Path Selection Metrics for Performance-Improved Onion Routing

Andriy Panchenko
RWTH Aachen University, Germany
panchenko@cs.rwth-aachen.de

Johannes Renner
JonDos GmbH, Germany
johannes.renner@jondos.de

***Abstract*—Providing anonymity for users on the Internet is a very challenging and difficult task. Currently there are only a few systems that are of practical relevance for the provision of low-latency anonymity. One of the most important to mention is Tor which is based on onion routing. Practical client usage of Tor often leads to delays that are not tolerated by the average end-user, which, in return, discourages many of them from using the system. In this paper we propose new methods of path selection that allow performance-improved onion routing. These are based on actively measured latencies and estimations of available link-wise capacities using passive observations of throughput. We evaluate the proposed methods in the public Tor network and present a practical approach to empirically analyze the strength of anonymity certain methods of path selection provide in comparison to each other.**

I. Introduction

Anonymous communication deals with hiding relationships between communicating parties. Many approaches have been proposed in order to provide protection on the network layer, though only some of them have been implemented in practice. One of the most popular and widespread systems today is Tor [1]. The Tor network is a circuit switched, low-latency anonymizing network. It is an implementation of the so-called *onion routing* technology, that is based on routing TCP streams through randomly chosen paths in a network of *onion routers* (ORs), while using layered encryption of the content. To anonymize Internet communications, 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 randomly chosen onion routers. A Tor circuit, per default, consists of three individual hops, while each hop only knows its predecessor and its successor on the path. The default path length of three hops states a reasonable trade-off between security and performance.

Currently, the publicly accessible Tor network consists of about 2,000 servers, whereas the number of users is estimated to be hundreds of thousands [2]. The overlay network itself is very dynamic: everybody can join it by running a server and thus offer available resources for the other users. Client usage of Tor though, often leads to significant additional delays caused by the network layers. These delays are often perceived as unnecessary and unaccepted by the end-users, who then most likely choose to continue surfing without Tor. Since the strength of anonymity provided by such a system is usually linked to its number of users, the protection for the remainders is weakened with any user leaving the network. The objective of our work is therefore to improve the quality of service of the anonymization channels, while being able to control the quality of protection. In order to achieve this, we propose new performance-driven methods of path selection for enabling QoS-improved onion routing. These are based on two orthogonal metrics: actively measured latencies and estimated available channel capacities using passive observations of link-wise throughput. Further, we evaluate the proposed methods regarding the achieved performance, compare them to each other, as well as to the vanilla Tor, and make a step towards analyzing the impact of alternative path selection on the strength of anonymity.

II. State of the Art in Path Selection

Ideally, all clients participating in the network would select the nodes to be used in virtual circuits uniformly from the set of all currently active routers. When no metrics are involved, attackers can in no way influence the path selection of clients, except for operating more routers. This method offers the maximum achievable anonymity, but at the cost of performance. The latter is due to the fact that routers with a weak performance are chosen with the same probability as very powerful nodes having abundant resources.

Therefore, the current state of the art of path selection in Tor is as follows: On startup, directory services are contacted to request a list of signed descriptors. These include self-advertised bandwidth information about every router. Since descriptors are published by the routers themselves, contained information cannot be considered as trustworthy though. As soon as enough directory information is gathered, clients begin establishing circuits. A client maintains at least one general purpose circuit in a preemptive way, i.e. before there is any application request.

Depending on their position within a path, the routers on a circuit are called *entry*, *middle* and *exit* nodes [3]. Choosing the first router on a path puts a major responsibility on it, since – as the network entry – it directly learns the IP of the traffic initiator. Therefore, Tor clients make use of so-called *guard* nodes. This means, they maintain a list of n (the default is $n = 3$) routers that have a long uptime and are

known to be fast and stable. One of these long-term guards is then picked as the entry node for all of this client's circuits. This way it is avoided that a client will eventually end up with a corrupted entry node when choosing a different one with every new circuit.

The actual selection of middle and exit nodes is performed in a weighted probabilistic manner, hence the probability of a router to be chosen for a path is proportional to its advertised bandwidth. To avoid that a router claims to have infinite bandwidth, there is an upper bound of 10 MBps. 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 bandwidth is lowered in a weighted way in order to not choose possible exit nodes on other positions too often for contributing towards load balancing among the nodes.

III. Related Work

Rollyson [4] proposes a method to improve the client performance in Tor by a modified method of middle node selection that is based on latencies between routers. The author proposes to use an approximation technique that is based on measuring latencies between responsible DNS servers [5]. The quality of the latter is questionable [6]. Thus, the practical acquisition of router-to-router latencies, which is of capital importance, is left to future work.

Snader and Borisov [7] 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 all other nodes recently experienced to the given one. According to their own measurements though, the opportunistic bandwidth estimation is less accurate than using self-advertised values from the descriptors. While they estimate the bandwidth of nodes and require data aggregation from all ORs, we split it up to the links between the nodes. This kind of estimation is even possible with a single node. Additionally, besides bandwidth, we are interested in measuring RTTs and improving latencies.

Our contribution [6] studies the performance of Tor under various circumstances in order to detect bottlenecks, as well as to learn about the overall situational behavior of the network and limits of nodes regarding performance metrics like latency and throughput. Further, it shows the influence of geographical diversity of routers in paths on the performance of circuits. Additionally, we studied the correlation between circuit setup time, RTT and throughput.

Murdoch et al. [8] explore the effectiveness of path compromise with regard to the Tor's default path selection algorithm as well as to the methods proposed in [7]. The main finding is that in the presence of a node-rich but bandwidth-limited attacker Tor's default path selection algorithm can offer an improved protection compared to the uniform path selection algorithm.

IV. Methods and Evaluation

This section proposes and evaluates new methods of measuring performance in the Tor network and choosing paths based on the results of such measurements. The proposed measurements can also be used to verify information that is self-advertised by the routers, and therefore increase security. To this end, we consider two approaches that are based on orthogonal performance metrics: RTT- and throughput-based path selection methods. We further compare these to GeoIP-based approaches that were suggested in [6]. The proposed methods can be used to increase the performance and security for end-users of the Tor network, but may also serve as input for improved designs of future anonymity systems.

A. Strategy I – Actively Measuring RTTs

The Tor protocol does currently not provide any mechanisms to measure latencies of the provided anonymous channels in terms of round-trip times (RTTs). Latency information about links in a virtual overlay network like Tor can be used as a routing metric for path selection. Links having lower latencies can be preferred over others, in order to achieve an improved performance.

Our implementation measures RTTs of circuits by violating the exit policy of the last router in an arbitrary path. This is done by sending a *relay connect* cell on a circuit that is to be tested, using *localhost* as a dummy-destination. Since the exit policies of all routers deny connections to *localhost*, these attempts result in errors that can be timed in a measuring client. Making use of the *leaky-pipe* circuit topology that is used in the design of Tor [1], it is further possible to extend this technique for measuring RTTs of partial circuits. These can be used to calculate link-wise RTTs between the single routers on any path. By diversifying the number of encryption layers when initiating a request, every single hop on a circuit can specifically be addressed as the targeted node. After performing measurements addressing every hop in a circuit once, the RTT of the link between hop $n-1$ and n can be estimated using the following equation:

$$RTT_{n-1,n} = RTT_{0,n} - RTT_{0,n-1}$$

For making use of the measured results, it is proposed to model the explored subnet of the Tor network as a graph structure. Such a *network model* contains nodes, links between these nodes and arbitrary *node-* or *link-wise* performance metrics. In our analysis we measured the performance of links between single nodes and considered complete paths that were combined from single links.

It is necessary to keep the random aspect in path selection for preserving anonymity. Our implementation chooses paths probabilistically from the current set of all path proposals regarding ranking indices. Practically this is done by sorting the list of path proposals for their rankings in descending order and choosing a random number x between 0 and the

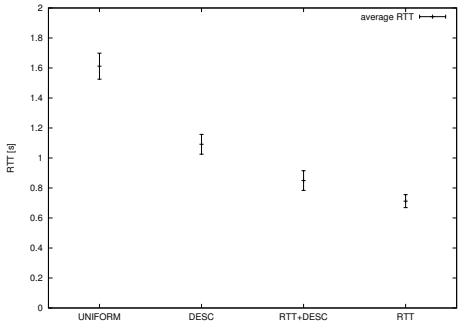

Figure 1. Comparison: RTT

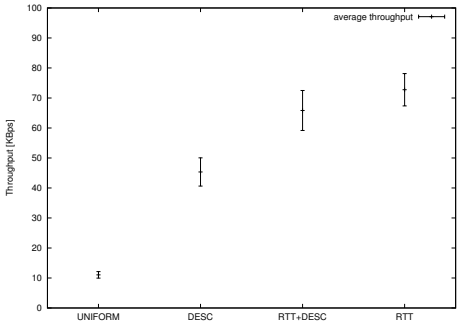

Figure 2. Comparison: Throughput

total sum of all rankings. A proposal is selected by iterating through the sorted list, and subtracting the ranking indices r_i of the single list elements i from x ($x := x - r_i$), until x is smaller than 0. The ith list element, whose index r_i was subtracted last, is the chosen path.

For the evaluation, we compare different methods of path selection with regard to the average performance of generated paths, in terms of latency and throughput of the resulting circuits. Throughout this work we refer to a uniform selection of the nodes in a path by the name *UNIFORM*, while we denote the currently used probabilistic path selection method that is based on information from descriptors only by *DESC*. In addition to these, two methods that choose paths based on measured RTTs of single links were tested: *RTT* and *RTT+DESC*. *RTT* selects paths probabilistically from a network model by weighting the available proposals with the expected added up RTTs of the complete paths only (sum over all links included). Thus, the lower the expected (measured) RTT of a proposal, the more likely it is that the proposal is chosen. *RTT+DESC* chooses paths based on a ranking index that is combined from both the expected total RTT of a path, as well as the minimum self-advertised bandwidth of the three routers included in a path. This is done by giving out two different scores (RTT- and bandwidth-score) to each path. To determine a path's RTT-score, the list of proposals is sorted by the RTTs of the paths in descending order. The position of a path in the resulting list states its score (for bandwidth-score the list is sorted by bandwidth in ascending order respectively). The ranking index r_p of a single path p is then determined using the following equation, where in our experiments a and b were both configured to 1 for having both metrics considered to the same proportion:

$$r_p = (a \cdot p.rttscore) + (b \cdot p.bwscore)$$

Figures 1 and 2 show the mean RTT and throughput of 600 circuits (including 95% confidence intervals), where respective paths were chosen using each of the single methods. Every circuit's RTT was therefore measured 5 times, while the mean of these 5 values was integrated into the final result. Throughput measurements on circuits consisted of a single TCP stream (512 KB) being requested from the measuring host itself and transferred using the tested circuit. In this experiment, choosing routers based on measured latencies (*RTT*) delivered the best performance on average regarding all of the considered categories. Table I additionally shows the median values, as well as GeoIP-based methods (see [6]) as a reference: *GEO-IP* (DE) chooses uniformly from nodes that are located in Germany only, while *GEO-IP* (EU) chooses a first hop in Germany and the other hops uniformly from distinct european countries. The comparison shows, that even *DESC* delivers a better performance than improvements that could be achieved by performing path selection based on geographical data only.

	Setup[s] (*median/mean*)	RTT[s] (*median/mean*)	Throughput[KBps] (*median/mean*)
UNIFORM	4.48/6.31	1.34/1.61	7.17/11.04
GEO-IP (EU)	4.22/5.49	1.23/1.52	6.94/9.52
GEO-IP (DE)	3.27/4.48	1.12/1.36	9.45/17.69
DESC	2.11/3.33	0.86/1.09	19.68/45.33
RTT+DESC	1.35/2.46	0.57/0.85	34.23/65.83
RTT	1.10/1.91	0.48/0.71	36.67/72.75

Table I
MEDIAN/MEAN PERFORMANCE OF DIFFERENT METHODS

Thus, we have seen that choosing paths based on actively measured RTTs can improve the average performance of Tor anonymization channels significantly. It is interesting that circuits created using RTT-based methods do not only provide the lowest average latency, but also the highest throughput, even compared to using self-advertised bandwidth metrics.

Please note that the ability to measure latencies of complete circuits further enables us to optimize load-balancing in the Tor network on the circuit layer. This can be done by ensuring within the clients that all user streams are attached to circuits that are currently having a low latency within a pool of established circuits. Our implementation therefore maintains a list of n established circuits at any time and measures their RTTs in regular time intervals. This list is

sorted after every measurement by the RTTs of circuits, while incoming streams are always attached to the circuit that is currently having the lowest latency (the evaluation in Section V includes this feature).

B. Strategy II – Passively Estimating Available Bandwidth

Depending on the specific application, a constantly high throughput rate might be more important to a user than latency. Therefore, throughput also needs to be considered as a metric for path selection. Instead of choosing paths based on self-advertised node-wise bandwidth values from the directory, it is proposed to measure throughput link-wise, for being able to more precisely predict the capacities of specific nodes or links.

Measuring throughput actively by transferring streams over certain nodes to probe their capacities is definitely too much overhead that would have negative impacts on the overall network performance. Therefore we propose to measure throughput passively from within the nodes, but consider single TLS links to other routers on the network, instead of counting only the total amount of traffic a node is relaying. Such measurings can be conducted without producing any additional load on the network, but rather by counting user traffic that is already being transferred via the links. From these measurings, a status document can be generated in regular intervals, containing estimations of the currently available bandwidth on the single TLS links to all endpoints a router is connected to. The length of the interval states a trade-off between the quality of the data and the quality of protection. The smaller the interval, the more current the data. Small intervals, however, leak more information about current streams a node relays. We propose that a single router estimates $AVAIL_{\mathsf{L_i}}$, the bandwidth that is currently available on the TLS link $\mathsf{L_i}$ to another router i in the network, using the following equation:

$$AVAIL_{\mathsf{L_i}} = min((MAX - CURR) + \frac{CURR}{\#CIRCS + 1}, \\ (MAX_{\mathsf{L_i}} - CURR_{\mathsf{L_i}}) + \frac{CURR_{\mathsf{L_i}}}{\#CIRCS_{\mathsf{L_i}} + 1}).$$

MAX and *CURR* (respectively $MAX_{\mathsf{L_i}}$ and $CURR_{\mathsf{L_i}}$) denote the maximum and current observed bandwidth within a recent period of time, considering all channels (respectively a TLS link $\mathsf{L_i}$ only). $\#CIRCS$ represents the overall number of currently existing circuits that include the measuring node, and $\#CIRCS_{\mathsf{L_i}}$ the number of circuits that include TLS link $\mathsf{L_i}$ to router i. The estimated available link-wise capacity $AVAIL_{\mathsf{L_i}}$ is therefore the minimum of the overall node capacity and link capacity of link $\mathsf{L_i}$. Each estimation is calculated as the difference between the maximum and the currently observed throughput, plus an additional fraction of the currently used capacity a circuit would get, if fair scheduling was applied.

For the evaluation, a single Tor router connected to the public Tor network was used, with an additional controller performing the estimations and communicating them to clients upon request. A comprehensive evaluation would require instances of this controller at every OR in the Tor network. For comparing the performance of paths that were chosen based on bandwidth information (referred to as *BW-INFO*) to those that were created using *DESC* and *RTT*, all of the test circuits were using the measuring router as their second hop, while entries and exits were chosen probabilistically based on the respective metrics. The results are shown in Table II. Every field contains the median and mean value of 300 circuits, while every circuit's performance was measured 5 times to compute a mean.

	Setup[s] (*median/mean*)	RTT[s] (*median/mean*)	Throughput[KBps] (*median/mean*)
DESC	1.07/1.86	0.49/0.71	68.17/144.00
RTT	0.57/1.74	0.26/0.54	37.37/110.42
BW-INFO	1.09/2.27	0.52/0.78	124.84/165.71

Table II
COMPARISON OF PATH SELECTION METHODS

These results were achieved after running the Tor server and controller for longer than a week, for collecting data from counting traffic in order to have estimates for almost all links. Since the machine that was used as the middle node of the evaluated circuits has a fast connection to the Internet and enough computing capacities, the reached average throughput values are comparatively high. Though, using the current implementation, significant improvements regarding the average throughput could be achieved in comparison to other methods.

V. COMPARISON FROM THE END-USER'S PERSPECTIVE

Privacy-friendly web browsing is one of the most popular use cases for anonymizing networks. In order to learn the impact of Tor anonymization and the proposed improvements on web browsing performance that is perceived by end-users, we measured the average time needed to fetch single HTTP-headers of 100 of the most popular websites[1]. For the experiment, only their headers were downloaded using the HTTP HEAD-method that asks for a response similarly to the one that corresponds to a GET request, but without the HTTP content body. Table III shows the averaged median and mean values of the time needed to fetch a single header, as well as the averaged standard deviation and min/max values. The test was configured to run 100 times with a pause of 10 minutes between the measuring rounds. The first row contains results from using the default Tor implementation, while the second row shows results from performing the same test using Tor including the proposed improvements.

[1]The used list contains 50 URLs from Germany's and 50 from the USA's traffic ranking regarding *http://www.alexa.com*

The third row is given for reference and contains values that were measured without anonymization.

	Median[s]	Mean[s]	Stddev[s]	Min/Max[s]
Default Tor	3.35	4.04	3.18	1.34/23.33
Optimized Tor	0.75	1.35	2.37	0.25/17.44
No Anonymization	0.26	0.39	1.25	0.01/11.37

Table III
TIME NEEDED FOR FETCHING HTTP HEADERS

In this evaluation, RTT-based path selection was used, as well as optimized load-balancing on the circuit layer.

VI. 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. Ideally, improved methods would offer no possibilities for attackers to influence the path selection of clients at all. Any such implications of the used methods have to be studied and a reasonable trade-off between the quality of protection and quality of service has to be chosen before any improved methods will become mature for real life usage.

In this work we will consider a user to be *deanonymized*, if an attacker owns the entry and the exit node of the user's circuit [1]. The anonymity analysis shall therefore be based on the following question: How easy is it for an attacker to operate the first and the last hop in as many paths as possible? From our point of view this is the most serious attack in this context (*end-to-end timing*, respectively *volume correlation*). Furthermore, when using a self-operated OR as the entry of all circuits in combination with dynamic path lengths, even a first and a last node belonging to the attacker is not a problem, if caution w.r.t. timing attack is taken [9]. Therefore, this analysis considers the case of a user operating an OP only. Using a self-operated OR increases protection compared to the case where the user only operates an OP.

For a first estimation, a sample of 10,000 paths was generated using several different path selection configurations, while measuring the total number of distinct nodes that were chosen on the single positions, as well as the total number of distinct combinations of specific entry and exit nodes (*EE-combinations*) that occurred in the generated paths. Table IV shows the numbers of distinct nodes on each position, while the number of distinct EE-combinations is given by the last column. The row containing *DESC* (*GEO-IP*) includes results from using an exemplary algorithm based on *DESC* that makes use of *GEO-IP* to enforce distinct countries for each hop, while the entry node was always chosen to be in Germany. The configuration did not allow ocean crossings, while continent crossings between Europe, Africa and Asia were allowed. Three simulations were done using *RTT*, without specifying a threshold at all, as well as specifying 1.0 and 0.5 seconds as maximum RTT of a path. Path selection with these methods was performed out of totally 7834, respectively 3966 and 2325 path proposals.

	#Entries	#Middles	#Exits	#EE combinations
UNIFORM	782	782	651	9912
UNIFORM (GUARDS)	411	782	651	9824
DESC	825	832	598	7629
DESC (GUARDS)	459	853	611	7102
DESC (*GEO-IP*)	116	167	182	4489
RTT	150	121	122	3511
RTT (1.0)	113	109	100	2439
RTT (0.5)	93	82	81	1632

Table IV
NUMBERS OF DISTINCT NODES AND EE-COMBINATIONS

For being able to specify a degree of anonymity for methods that choose routers probabilistically it is necessary to take into account that some nodes are chosen more often than others. For a better approximation, another 100,000 paths were generated using different methods, to calculate entropies $E(X)$ regarding empirically measured probabilities of occurring EE-combinations, as well as a respective bounded degree of anonymity d from a sample of paths X. This is done in analogy to [10], 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 respective probability of the combination i to be chosen. The maximum entropy E_{Max} that 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 every EE-combination with the same probability than any other. Table V shows the number of distinct combinations that occurred, the entropy, and the degree of anonymity of 4 different methods of path selection.

These results show, that – regarding the considered metric – *UNIFORM* nearly reached perfect protection, while *RTT* with the used model (containing 505 nodes, 1667 edges) provided comparably restricted protection. Figure 3 further shows the percentage of sample paths that included an EE-combination with a certain probability, where the x-axis is shown in a logarithmic scale. Most probably, the anonymity

	N	$E(X)$	d
UNIFORM	86160	16.32	0.98
DESC	32188	13.86	0.83
RTT	5951	11.94	0.72
RTT (0.5)	1790	10.39	0.63

Table V
ENTROPIES AND DEGREES OF ANONYMITY

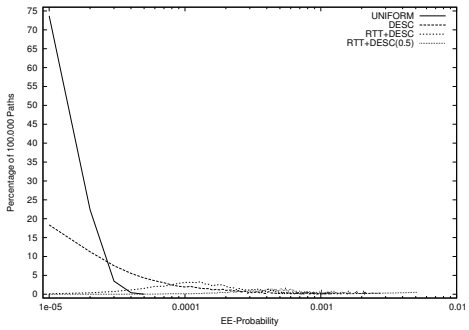

Figure 3. Percentage of Paths and Probabilities of EE-Combinations

properties of *DESC* can eventually be reached or even topped, when choosing paths based on measured RTTs from a complete model of the Tor network.

Finally, Figures 4 and 5 show the negative correlation of the average performance of circuits and the corresponding degree of anonymity. The plots show 4 different methods of path selection, while the respective performance axis (throughput/RTT) reflects the performance that was reached on average by circuits created with a specific method.

VII. DISCUSSION

For practical adoption of the proposed methods in any existing overlay network, many important aspects will need to be considered first. The variety of different design decisions as well as the potential influence on the effectiveness of known attacks shall be demonstrated in this section. Initially, it has to be decided which specific metric to use, which implies whether to use node-wise or link-wise metrics. RTTs are generally a link-wise metric, while throughput may either be measured node- or link-wise. Note that passively measured link-wise throughput (as described in Section IV-B) can be converted to a node-wise metric, by averaging over collected values in a centralized instance (e.g. using the median in order to reduce the influence of malicious nodes). This is potentially useful for detecting colluded nodes or even groups of colluded nodes reporting wrong capacity informations. While any link-wise metrics will allow to more precisely predict the performance of certain paths, the amount of data that needs to be measured, stored in a model and communicated to the clients is quadratically increasing with the number of routers in the network. If a complete network model shall be used though, this could be provided by a trusted centralized directory, in order to be

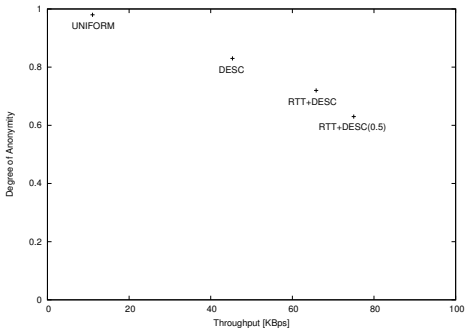

Figure 4. Anonymity vs. Throughput

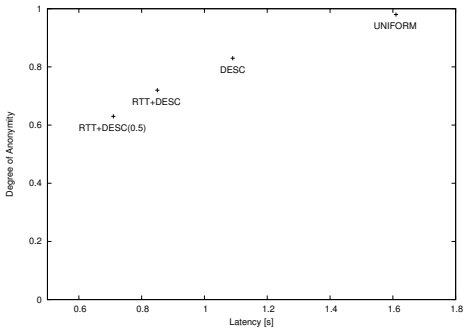

Figure 5. Anonymity vs. Latency

downloaded by clients in a compressed format. Described measurements of RTTs can either be done from within the clients, respectively routers, or one or more centralized *network monitors* that create circuits with the purpose of measuring the network only. While the first possibility puts more additional load on the network, the latter suffers from the fact that there is a single point of failure and trust. By performing measurements of RTTs in the clients only, without sharing a network model among the participants, every client would discover its own performance-improved part of the network. However, anonymity implications of such an approach still have to be analyzed.

It is also possible to allow users to specify their desired performance-anonymity trade-off, depending on concrete needs. This can be done as follows: An updated ranking r_u can be derived from the previously calculated one r as follows: $r_u = \alpha + r \cdot (1 - \alpha)$, where $\alpha \in [0, 1]$ is a parameter that indicates the performance-anonymity trade-off. $\alpha = 1$ will lead to a completely uniform selection, while $\alpha = 0$ will provide weighted probabilistic selection as it was described before.

While performing active measurements (in our proposal these are RTTs), it has to be taken into account that if measurement probes can be detected by routers, this can be misused by malicious nodes in order to allocate more resources to these probes, and thus get higher rankings than deserved. After the initial network model is created,

it is proposed to update performance parameters with the measured/estimated ones during operational mode in an exponential weighted moving average way. Thus, not only the periodic probes, but also actually achieved performance is considered in the metric.

Concerning performance metrics itself, it is possible to consider other ones like jitter or queue sizes as well. From the authors' point of view, however, the proposed metrics – RTT and throughput – are on the one hand more important ones, and on the other hand significantly easier to obtain. Our final proposal is to use RTT as the primary metric for performance-improved path selection in Tor. It can be obtained by clients itself and thus neither requires to trust any other parties nor needs any complicated mechanism for data aggregation. Further, RTT-based metric not only significantly improves latency – the most important metric as perceived by the users [11] – but bandwidth as well. Even though not as accurate as our bandwidth estimation method, the bandwidth achieved by the proposed RTT-based path selection method still outperforms the most precise prior studied node-wise bandwidth metric *DESC* (c.f. [7]) by the factor 1.6 (c.f. Figure 2).

Finally, one could wonder whether providing better load-balancing and utilization of Tor resources will really improve the performance as perceived by end-users. When the proposed improvements to path selection are deployed by many Tor users, an increase in performance will probably not be as great as the data in our evaluation suggests. As the matter of fact, the performance of the anonymity network is the slowest tolerable speed by its users[2]. Therefore, it is a naïve hypothesis that e.g. doubling the available bandwidth would lead to a state where users would experience twice the throughput as before. The point is that this is not true because the number of users varies with the available bandwidth. The same applies for latency. Even if the performance would not be improved drastically, better load balancing and more efficient resource utilization will at least lead to a significant growth of the user base, which will contribute to improve quality of protection.

VIII. Conclusions

In this paper we proposed new methods of measuring performance in anonymizing overlay networks like the Tor network, while performing path selection based on the results of these measurements. The specific metrics are in fact actively measured round-trip times and estimated available bandwidth capacities using passive observations of link-wise throughput. As our evaluations show, the proposals can be used to significantly improve the average performance that is achieved by end-users of Tor. Further, the security of the system regarding certain attacks is increased, when self-advertised performance values are replaced by the remotely measurable ones.

We additionally presented an approach for practically estimating the strength of anonymity that is provided by different methods of path selection in comparison to each other. This can be used to prove and document any possible loss or gain of anonymity that is induced by modifications to the method that is currently applied. Finally, users can select their own trade-off between anonymity and performance depending on their concrete requirements and thus select the corresponding method of path selection.

[2]http://www.lightbluetouchpaper.org/2007/07/18/economics-of-tor-performance/

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium*, 2004.

[2] "Tor Node Status," https://torstat.xenobite.eu/.

[3] R. Dingledine and N. Mathewson, "Tor Path Specification," https://www.torproject.org/svn/trunk/doc/spec/path-spec.txt.

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

[5] 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. [Online]. Available: citeseer.ist.psu.edu/gummadi02king.html

[6] 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.

[7] 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.

[8] 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)*, N. Borisov and I. Goldberg, Eds. Leuven, Belgium: Springer, July 2008, pp. 115–132.

[9] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How Much Anonymity does Network Latency Leak?" in *Proceedings of the 14th ACM Conference on Computer and Communications Security (ACM CCS)*, Alexandria, Virginia, USA, October 2007.

[10] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards Measuring Anonymity," in *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, R. Dingledine and P. Syverson, Eds. Springer-Verlag, LNCS 2482, April 2002.

[11] S. Köpsell, "Low Latency Anonymous Communication - How Long Are Users Willing to Wait?" in *ETRICS*, ser. Lecture Notes in Computer Science, G. Müller, Ed., vol. 3995. Springer, 2006, pp. 221–237.