunnel," in *Proceedings of 18th USENIX Security Symposium*, August 2009.

[9] K. Loesing, W. Sandmann, C. Wilms, and G. Wirtz, "Performance Measurements and Statistics of Tor Hidden Services," in *Proceedings of SAINT 2008*.

[10] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, "Shining light in dark places: Understanding the Tor network," in *Proceedings of PETS 2008*.

[11] K. Loesing, "Measuring the tor network from public directory information," in *HotPETs 2009*.

[12] M. Perry, "Torflow: Tor network analysis," in *HotPETs 2009*.

[13] "Authorities vote for bandwidth offsets in consensus," https://git.torproject.org/checkout/tor/master/doc/spec/proposals/160-bandwidth-offset.txt.

[14] "Computing bandwidth adjustments," https://git.torproject.org/checkout/tor/master/doc/spec/proposals/161-computing-bandwidth-adjustments.txt.

[4] Each OR uses a token bucket for the maximum number of bytes to be relayed per second, known as "RelayBandwidthBurst". After the token bucket for a particular second has been emptied, no further relay cells are read from or written into any of the TCP connection sockets before the token bucket is replenished at the beginning of next second.