Improving Performance and Anonymity in the Tor Network

Andriy Panchenko, Fabian Lanze, and Thomas Engel
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
{firstname.lastname}@uni.lu

***Abstract*—Anonymous communication aims to hide the relationship between communicating parties on the Internet. It is the technical basis for achieving privacy and overcoming censorship. Presently there are only a few systems that are of practical relevance for providing anonymity. One of the most widespread and well researched is Tor which is based on onion routing.**

Usage of Tor, however, often leads to long delays which are not tolerated by end-users. This, in return, discourages many of them from using the system and lowers the protection for the remaining ones. In this paper we analyze the bottlenecks in the Tor network and propose new methods of path selection that better utilize available capacities in the heterogeneous network and allow performance-improved onion routing. Our methods are based on the combination of remotely measured current load of the nodes and an estimation of their maximum capacity. We evaluate the proposed methods in a Tor network running in PlanetLab where we tried as far as possible to recreate real-world conditions. Finally, we present a practical approach to empirically analyze the strength of anonymity that different methods of path selection provide in comparison to each other. We show the risk of the currently used method for path selection in Tor and provide a countermeasure to protect against this risk by effectively detecting nodes that lie about their capacity.

Keywords: Privacy, Anonymous Communication, Onion Routing, Tor, Performance

I. Introduction

With the growth of the communication over the Internet, privacy issues become more and more important. Anonymous communication is a fundamental building block to protect privacy by obscuring relationships between communicating parties. Without this protection attackers are able to deduce information about the network addresses of involved senders and recipients. This is often enough to uniquely identify persons. Time, duration, and volume of communications can be used by attackers to infer further information such as the social relation between the communicating parties.

Many approaches have been proposed to provide anonymity at the network layer, e.g., Crowds [1] and Shalon [2], though only some of them have reached wide scale deployment and usage, e.g., Tor [3], I2P [4], and JAP [5] with Tor being the most widespread and popular system today. The Tor network is a circuit switched, low-latency anonymizing network. It is an implementation of the *onion routing* technology, which is based on routing TCP streams through randomly chosen paths in a network of *onion routers* (OR), while using layered encryption and decryption of the content (In fact, routers are not selected uniformely at random but in a weighted probabilistic way as we will explain later).

Currently, the publicly accessible Tor network consists of about 3,000 ORs[1] [6], while the daily number of users is estimated to be hundreds of thousands [7]. Acting as an overlay network, Tor is very dynamic: anybody can join it by running a router and thus offer available resources for the other users. Thus, the Tor network is heterogeneous with respect to available capacities of the nodes. As a consequence, the client usage of Tor leads to significant additional delays caused by the network layers and unevenly distributed capacities. These delays are often perceived by the users as unacceptable. A study indicates a tolerated latency of about 4 seconds for requesting a website [8]. Although some users may tolerate the low performance of Tor, e.g., due to censorship in their country, the number of those who are not willing to accept the performance degradation is significant [9].

Since the strength of protection in anonymization networks is usually linked to the number of active users, any user leaving the network weakens the protection for the remainder. The objective of our work is therefore to improve the performance of anonymization channels by better utilizing available capacities, while being able to control the quality of protection. According to a corresponding study [10], only half of the available bandwidth in the Tor network is actually used. In the light of the poor performance experienced by end-users, there is strong need for alternative path selection methods that allow for better load balancing and utilization of the available resources. However, a performance-optimized strategy for path selection can only be of practical relevance if it does not introduce a significant decrease of the anonymity provided by the system.

In Tor, routers are not chosen uniformly at random but in a weighted probabilistic manner to improve the circuit performance. Therefore, ORs are ranked for selection based on their bandwidth capacities. This path selection metric does not consider the current load and hence, cannot exploit the existing resources optimally. In order to create an optimized metric, the bottlenecks which cause the experienced delays must be identified. In particular, it is important to determine whether delays mainly occur on the ORs (i.e., the nodes) or if they are predominantly caused by the connections in the overlay

[1] as in August 2012.

topology (i.e., the edges). If overloaded nodes are the primary bottleneck regarding experienced latency and throughput (thus, the latter are the nodes' property and not the links'[2]) this would result in a simpler infrastructure for measuring these performance metrics, since only N values (N is the number of nodes in the network) instead of N^2 have to be considered. In this paper we examine the above mentioned issues, propose new metrics for ranking ORs, and evaluate these metrics in terms of the attainable performance and anonymity.

Contribution: Our contribution is as follows:

- we show that performance in the Tor network is mainly the property of *nodes* rather than *edges*. For a metric, it is absolutely sufficient to rank only nodes and, hence, resource and time expensive edge ranking can be omitted without any performance drawbacks;
- we propose and evaluate numerous metrics for path selection. These are based on the combination of the remotely measured current load of the nodes together with an estimation of their maximum capacity. To the best of our knowledge, ours is the most extensive evaluation that has ever been conducted;
- we show the risk of the currently used method for path selection in Tor and provide a countermeasure to protect against it by effectively detecting nodes lying about their capacity;
- we propose a comprehensive and objective metric enabling the comparison of path selection algorithms regarding the provided *degree of anonymity* and show that with our methods the users can significantly improve their network performance without sacrificing anonymity. Alternatively, users can drastically increase the performance by accepting a slightly reduced anonymity.

Roadmap: The paper is structured as follows: After introducing the fundamentals of the onion routing technology, we briefly discuss the current state of the art in path selection methods that is performed by the Tor clients. The summary of related works is presented afterwards, mentioning existing attempts to improve the performance in the Tor network. In Section V we address the fundamental question to be answered before proposing any metric for path selection, namely whether overloaded nodes or overloaded links constitute the major performance bottleneck. The main contribution of this work – proposal and evaluation of new metrics for performance-improved path selection in Tor – is presented in Section VI. Subsequently we provide an empirical anonymity analysis of the new metrics and show their robustness against nodes faking theier capacity estimation. A discussion of the results and their implications in the spot of a current paradigm change in the path selection metric of Tor concludes this paper.

II. Onion Routing Fundamentals

The Tor overlay network consists of single servers that are called *onion routers* (ORs). Each OR runs on an Internet end-host and maintains TLS connections to other ORs through which anonymized Internet communications are multiplexed. Therefore, end-users run an onion proxy (OP) that is listening locally for incoming TCP connections to redirect them as *streams* through the Tor network. To achieve this, the OP constructs *circuits* of encrypted connections through paths of weighted randomly selected onion routers (see Section III for more details). A Tor circuit, by default, consists of three individual hops, while each hop only knows its predecessor and its successor in the path. To avoid that the last node of a path learns the first, an additional third node is used. The default path length of three hops states a reasonable trade-off between security and performance. Depending on their position within a circuit, the routers are called *entry*, *middle* and *exit* nodes [11].

Clients choose paths for creating circuits by selecting three suitable servers from a list of all currently active routers, the so-called *directory*. Certain trusted nodes therefore act as *directory servers* and provide signed documents that are downloaded by users periodically via HTTP, including the *descriptors* of all currently known ORs. A router's descriptor contains, besides fundamental information such as the used IP address and port of the router, self-advertised information regarding its capacities.

During circuit creation, Diffie-Hellman key exchange is used to establish shared symmetric session keys with each router of a path. The OP encrypts all traffic that is to be sent over a circuit, using these keys in corresponding order. While relaying the data, every hop on the path removes or adds a layer of encryption, depending on the flow direction. This way only the exit node learns the final destination of a stream. Application data is generally transferred unencrypted on the link from the exit node to the destined Internet end-host, unless the user explicitly applies additional encryption, e.g., by using TLS/SSL. Consequently, the operators of exit nodes are in the first instance responsible for any abuse that is done using their nodes, which can lead to legal prosecution in some countries. For this reason node operators can specify a so called *exit policy* to narrow or prohibit connections to hosts outside of the Tor network (e.g., operators can particularly avoid their ORs being used as exit nodes).

Once a circuit is established, it can be used as a tunnel for arbitrary TCP streams through the Tor network, while many streams can share a single circuit. Proxies stop using a specific circuit after a configured amount of time (or data volume), which prevents users from certain profiling attacks. On the application layer, the SOCKS protocol is used to tunnel TCP traffic.

III. State of the Art in Path Selection

Ideally for anonymity, all Tor clients would select the nodes to be used in circuits uniformly from the set of all active routers. In this case attackers cannot influence the path selection of clients, except by operating more routers. However, this would lead to poor performance, as routers with a weak performance are chosen with the same probability as very powerful nodes having abundant resources.

[2]We use the terms *link* and *edge* as synonyms in this paper.

Therefore, the current state-of-the-art in path selection in Tor briefly works as follows: On startup, directory authorities are contacted to request a network status document and a list of nodes' descriptors. These include self-advertised bandwidth information about every router. Since descriptors are published by the routers themselves, the contained information cannot be considered as trustworthy however. To overcome this, since Tor version 0.2.2.6-alpha, the directory authorities actively measure the throughput of nodes and publish this information in the network status document (see Section VIII for more details). However, if for some reason no information about throughput is provided in the network status, the self-advertised information from the descriptor is used instead. As soon as enough directory information has been gathered, clients begin establishing circuits. A client maintains at least one general-purpose circuit in a preemptive manner, i.e., before there is any application request.

Choosing the first router on a path puts a major responsibility on it, since – as the network entry – it directly learns the IP address of the traffic initiator. Therefore, Tor clients make use of so-called *guard* nodes. They maintain a list of n routers (by default, $n = 3$) that have a long uptime and are known to be fast and stable. One of these long-term guards is selected by a client as the entry node for all of its circuits. This prevents a client from eventually ending up with a corrupted entry node as it chooses a different one for every new circuit (it is under certain circumstances possible to force anonymous clients building new circuits [12]).

The actual selection of middle and exit nodes is performed in a weighted probabilistic manner. A router is chosen with a probability proportional to its bandwidth, which is either actively measured by the authorities or self-advertised. If authorities do not provide bandwidth information for a router, there is an upper bound of believable bandwidth which is currently 10 MBps. This should prevent a router from claiming to have an unbelievably high or infinite bandwidth.

Exit nodes are considered for entry and middle positions only if the available total bandwidth of exit nodes is at least one third of the overall available bandwidth of all routers. In this case, their probability of being chosen is lowered in a weighted way in order to improve load balancing.

IV. Related Work

Rollyson [13] proposes a method to improve the performance in Tor by a modified method of middle node selection that is based on latencies between routers. Since link-wise latencies between ORs are not available, the author proposes using an approximation technique that is based on measuring latencies between responsible DNS servers [14]. However, since DNS servers are often located far from the actual hosts and latency between DNS servers does not reflect the load of the Tor nodes, the accuracy of the method is questionable [15].

Snader and Borisov [16] propose an opportunistic bandwidth measurement mechanism for Tor nodes. It is based on the idea of assigning a node's capacity equal to the median of the peak bandwidth that all other nodes recently experienced to the given one. According to the authors' measurements though, the opportunistic bandwidth estimation is less accurate than using self-advertised values from the descriptors. The former, however, cannot so easily be manipulated by malicious nodes (Bauer et al. [17] demonstrated in a small testbed that by artificially increasing bandwidth reports, an attacker can compromise 46% of all circuits while controlling only 6 out of 66 routers in a Tor network). Moreover, their method considers only the maximum capacity and neglects the current load.

In [15] authors study the performance of Tor under various circumstances and show the influence of geographical diversity of routers in paths on the performance of circuits. The authors justify that client performance can be improved by choosing routers that are located geographically close to clients, respectively destinations. The diversity of the nodes in a path, though, is an important substance to the security of anonymity systems [18], [19]. Choosing nodes located in different countries involves different jurisdictions, and thus, increases the protection of users. Additionally, if chosen routers are located close to each other, it is more likely that they belong to the same operator. Therefore, it is better to achieve performance improvements by other means, i.e., perform routing based on geo-independent performance metrics. In the follow-up work [20], the authors propose to use link-based RTT for path selection in the Tor network and evaluate its influence on performance and anonymity. Though link-based RTT may help to increase performance, its practicability is limited by scalability (N^2 measurements must be made in the network of N hosts). Contrary to this, we propose to use node-based RTT as described later in Section VI. Moreover, RTT is only one component of our metric.

Murdoch et al. [7] explore the effectiveness of path compromise in Tor. The main finding is that, in the presence of a node-rich but bandwidth-limited attacker, Tor's default path selection algorithm can offer improved protection compared to the uniform path selection algorithm. Thus, the vulnerability of path selection is not affected that much by the proliferation of botnets. These usually have a large number of nodes with a high geographical diversity, but poor upstream bandwidth.

Tang et al. [21] proposed a scheduling algorithm that allows bursty (i.e., with varying bandwidth demand) circuits to get higher priority over busy (i.e., with constant bandwidth demand) ones. The idea is to give priority to circuits that recently consumed only few resources over other circuits.

Pries et al. [22] analyzed performance in the Tor network in order to identify causes of performance shortfalls. The main three explanations that the authors could identify are as follows: (i) routers with low bandwidth have a high probability of being part of a path; (ii) worldwide scattering of the Tor network leads to large RTT values; (iii) insufficient number of servers for the given user base.

Dhungel et al. [23] conducted a study which shows that a significant delay in the Tor network occurs on the ORs. To improve performance, Chen et al. [24] propose to use fewer hops, restrict the selection of routers to those with the *fast* and

stable flags, and to limit the geographical distance between the nodes. All these measures significantly compromise anonymity and, moreover, Tor's currently used path selection method outperforms them.

AlSabah et al. [25] propose using multiple separate paths that join at a common exit node. This method requires the modification of the client and exit node and, according to the authors' evaluation, achieves only marginal improvement in the regular usage of Tor. A significant improvement was possible only for users of bridges (nodes that are used to overcome censorship) as these often suffer from low bandwidth.

To sum up, none of the related works considers the current load of the nodes. This does not allow to optimally utilize available capacities and, as their evaluation shows, allows at most a marginal performance improvement. Moreover, majority of the proposed approaches make use of information which is self-reported by the nodes. Hence, it is vulnerable to attacks where malicious nodes lie about their capacities and attract disproportionally many clients.

V. Problem Analysis

The fundamental question to be answered before proposing any metric for path selection is whether overloaded nodes or overloaded links are the primary performance bottlenecks in Tor. The observed correlation between latency and throughput [15] may be a sign of the same underlying cause for bad performance in both, namely, an overloaded node. Another indication of this is the fact that latencies in Tor are often much higher than those experienced on regular Internet connections (several seconds vs. a few hundred milliseconds). Since Tor uses three overlay nodes, the RTT in Tor should be roughly equal to three times the RTT on the Internet. It is, however, significantly higher. If the experienced latency and throughput are mainly due to overloaded nodes (thus, are the nodes' property and not the links') this would result in a simpler infrastructure for measuring these performance metrics, since only N values instead of N^2 have to be considered.

There is a need to differentiate the terms *node* and *edge* as used in this section. Our definition of "property of a node" refers not only to a node itself but also includes its connection to the Internet (i.e., link to the ISP). "Property of an edge" starts only from the point where it is possible to select different routes for a connection. This allows us, without loss of generality, to treat a congested link of a node as a property of a node (even though it is a link). As the consequence, we will have to consider only N values for measurements and path selection metrics that reflect the congestion of the connection to the Internet.

To investigate whether the performance of Tor is a property of the nodes or the edges, we conducted two experiments.

Experiment 1: The idea of the first experiment is to build a *set of circuits* where each circuit uses the same set of nodes but different links to connect them. This is practically achieved by varying the positions of ORs in every circuit. To this end, we fix three ORs and construct a set of three circuits such that each router takes the entry, middle, and exit position exactly once. Hence, these circuits share the same nodes but the links vary. If now the performance of the circuits within a set does not differ significantly, the conclusion could be drawn that the performance in Tor is mainly determined by the nodes and not by the edges.

We built 2,000 sets with random ORs in the real Tor network and measured RTT and throughput of the resulting circuits. For each circuit, the throughput was measured five times and the RTT was measured ten times. About 80% of all sets show a performance variance among the three circuits of less than 30% from its mean. Therefore, there is only a small performance fluctuation within each set. This gives an indication that performance in Tor is mainly a property of the nodes. The second experiment provides stronger evidence to support this hypothesis.

Experiment 2: In the second experiment we measured the RTTs to all ORs in the real Tor network using two different methods on the same link: A *TCP Ping* measures the time between sending a SYN packet and receiving a SYN-ACK answer while establishing a direct TCP connection to an OR. A *Tor ping* uses the method propsed in [20] to measure the RTT on a 1-hop circuit containing only the considered OR. Hence, the Tor ping is subject to additional processing on the application layer. In case the node is overloaded this method would experience additional delay. Each OR was measured 20 times with both methods and the respective median was taken as the outcome.

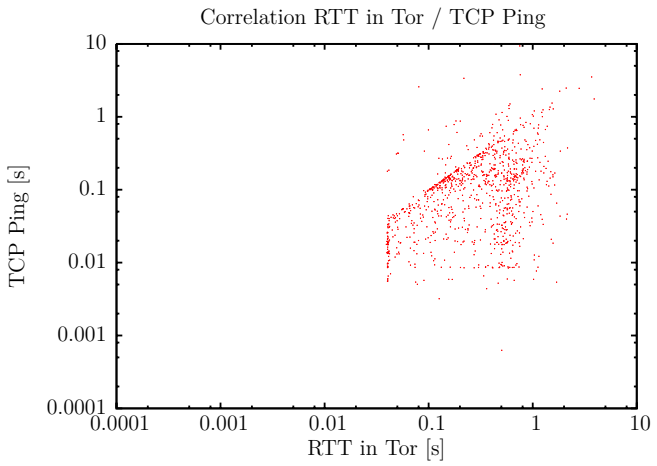

Fig. 1. Correlation of RTT in Tor (Tor ping) and TCP ping

Figure 1 shows the correlation between the two methods for measuring RTT. There is only a very small correlation between the TCP ping and the Tor ping: The Pearson's correlation coefficient of the measured values is 0.39. Since the same edge is used by both RTT measurements, the irregular delays are caused by processing delays on overloaded nodes. This supports our assumption that the performance in Tor is mainly determined by the properties of nodes. In the next section, comparing path selection metrics based on nodes and links, we will provide further evidence for this.

VI. Methods And Their Evaluation

In this section we propose and evaluate new methods for performance-improved path selection in onion routing. According to a corresponding study [10], only half of the available bandwidth in the Tor network is utilized. In the light of the poor performance experienced by end-users, there is strong need for alternative path selection methods that allow for better load balancing and utilization of the available resources.

Considering new methods, it is important to keep the random aspect in path selection in order to preserve anonymity. Therefore, our implementation chooses paths (either edges or nodes depending on the metric) in a weighted probabilistic manner from the set of all path proposals with respect to ranking indices. The probability of a router i (in case of a node-based selection) or of a link i (in case of an edge-based selection) being chosen for a path is proportional to its ranking index $perf(i)$. The ranking indices are calculated for any considered metric of path selection. In all metrics where higher values imply better performance the ranking indices simply correspond to the metric values. For metrics where lower values mean better performance (e.g., RTT), the metric values are inverted. We take the value of the 0.995 quantiles as the numerator in inversion instead of the maximum value. This prevents extreme outliers from strongly distorting the indices. For a detailed description of the path selection algorithm see [20].

We implemented and evaluated the following path selection metrics:

- *UNIFORM*: random selection from all available nodes. This method is used as a reference only;
- *DESC*: recall that currently in Tor, the descriptor value is used for selecting routers for anonymization paths if no bandwidth measurement could be performed by directory authorities. This value corresponds to the minimum of the bandwidth cap of the router and the observed maximum bandwidth within a 10 second interval during the last 24 hours. This value is reported by each router itself and updated every 18 hours, or when the *DESC* value changes by a factor of at least two;
- *CPU*: current CPU load;
- *RTT*: both link-based and node-based. Link-based RTT is measured from several nodes as described in [20]. Node-based RTT is measured directly from a single client to every node in the network through a one-hop circuit (hence, this node-based RTT corresponds to the Tor Ping described in Section V);
- *QWAIT-BW:* an estimation of the available capacity. The mean queuing delay of a cell in an OR i is mapped onto a scale $\delta_q(i) \in [0.2, 1]$. The metric is $perf(i) = \delta_q(i) \cdot \mathrm{BW}_{\mathrm{DESC}}$, the product of $\delta_q(i)$ with the maximum observed capacity (either by the directory authority, or, if this information is not available, self-advertised value from the descriptor). Hence, in this case the metric considers only 20% of the capacity for routers assigned to the last quantile in terms of load (expressed by mean queuing time);
- *IDLE-BW:* another way to estimate the available capacity, which is an enhanced modification of the BW-INFO metric proposed in [20]. Here, we use it to rate nodes instead of links. It is given by $perf(i) = (\mathrm{BW}_{\mathrm{DESC}} - \mathrm{BW}_{\mathrm{CURRENT}}) + \frac{\mathrm{BW}_{\mathrm{CURRENT}}}{\#\mathsf{CIRCS}+1}$, the difference between the maximum and the current bandwidth usage plus a fraction of the current bandwidth a new circuit gets in the case of a uniform distribution of capacities between circuits. This is due to the fact that if the current bandwidth is equal to the maximum it does not mean that the newly created circuit would not get any capacity. The current bandwidth is the average throughput within the last 20 minutes. It is observed by the ORs themselves and reported to the directory after every update, i.e., every 20 minutes;
- *NORM-QWAITBW+IDLEBW:* the values of QWAIT-BW and IDLE-BW are separately normalized (dividing by the value of the 0.95th quantile). Finally, the normalized values are multiplied.

A. Our test environment

Since some of the studied performance metrics require ORs to report specific statistics, this implies modifications to ORs' functionalities. Hence, not all path selection methods can be tested in the regular Tor network. In order to have a comprehensive evaluation of all the methods, we deployed a private Tor network in PlanetLab [26]. We used the vanilla Tor version $0.2.2.5 - alpha$ and tried as far as possible to recreate the real-world conditions of the real Tor network. To this end, in the same way as the real Tor network, the fraction of exit nodes was set to 35%. Since the performance of PlanetLab nodes was significantly higher (especially in terms of bandwidth) than those in the real Tor network, we applied two adjustments to tune the testbed and make it comparable to the real network. Firstly, we capped the bandwidth of the ORs. Secondly, we simulated clients that penetrate the network.

We defined 12 groups for the bandwidth cap: 20, 40, 60, 80, 100, 150, 200, 400, 600, 1000, 2000 kB/s, and a final group in which the bandwidth remained uncapped. PlanetLab ORs were randomly assigned to each group, with the number of ORs in each group corresponding to the bandwidth cap distribution in the real Tor network.

To our benefit, nodes in PlanetLab are already running several experiments in parallel, so therefore they are subjected to the load produced by others. In particular, the CPUs of most nodes are highly saturated. To simulate user traffic we installed so-called penetrators and streaming servers on every PlanetLab node. 80% of the penetrators build circuits according to *DESC*, whereas the remaining 20% use the *UNIFORM* path selection algorithm. Their behavior is rather simple: every penetrator continuously creates between one and three circuits in parallel and downloads from a random streaming server between 300 kB and 5,000 kB through each circuit.

In such a network consisting of between 500 and 600 ORs we tested our methods and performed evaluation and analysis.

This corresponds to about one half of all the available nodes in PlanetLab, since we took care to select only one node per institution to ensure geographical diversity. The number of available nodes fluctuated. We regarded this as benefit since it reflects the situation in the real Tor network where a similar fraction of nodes drop out and become available again later.

Each path selection strategy was evaluated over a period of 24 hours to alleviate the effect of variable load dependent on the time of the day. The bandwidth was measured transferring 300 kBytes of data and considering the last 200 kBytes (the first 100 kBytes were skipped to mitigate the effect of TCP slow start algorithm). These values were selected based on empirical observation. For RTT measurements, the `TCP-NODELAY` option was used to disable the Nagle algorithm [27] (this eliminates delays that may occur before the transmission of small packets). To reliably measure performance values, each measurement on a circuit was repeated 20 times from three identical dedicated computers in our lab with a 1 Gbps Internet connection. Path selection was subject to the same restrictions as in the real Tor network, e.g., all ORs in a path must be in a different /16 subnet.

If for some reason a node or an edge was not graded the default value, corresponding to the 0.95th quantile of all graded objects was used. The effect is to classify them the weakest 5% of objects, giving them some chance of still being selected for paths.

B. Evaluation

Figures 2, 3, 4 show the complementary cumulative distribution function (CCDF) for bandwidth, RTT, and jitter (extracted from the RTT measurements).

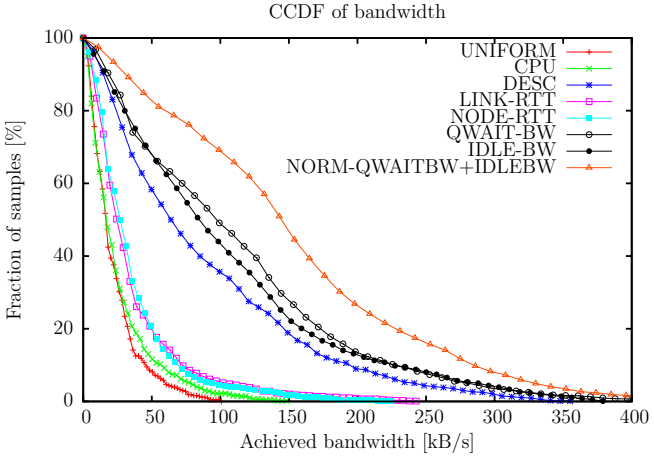

Fig. 2. Results for bandwidth

The *CPU* method does not perform significantly better than uniform path selection. Most probably, this is caused by the traits of PlanetLab: Because of the concurrent processes running on every node in the testbed, the CPU on most of them is loaded to almost 100%. Hence, all nodes get a very similar ranking leading to an almost uniform selection probability. Consequently, the resulting performance is comparably

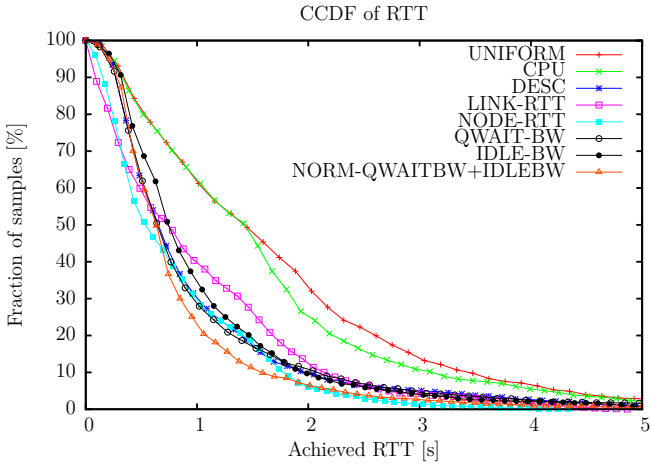

Fig. 3. Results for RTT

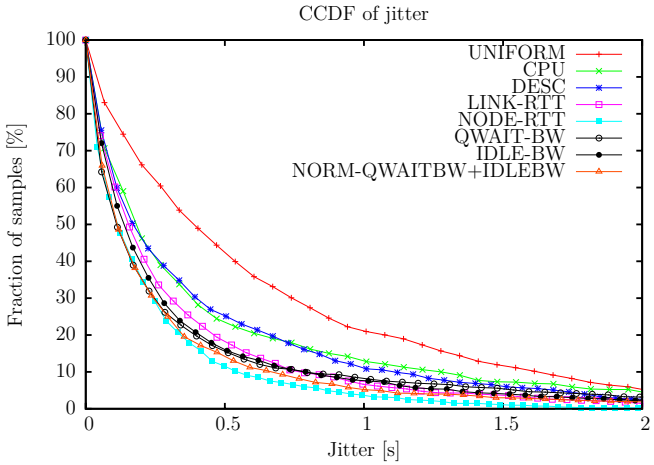

Fig. 4. Results for jitter

low. Thus, we conclude that a CPU related metric, although potentially beneficial, can not be comprehensively evaluated using a PlanetLab based testbed.

The strategies QWAIT-BW and IDLE-BW produce up to 22% more bandwidth than *DESC*. The normalized combination of QWAIT-BW and IDLE-BW increases the average throughput by 70% compared to *DESC* with an average achieved bandwidth of 152.48 kB/s. Jitter is halved, compared to *DESC*, whereas the RTT is only slightly decreased. Over 70% of all circuits achieve at least 100 kB/s (compared to only 35% in *DESC*).

One of the most important findings in this experiment is that the performance of the edge-based metric LINK-RTT is in most cases significantly worse than of the node-based NODE-RTT. Hence, this confirms our initial assumption that performance in the Tor network is mainly a property of a node and not of an edge.

However, the proposed combined metric (*NORM-QWAITBW+IDLEBW*), which achieves the best performance, relies on the self-reported load (i.e., the mean queuing time) and capacity indicators. Hence, malicious routers

	QRTT-BW	QWAIT-BW
Throughput	117.66 kB/s (**+28%**)	111kB/s (**+22%**)
RTT	0.84s (**-12.5%**)	0.96s (+/-0%)
Jitter	0.29s (**-27.5%**)	0.3s (**-25%**)

TABLE I
INFLUENCE OF REMOTELY MEASURABLE LOAD INDICATOR

can still manipulate their ranking. To overcome this, we propose substituting the self-reported load indicator with one that is remotely measurable: NODE-RTT (we assume that overloaded nodes would get higher processing delays reflected in their NODE-RTT values). As Table I shows, the new metric, QRTT-BW, which replaces QWAIT-BW and is built in the same way but using NODE-RTT instead of mean queuing delay, not only improves security (as it uses a remotely measurable load indicator instead of a self-reported), but also increases performance: there is a slight increase in bandwidth and decrease in both, RTT and jitter (the relative rates correspond to the difference compared with the *DESC* method). RTT is improved by 12.5% compared to *DESC* and QWAIT-BW. Hence, NODE-RTT is a better choice as a load indicator. With regard to the self-reported bandwidth information, in Section VII-A we will show how to exclude nodes that lie about their bandwidth capacity.

Please note that our results for basic path selection metrics *DESC* and *UNIFORM* differ from those obtained in [20] as since then the Tor network has undergone significant changes in terms of number of users, nodes, and available capacities.

VII. ANONYMITY ANALYSIS

Before any of the proposed modifications can be integrated into Tor, it is of great importance to study their impacts on the security and anonymity the system provides. Therefore, this section presents an approach for specifying a *degree of anonymity* for any given Tor network status and method of path selection to estimate the strength of anonymity that is achieved by end-users.

In [20] the authors proposed an entropy-based metric that considers a user to be *deanonymized* if an attacker owns the entry and the exit (EE) node of the user's circuit [3]. We recapitulate briefly here. To calculate an entropy $E(X)$ of a path selection metric the authors proposed using empirically measured probabilities of occurring EE-combinations, as well as the corresponding bounded degree of anonymity d from a sample of paths X. This is done using the following equations:

$$E(X) := -\sum_{0}^{N-1} p_i \cdot log_2(p_i) \tag{1}$$

$$d := 1 - \frac{E_{Max} - E(X)}{E_{Max}} = \frac{E(X)}{E_{Max}} \tag{2}$$

The N in Equation (1) denotes the number of distinct EE-combinations that occurred in the sample of paths X, while p_i represents the corresponding probability of the combination i to be chosen. The maximum entropy E_{Max}, which is used to calculate d in Equation (2), denotes the maximum value that can occur. This is the case when there is a uniform probability associated with each EE-combination. A path selection method having $d = 1.0$ will therefore choose any EE-combination with the same probability.

In this paper, we propose an improved metric to quantify the degree of anonymity. While owning the entry and exit nodes is the most serious attack in this context (the attacker can trivially deanonymize users by correlating input and output streams of a circuit [3], [28]), other positions should not be ignored. For example, no existing metric considers the fact that the same node can be present in all the paths. This is especially dangerous in the context of modern fingerprinting and traffic analysis attacks [29]. We refine the entropy metric by considering the distribution of middle (d_M) and exit nodes (d_E) in a similar way as is done for EE-combinations (d_{EE}). We deliberately omit the entry nodes: according to the guard concept [11], [12], the rotation of entry nodes is explicitly undesirable. We define the *combined degree of anonymity* as follows: $A(X) = \alpha d_{EE} + \beta d_E + \gamma d_M$, where α, β, and γ define the weight of single attack scenarios. In order to be comparable to the basic degree of anonymity, we require that $(\alpha + \beta + \gamma) = 1$. This ensures that the values of $A(X)$ are between zero and one. As already mentioned, the most dangerous situation, which directly leads to deanonymization, is when the attacker controls both the exit and the entry nodes. Hence, the coefficient α should be given the highest value, so that d_{EE} gets the highest weight. A corrupt exit node is possibly in a position to perform extensive profiling of the users. New circuits originating from the same user can be identified, e.g., with the help of *HTTP cookies*. If some time later the user communicates personally identifiable information through a circuit ending at the same exit node and the exit node can link the originator of the circuit to some previously established profile, this exit node can link the hitherto existing profile to the concrete identity. Consequently the user is deanonymized. The middle node is not able to deanonymize users; however, it knows which EE-combination to contact in order to perform this task. According to this reasoning, the following inequality should hold: $\alpha \gg \beta > \gamma$. Hence, in our evaluation we selected the weighting for the coefficients as follows: $\alpha = 0.8$, $\beta = 0.15$, $\gamma = 0.05$. In case different coefficients are selected that are complient with the provided above specifications, this would only slightly change the result. The tendency would remain unchanged: the more uniform the distribution of selected nodes on considered positions is, the better is the anonymity.

To evaluate the degree of anonymity according to different anonymity metrics we simulated the creation of 50,000 paths for each metric in the network of 500 PlanetLab nodes.

Figures 5 and 6 show the results of the evaluation. *UNIFORM* offers the highest possible anonymity, but at the price of very poor performance. As expected, improved performance implies less anonymity. However, the QRTT-BW method significantly improves performance and provides a marginally improved degree of anonymity compared to the currently

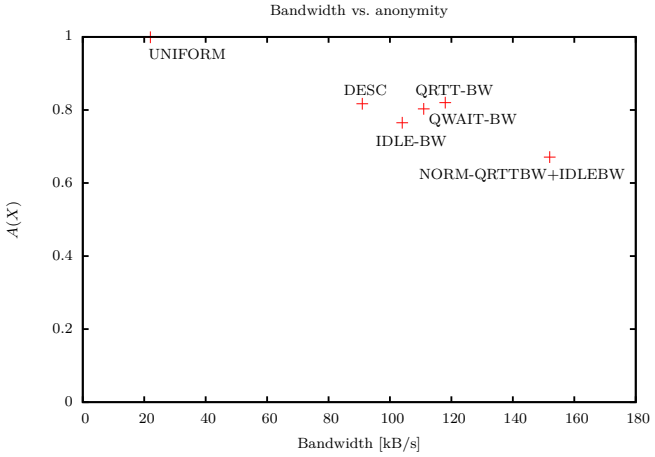

Fig. 5. Performance (Bandwidth) vs. anonymity

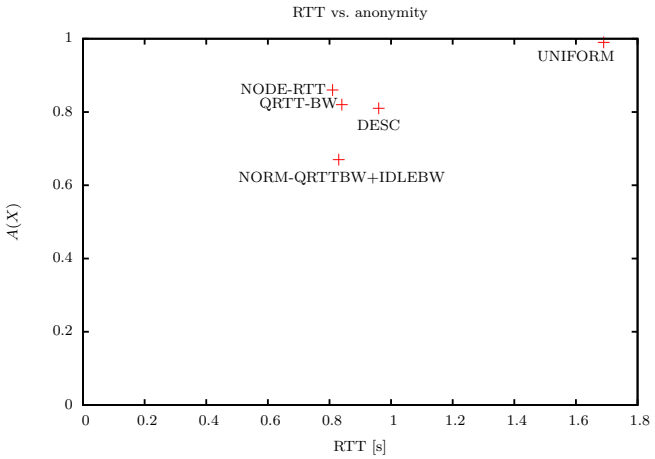

Fig. 6. Performance (RTT) vs. anonymity

used *DESC* method. Moreover, the normalized combination of QRTTBW and IDLEBW provides drastically better performance at the price of only slightly reduced anonymity. Users are likely to consider that the reduction of anonymity is justifiable in the light of greatly improved performance. In terms of RTT, the NODE-RTT and QRTT-BW methods not only improve the performance, but also increase anonymity when compared to the currently used method *DESC*.

A. Influence of Malicious Nodes

In this section we consider the scenario where the attacker controls one or more nodes and lies about their capacities, i.e., provides intentionally high *DESC* values in order to attract more traffic. We call these nodes malicious. A potential danger of this behavior is that, by investing few resources (i.e., bandwidth), the attacker may be present on a disproportionately high fraction of anonymization paths and compromise their anonymity. In this section we evaluate possible threats and propose a simple yet effective countermeasure to mitigate this kind of attack.

We want to emphasize that the new metric proposed by the Tor developers [30], [31], namely the actively measured bandwidth by the directory authorities, according to the authors themselves, does not eliminate the attack described above. Since the performance probes are generated by only a few known nodes, an attacker is able to allocate more resources to these probes, and thus get higher rankings than deserved. Alternatively, if malicious nodes simply drop such probe circuits, the clients would revert to using self-advertised bandwidth values from the descriptors, which can easily be faked.

The basic idea for the protection strategy is that the discrepancy between the actual and the advertised performance can be detected. If a node that does not have the advertised capacity is used by disproportionately many clients, its load will increase and performance will degrade significantly. The node's RTT, which can be measured remotely, is the metric used to detect this disproportional load. We propose a method called Exclude High Latency Nodes (EHL) to defend against the above mentioned attack. The EHL add-on collects 10 OR proposals for each position within a path (according to a selected base metric such as DESC) and then randomly selects one of five ORs having the smallest RTTs from the candidates in each position.

We performed the evaluation in the PlanetLab testbed with 500 nodes and created 50,000 path proposals. Two methods for path selection were tested: *DESC* and *DESC* with the EHL add-on. For each position within a path (entry, middle, and exit) we created 1, 2, 3, and 5 malicious nodes in the sense that their actual bandwidth was capped to 100 kB/s whereas their bandwidth value in the descriptor was set to 5,000 kB/s in order to put the ORs on top positions in the list of ORs according to *DESC*. We call a circuit *contaminated* if it includes at least one malicious node. A circuit is *compromised* if both entry and exit node are malicious.

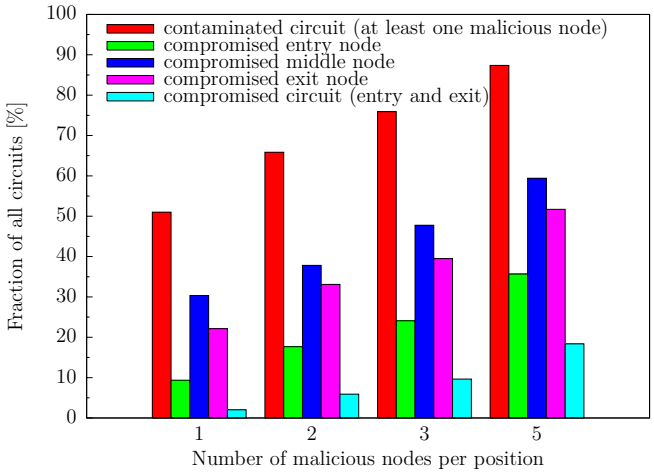

Fig. 7. Influence of cheating routers on DESC

Figure 7 shows the influence of malicious nodes that lie about their bandwidth on *DESC* path selection. Astonishingly, even having only one malicious node per position (i.e., 3 out of 500 nodes are malicious) leads to 50% contaminated circuits when using *DESC*. Having 5 malicious nodes per position, i.e.,

15 out of 500 nodes belong to the attacker, leads to almost 90% of contaminated circuits, with approximately every 5th circuit being compromised. The lower fraction of compromised entry nodes is due to the fact that the fraction of guard nodes in the system is relatively high (as in the real Tor network). The higher fraction of guard nodes at a fixed number of malicious nodes reduces the overall number of compromised nodes compared to other positions.

Applying the *DESC* method together with the EHL add-on shows completely opposite results. Having one and two malicious nodes per position does not lead to a single contaminated circuit after generating 50,000 path proposals. Their fraction in case of three and five malicious nodes per position is correspondingly 0.004% and 0.3%. Not a single circuit is compromised in this case.

As our evaluations show, malicious nodes that fake their bandwidth estimation constitute a big threat for the *DESC* path selection and for the method of actively measuring bandwidth as currently performed in the Tor network. However, the EHL add-on is able to completely eliminate this threat.

VIII. Discussion

As briefly mentioned before, starting from version 0.2.2.6-alpha, the Tor developers made a paradigm change in the path selection metric. Contrary to the *DESC* metric, which is based on a passive observation of bandwidth by the nodes themselves, the new metric (which we call *BW-AUTH*) is based on active measurements of the throughput of ORs. These measurements are executed by the directory authorities [30], [31]. The idea is to select two ORs with a similar performance according to the *DESC* metric for a two-hop circuit and to transmit between 128 kB and 2 MB of data, depending on the *DESC* value. The higher the *DESC* value, the more data will be transmitted. This procedure is repeated until every OR has participated in at least five measurement circuits. After this, the average stream throughput of every node is derived. The BW-AUTH metric is a weighted moving average that is produced by a product of the derived mean throughput and the ratio between this throughput and the average throughput of the group of 50 similar ORs according to *DESC*.

This implies that the BW-AUTH metric does not estimate the maximum capacity of ORs, but rather their available capacity, similar to the approach of IDLE-BW. The major difference is that the estimation is made by actively transmitting data by means of node scanners installed on the already highly loaded directory authorities. Active measuring of throughput may produce a tremendous load in the network. The fact that measurements are made by directory authorities additionally burdens the overloaded authorities. In contrast, the RTT metric measurements that we propose are very lightweight and do not put much load on the network.

Even though the performance of the BW-AUTH method is measured remotely and does not rely on self-reported values, the approach is still susceptible to manipulations. Since two-hop circuits are used to measure the throughput, the measurements can be trivially identified. The entry node checks whether the previous hop is the directory authority; the exit node checks whether the requested object is the file used for the throughput measurement. If a performance measurement is detected, malicious nodes may allocate more resources to the measuring circuit and, hence, produce a falsely high estimation of the available bandwidth. Another option would be to simply tear down the measuring circuit. If performance cannot be externally measured, the *DESC* value is used instead. Recall that *DESC* values are reported by the routers themselves, thus can be easily manipulated. Yet another option would be to restart the OR under a new ID if poor performance has been detected and reported by the scanners. This would force clients to use the *DESC* values, as no bandwidth information will initially be provided by the authorities. Hence, currently in the Tor network there is still a plethora of ways to abuse the system and to cheat concerning the available capacities. Even after the paradigm change in Tor, the EHL add-on we propose is a valuable and effective tool to exclude malicious nodes that lie about their capacities. Please note that our RTT measurements can also be detected. However, overloaded nodes are not able to improve the RTT measurement as they cannot arbitrarily prioritize Tor cells.

To investigate implications of the new path selection metric, we evaluated the BW-AUTH method in our test environment and compared it to *DESC*. The performance gain of metrics after substituting *DESC* with BW-AUTH in relation to BW-AUTH is proportional to the gain in unmodified metrics with respect to *DESC*. Hence, it is possible to say that the improved methods of path selection remain valid if the *DESC* values are replaced by a more accurate metric, namely the remotely measurable throughput.

It is of great importance to answer the question whether the proposed methods of path selection are transferable, i.e., also have the same or similar effect on the Tor network "in the wild". To this end, we evaluated our methods in the real Tor network. Obviously, we could only do this for the methods that do not require any changes in onion routers, i.e., BW-AUTH, *DESC*, QRTT-BW, and QRTT-BW-AUTH. The performance ratio between these metrics in the real Tor network was very similar to the performance ratio between the metrics in the PlanetLab testbed (e.g., QRTT-BW achieves about 30% more bandwidth and 15% less RTT than *DESC*). Hence, we are able to conclude that our results are transferable to the real Tor network and expect similar benefits from the remaining methods which could only be evaluated in the PlanetLab testbed.

IX. Conclusions

In this paper we have shown that the performance in the current Tor network is mainly a property of nodes and not edges. Hence, resource and time intensive edge ranking can be omitted without any performance drawbacks. Most notably, we proposed new metrics for measuring performance in anonymizing overlay networks such as Tor, while performing path selection based on the results of these measurements. The major advantage of our metrics is that they can increase

the performance up to 70% by considering current load of the nodes and allow better utilization of available capacities. Moreover, our metrics are either remotely measurable or allow exclusion of nodes cheating about their capacities. This is done in a lightweight way, without putting any significant load on the network.

In order to control the degree of anonymity, we presented an approach for practically estimating the strength of anonymity that is provided by different methods of path selection. The results show that by applying our methods, users can obtain a significant increase in performance without harming their anonymity. Alternatively, users can get a dramatic performance boost with little sacrifice in anonymity.

Finally, we have shown the risk of the currently used method for path selection in Tor and proposed a countermeasure to protect against it by effectively detecting nodes that lie about their capacity.

ACKNOWLEDGEMENTS

This work has been supported by the National Research Fund of Luxembourg (FNR) within the CORE project MOVE, AFR grant, and the EU FP7 project OUTSMART.

REFERENCES

[1] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, June 1998.

[2] A. Panchenko, B. Westermann, L. Pimenidis, and C. Andersson, "Shalon: Lightweight anonymization based on open standards," in *Proceedings of the 18th International Conference on Computer Communications and Networks (IEEE ICCCN 2009)*. IEEE Computer Society Press, Aug. 2009.

[3] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium*. USENIX Association, Aug. 2004, pp. 303–320.

[4] M. Herrmann and C. Grothoff, "Privacy Implications of Performance-Based Peer Selection by Onion Routers: A Real-World Case Study using I2P," in *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)*, Waterloo, Canada, July 2011.

[5] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: A System for Anonymous and Unobservable Internet Access," in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, ser. Lecture Notes in Computer Science, H. Federrath, Ed., vol. 2009. Springer-Verlag, Jul. 2000, pp. 115–129.

[6] "Tor Network Status," http://torstatus.blutmagie.de/.

[7] S. J. Murdoch and R. N. Watson, "Metrics for security and performance in low-latency anonymity systems," in *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, ser. Lecture Notes in Computer Science, N. Borisov and I. Goldberg, Eds., vol. 5134. Springer-Verlag, Jul. 2008, pp. 115–132.

[8] 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 (PET 2007)*, ser. Lecture Notes in Computer Science, N. Borisov and P. Golle, Eds., vol. 4776. Springer-Verlag, Jun. 2007, pp. 233–253.

[9] S. Köpsell, "Low Latency Anonymous Communication – How Long Are Users Willing to Wait?" in *Emerging Trends in Information and Communication Security*, ser. Lecture Notes in Computer Science, G. Müller, Ed., vol. 3995. Springer-Verlag, Jun. 2006, pp. 221–237.

[10] K. Loesing, "Measuring the Tor Network from Public Directory Information," https://metrics.torproject.org/papers/hotpets09.pdf, June 2009.

[11] R. Dingledine and N. Mathewson, "Tor Path Specification," https://gitweb.torproject.org/torspec.git?a=blob_plain;hb=HEAD;f=path-spec.txt.

[12] L. Øverlier and P. Syverson, "Locating hidden servers," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE CS, May 2006.

[13] S. Rollyson, "Improving Tor Onion Routing Client Latency," Georgia Tech College of Computing, Tech. Rep., 2006.

[14] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating Latency between Arbitrary Internet End Hosts," in *Proceedings of the SIGCOMM Internet Measurement Workshop (IMW 2002)*, Marseille, France, November 2002.

[15] A. Panchenko, L. Pimenidis, and J. Renner, "Performance Analysis of Anonymous Communication Channels Provided by Tor," in *Proceedings of the Third International Conference on Availability, Reliability and Security (ARES 2008)*. Barcelona, Spain: IEEE Computer Society Press, March 2008.

[16] R. Snader and N. Borisov, "A Tune-up for Tor: Improving Security and Performance in the Tor Network," in *Proceedings of the Network and Distributed Security Symposium - NDSS '08*. Internet Society, February 2008.

[17] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against anonymous systems," University of Colorado at Boulder, Tech. Rep. CU-CS-1025-07, Feb. 2007.

[18] S. J. Murdoch and P. Zielinski, "Sampled Traffic Analysis by Internet-Exchange-Level Adversaries," in *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, ser. Lecture Notes in Computer Science, N. Borisov and P. Golle, Eds., vol. 4776. Springer-Verlag, Jun. 2007, pp. 167–183.

[19] N. Feamster and R. Dingledine, "Location Diversity in Anonymity Networks," in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2004)*, Washington, DC, USA, October 2004.

[20] A. Panchenko and J. Renner, "Path selection metrics for performance-improved onion routing," in *Proceedings of the 9th IEEE/IPSJ Symposium on Applications and the Internet (IEEE SAINT 2009)*. Seattle, USA: IEEE Computer Society Press, July 2009.

[21] C. Tang and I. Goldberg, "An improved algorithm for tor circuit scheduling," in *Proceedings of the 17th ACM conference on Computer and communications security*, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 329–339.

[22] R. Pries, W. Yu, S. Graham, and X. Fu, "On performance bottleneck of anonymous communication networks," in *IPDPS*. IEEE, 2008, pp. 1–11.

[23] P. Dhungel, M. Steiner, I. Rimac, V. Hilt, and K. Ross, "Waiting for anonymity: Understanding delays in the Tor overlay," in *Peer-to-Peer Computing (P2P), 2010 IEEE Tenth International Conference on*, aug. 2010, pp. 1 –4.

[24] F. T. Chen and J. Pasquale, "Toward improving path selection in Tor," in *Proceedings of IEEE GlobeCom 2010*, Miami, FL, December 2010.

[25] M. AlSabah, K. Bauer, T. Elahi, and I. Goldberg, "The path less travelled: Overcoming Tor's bottlenecks with multipaths," University of Waterloo Centre for Applied Cryptographic Research, Technical Report CACR 2011-29, February 2012.

[26] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 3–12, 2003.

[27] J. Nagle, "Congestion control in IP/TCP internetworks, RFC 896, Internet Engineering Task Force," ftp://ftp.isi.edu/in-notes/rfc896.txt, January 1984.

[28] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A new cell counter based attack against tor," in *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 578–589.

[29] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 18th ACM Computer and Communications Security (ACM CCS) Workshop on Privacy in the Electronic Society (WPES 2011)*. Chicago, Illinois, USA: ACM Press, Oct. 2011.

[30] R. Dingledine, "Proposal 160: Authorities vote for bandwidth offsets in consensus," https://gitweb.torproject.org/torspec.git/blob_plain/HEAD:/proposals/160-bandwidth-offset.txt, May 2009.

[31] M. Perry, "Proposal 161: Computing bandwidth adjustments," http://gitweb.torproject.org//tor.git?a=blob;f=doc/spec/proposals/161-computing-bandwidth-adjustments.txt, May 2009.