Toward Improving Path Selection in Tor

Fallon Chen
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404
Email: ftchen@cs.ucsd.edu

Joseph Pasquale
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404
Email: pasquale@cs.ucsd.edu

***Abstract*—Tor (The Onion Router) is a popular anonymity overlay network that seeks to provide anonymity without significant cost to performance. Tor's support for anonymity is indeed strong, but its network performance is a problem, and one that is widely recognized. While there are some studies that investigate changing the structure of the Tor network to improve performance, we focus on investigating different path selection strategies given the Tor network as is. Specifically, we explore varying the number of hops in a circuit, varying the performance flags in a circuit, and varying the geographic distance between routers in a circuit. We show how much improvement can be had by reducing the path length, which gives the user guidance on how to trade off anonymity for performance.**

I. Introduction

Tor [1], also known as The Onion Router, is one of the most popular tools for anonymous communication over the Internet available today. Its maintainers designed Tor to strongly protect the anonymity of the Tor network user for the purposes of privacy and censorship resistance. At the same time, the developers strive to achieve good performance in order to encourage use of the network. Although Tor is relatively successful at providing anonymity, improving the performance of the network has been difficult due to the nature of its design.

Our approach to improving the performance of the Tor network was to study the impact of different Tor path selection strategies on bandwidth, circuit failure rates, and the number of attempts required to build a circuit. The benefit of studying such strategies is that they enable the evaluation of features of the current network as well as potentially providing an easily deployable solution for improving performance.

Since the Tor routers are run by people volunteering processer time and storage space on their personal computers, most information about their properties are out of reach. However, their IP addresses, and the performance ratings assigned to them, are readily available. In addition, it is possible to control the number of routers in a circuit.

Given this available information, we explored the effects of three strategies that respectively seek to take advantage of: (1) geographic proximity; (2) performance ratings; (3) the number of routers in a circuit. The performance and reliability measurements that we obtained gave us information about the accuracy of the performance ratings, the impact of geographic distance, and the effect of varying the number of hops.

The main results we show in this paper are that geographic distance has a negative impact on throughput and reliability, that the performance ratings in Tor are accurate, and that reducing the number of hops in a circuit improves throughput by 1-3 Kbps per hop removed. The second section covers Tor background, the third our experiments, the fourth our results, and the last our conclusions.

II. Background

A. Tor

Tor (The Onion Router) is an overlay network that provides anonymity for applications built on TCP, and is a real world implementation of onion routing [2]. Its goal is to provide low latency anonymity for its users. It achieves anonymity by setting up a path in the network, known as a *Tor circuit* that is comprised of *Tor routers*, one per hop. A key aspect of supporting anonymity is that each router in the circuit only knows of its predecessor and successor in the circuit. Once a circuit is established, multiple TCP streams can use it to send data anonymously over the circuit.

Data is encrypted in layers, which are unwrapped by each router in the circuit using a symmetric key, and then relayed to the next router. When it arrives at its destination, it is completely stripped of onion routing encryption. When the destination sends information back along the circuit, layers of encryption are added on at each router in the circuit. This makes it difficult for those who intercept data at a compromised router to get the plaintext data, since the key for the layer that gets unwrapped at the next hop does not exist at the compromised router.

Volunteers can specify through their exit policies how much bandwidth to provide, which ports and addresses are or are not allowed, and whether or not the router can be used as an exit. There are also are several other router classifications that give their stability, bandwidth, and ability to respond to certain protocols.

The standard number of routers in a circuit is three, which is a number Tor's designers determined was a good compromise for providing both anonymity and good performance [1]. Tor currently chooses routers randomly with probability weighted by bandwidth [3].

Finally, a *Tor stream* is an encrypted TCP stream, formatted so that it can be passed through the Tor network. Multiple streams can be attached to a single circuit.

B. Tor Performance Problems and their Impact on Anonymity

Network throughput over the Tor network is orders of magnitude slower than throughput over a normal Internet connection. This contributes to poor usability, which hinders the adoption of Tor, resulting in fewer Tor users and fewer volunteers to run Tor routers. Fewer Tor users means that it is easier for attackers to confirm that traffic patterns belong to a specific user, since there is less of a crowd for the user's traffic in which to hide. Fewer routers means that is easier for attackers to compromise circuits and gain a wide view of the network, since they can more easily add enough routers to control a significant subset of all routers. Also, poor usability prevents the use of high bandwidth applications over Tor, which further limits use of Tor since it does not provide the services needed. For these reasons, it is important to understand the performance of Tor.

III. Experiments

A. Choosing Metrics for Performance and Reliability

We chose throughput as the representative metric for performance rather than latency for the following reasons. First, most users of Tor use it with applications that require reasonable throughput for transferring large files. Secondly, measuring throughput is an integral part of how Tor builds faster circuits through its network, which shows that it is a useful metric when considering the performance of Tor. We also noticed in preliminary measurements that latency, measured as the time between sending the first byte of a request and receiving the first byte, had high variability. Its standard deviation was nearly three times the average. This made it less suitable for use in evaluating different path selection strategies.

We measured throughput by measuring the time required to download a 100 kilobyte file over a route in the Tor network. We did this for approximately one hundred unique routes, downloading the file ten times over each route. We repeated this throughput measurement at several locations around the world, to be sure that the throughput measured is not responding to unique conditions in a particular region.

We chose to look at two metrics for reliability. The first is the average of the number of attempted paths tried before a circuit that could be successfully built was found. The number of attempted paths was chosen because circuits frequently fail while being built. These failures make it difficult for users to switch circuits quickly for the purposes of security. If a circuit has not yet been prebuilt, a higher number of attempts required also means a longer wait is required. The second measure of reliability is the percentage of successfully built circuits that failed, either during a download or while trying to connect to the server. This metric helps us evaluate whether or not a given strategy is useful. If a user constantly has to build new circuits, they will not want to use the strategy.

B. Implementation

To carry out our measurements, we set up ten Tor clients on PlanetLab [4], [5] nodes distributed around the world. We hosted the 100 kilobyte file at a server in La Jolla, CA (USA) using thttpd [6]. Downloads were conducted using cURL [7].

The measurement scripts were written in Python [8], so that it would be easy to interface with the Tor control port using the TorCtl library. The Tor control port can be used to tell the Tor client to generate circuits with whatever routers the user chooses, as well as providing network status information on which routers are available, their recent bandwidth measurements, and other characteristics.

We used a statically compiled version of the Tor client for testing [9]. Tor circuits are usually replaced after ten minutes of use; we changed this to half an hour so that a circuit would not be replaced in the middle of testing its performance. Geographic performance tests were all done on version 0.2.0.34, while the other performance tests were done on version 0.2.35. Version 0.2.0.35 fixes a memory leak in descriptor caches that existed in version 0.2.0.34 (we did not run into this memory leak). These versions were the latest versions of the Tor binary available at the time of testing.

IV. Path Selection Strategies

A. Varying Distance Between Hops

Varying geographic distance is a way of looking at the impact of geographic diversity on performance, as an increased number of miles between routers generally means more diversity in location. The general idea is that anonymity is improved by ensuring that organizations such as governments and ISPs are not able to observe both the entry and exit nodes of a circuit. Our experiments examined the impact of geographic diversity on performance to understand the feasibility of implementing path selection algorithms in Tor that ensure geographic diversity.

Our experiments built circuits with a specified distance between each router in the circuit and measured their performance using functions from the TorCtl library. Routers were selected using uniform random selection, with "Running" as the only router status flag required. The distance between each Tor router in the circuit had to be within the specified range.

Geographic distance was calculated using the great circle distance formula on geographic coordinates of each node, as provided by the MaxMind GeoLite City Database[10]. The ranges measured between hops were 400 ± 200 miles, 4000 ± 200 miles, 8000 ± 200 miles and 10000 ± 200 miles.

B. Varying Performance Flags

We chose to examine the impact of varying performance flags because if they are accurate, it should be a simple way to improve performance, and if they are inaccurate, then it would be worthwhile to reconsider the methods being used to classify routers with specific flags. Tor routers are given different flags by directory servers. The flags that relate to performance are "stable", and "fast". A router having a "stable" flag indicates that its uptime is above the known median for valid, running routers and that it is not running versions of the software that are known to erroneously drop circuits. A router having a "fast" flag indicates that its "observed bandwidth" (as defined by Tor) is in the top seven-eighths for known, running, valid routers or at least 100 kilobytes per second [11].

Observed bandwidth is actually an estimate. The server observes the maximum throughput sustained output over any ten second period in the past day, as well as the maximum throughput sustained input over any ten second period in the past day. The observed bandwidth is the lesser of the two measurements [11]. To examine the impact of Tor directory flags on performance, we created circuits with stable but no fast flag, fast but no stable flag, both stable and fast, and neither flags. Routers were randomly selected from the pool of routers that contain the appropriate flags.

C. *Variable Number of Hops*

This experiment determines the increase in throughput that comes with decreasing the number of hops, and thus routers, in a circuit. This experiment was also chosen because of the simplicity of its approach to improving performance. A vast majority of routers in the Tor network do not currently accept single hop circuits, so the experiment focused on circuits with two, three, and four hops instead. Only the “running” flag was specified, and routers were chosen using uniform random selection.

V. Results

A. *Varying Geographic Distance*

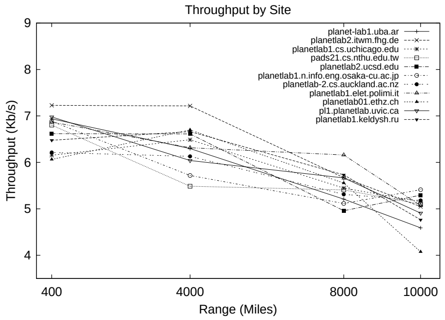

Fig. 1. Average throughput while restricting geographic distance between routers.

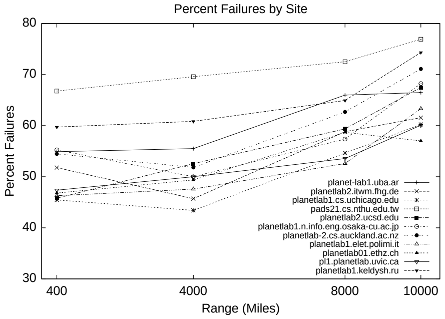

Fig. 2. Average percentage of failed circuits while restricting geographic distance between routers.

Figures 1, 2, and 3 show that both performance and reliability are about the same for the ranges 400 ± 200 and 4000 ± 200, at about 6.5 Kbps, with 53% failed circuits and about 2 attempted circuits before a successful one was found. They get worse for range 8000 ± 200, with throughput decreasing by about 1Kbps, percent failed circuits increasing by about 7% and number of attempted circuits increasing from about 2 to about 5. This gets even worse for range 10000 ± 200, with throughput decreasing by another 0.5 Kbps, percent failed circuits increasing by another 7% and average number of attempted circuits increasing from about 5 to about 25.

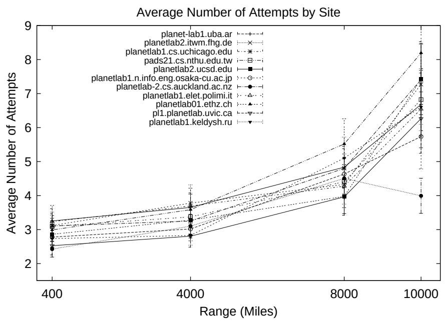

Fig. 3. Average number of attempted circuits while restricting geographic distance between routers.

In addition, the number of routers with at least one neighbor in the range was greatest for the range 400 ± 200 with 1517 routers, followed by the range 4000 ± 200 with 1432 routers. This suggests that better anonymity is provided by these ranges as well.

B. *Varying Performance Flags*

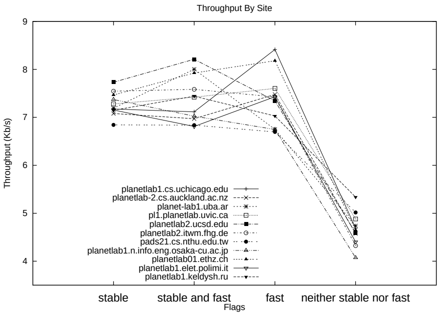

Fig. 4. Average throughput while varying performance flags.

Figures 4, 5, and 6 show that the best performance was found in circuits with both stable and fast flags, though it was very close to the performance of circuits with stable flags only or the fast flag only. Performance got significantly worse in circuits without both the stable flag and the fast flag, dropping from about 7Kbps to about 4Kbps. The recommendation for improving performance from these experiments is to have at least one of those flags set, but especially the fast flag since there are so many more routers that have only the fast flag than only the stable flag.

The most reliability came from circuits with stable flags only at about 40% failed circuits and an average of about 2 attempted circuits, but circuits with both stable and fast flags

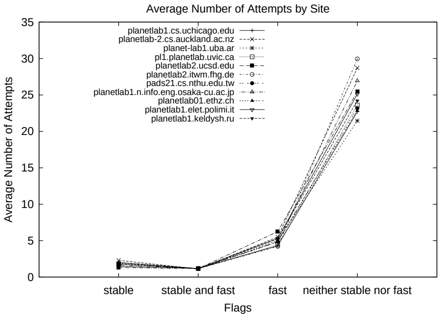

Fig. 5. Average number of attempted circuits while varying performance flags.

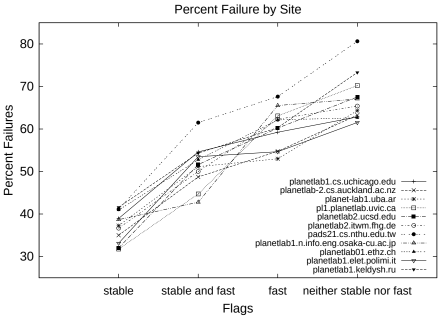

Fig. 6. Average percentage of failed circuits while varying performance flags.

were a very close second, at about 52% failed circuits and an average of about 1 attempted circuit. Since there are only 23 routers with only stable flags, it makes sense from anonymity, performance and reliability standpoints to use routers with both flags.

C. Varying Number of Hops

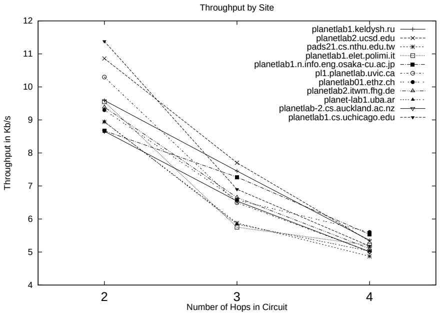

Fig. 7. Average throughput while varying number of hops.

Figures 7, 8, and 9 show that given a uniform random selection of routers (one per hop), performance improves with decreasing number of hops. The expected throughput for two hops is between 8 Kbps and 12 Kbps, the throughput for three hops is between 6 Kbps and 5 Kbps, and the throughput for four hops is between 4Kbps and 6 Kbps. With less certainty, we also find that reliability improves with a decreasing number of hops. Anonymity decreases with the number of hops, but this problem could be mitigated by the use of guard nodes.

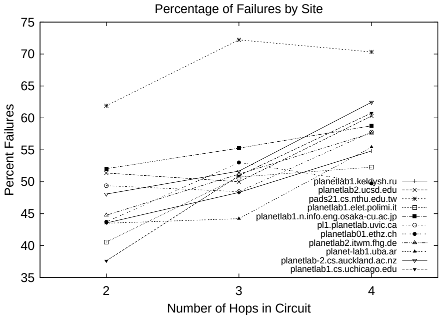

Fig. 8. Average percentage of failed circuits while varying number of hops.

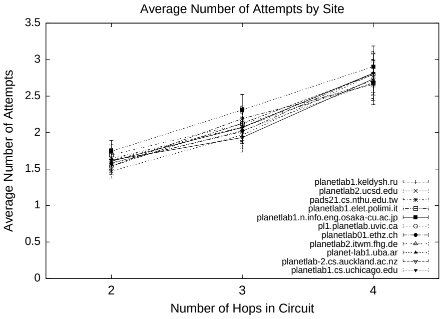

Fig. 9. Average number of attempted circuits while varying number of hops.

VI. Comparison of All Path Selection Strategies

Table I provides the throughputs averaged over all sites for each path selection strategy. The Bandwidth Weighted strategy is the one that the Tor client uses to choose fast circuits. Measurements for it were taken at each site using the same scripts as the rest of the path selection strategies. The Bandwidth Weighted algorithm still has the best performance, at 12.7 Kbps. The path selection strategy that the Tor client uses for regular circuits is one of uniform random selection. This is the same as the Three Hop Uniform path selection strategy, and has about half the throughput of the bandwidth weighted algorithm.

It is surprising that the two hop uniform random selection is slower, and reinforces the idea that the quality of Tor routers makes the biggest difference. The worst performance was from the circuits that had uniformly selected circuits that had no fast or stable flags, though the four hop circuits and the circuits that had between 10000 ± 200 miles between them were close.

Table II shows that bandwidth weighted and circuits with both stable and fast flags have the least number of attempts before establishing a successful circuit. Circuits with neither fast nor stable flags have the most number of attempts, which is almost four times higher than the next highest number of attempts.

Bandwidth Weighted has a surprisingly high percentage of failure at 66%, third highest after circuits with no fast or stable flags, and 10000 ± 200 miles between hops. This can be seen in Table III. Stable flags only has the lowest percentage, followed by two hop uniform and 400 ± 200 miles between hops. However, even the lowest failure rate is at 37%.

TABLE I
THROUGHPUTS AVERAGED OVER ALL SITES OF ALL PATH SELECTION STRATEGIES FOR VARYING PERFORMANCE FLAGS

Path Selection Strategy	Throughput (Kbps)
Bandwidth Weighted	12.7 ± 0.5
Two Hop Uniform	9.6 ± 0.5
Stable and Fast Flags	7.4 ± 0.3
Fast Flag Only	7.4 ± 0.3
Stable Flag Only	7.3 ± 0.1
Three Hop Uniform	6.7 ± 0.3
400 ± 200 Miles Between Hops	6.7 ± 0.3
4000 ± 200 Miles Between Hops	6.3 ± 0.2
8000 ± 200 Miles Between Hops	5.5 ± 0.2
Four Hop Uniform	5.2 ± 0.2
10000 ± 200 Miles Between Hops	5.0 ± 0.2
No Stable, No Fast	4.7 ± 0.1

TABLE II
ATTEMPTS BEFORE ESTABLISHING A CIRCUIT AVERAGED OVER ALL SITES

Path Selection Strategy	Average Number of Attempts
No Stable, No Fast	24.8 ± 1.5
10000 ± 200 Miles Between Hops	6.6 ± 0.3
Fast Flag Only	5.0 ± 0.2
8000 ± 200 Miles Between Hops	4.6 ± 0.2
4000 ± 200 Miles Between Hops	3.3 ± 0.1
400 ± 200 Miles Between Hops	2.9 ± 0.1
Four Hop Uniform	2.77 ± 0.08
Three Hop Uniform	2.09 ± 0.06
Stable Flag Only	1.81 ± 0.08
Two Hop Uniform	1.60 ± 0.04
Bandwidth Weighted	1.32 ± 0.02
Stable and Fast Flags	1.17 ± 0.02

TABLE III
PERCENTAGE OF FAILED CIRCUITS AVERAGED OVER ALL SITES

Path Selection Strategy	Average Percent of Failure
No Stable, No Fast	68 ± 3
10000 ± 200 Miles Between Hops	67 ± 5
Bandwidth Weighted	66 ± 2
8000 ± 200 Miles Between Hops	61 ± 3
Fast Flag Only	61 ± 3
Four Hop Uniform	58.9 ± 0.1
Three Hop Uniform	53.6 ± 0.2
4000 ± 200 Miles Between Hops	54 ± 4
400 ± 200 Miles Between Hops	53 ± 4
Stable and Fast Flags	52 ± 3
Two Hop Uniform	47.6 ± 0.2
Stable Flag Only	37 ± 2

Bandwidth Weighted path selection is clearly the leader in performance, as it provides the highest throughput as well as the speediest establishment of a circuit. For the strategies we considered, circuits with both stable and fast flags appear to perform the best and also have good reliability. Although circuits with only stable flags have better reliability, there are few of them, so circuits with both stable and fast flags are the better choice.

A. Impact on Anonymity

This study is focused on finding strategies for users that will accept the loss of some anonymity in exchange for better performance. A basic metric for anonymity is the size of the pool of routers available for making a circuit, though the length of the circuit has an impact as well. Having a smaller pool of routers results in less anonymity, since an attacker needs to compromise fewer nodes to discover information about the circuit. Thus, the three hop, uniform random path selection algorithm originally used by Tor provides the best anonymity out of the options examined in this study since it draws from the pool of all running routers. The rest reduce the anonymity provided by Tor to some extent, since they either alter the length of the path or reduce the size of the set of routers.

We first consider the impact of altering the length of the path first. Two hop circuits are the least anonymous of the types of circuits described in this paper, since all the attacks that apply to three hop circuits apply to two hop circuits, but can be done more quickly. One solution to this may be to use guard nodes, which are trusted nodes that Tor users repeatedly use as entry nodes [3], or to use stronger encryption over the Tor circuit. There are differing views on the impact of having a four hop circuit on anonymity. Borisov, Danezis, Mittal, and Tabriz [12] suggest that if a sufficient number of nodes in the Tor network are compromised, having a four hop circuit increases the chances of having more compromised nodes in the circuit, which makes it easier to discover information about the user. However, if the network has a small number of compromised nodes, the chances of having many compromised nodes in the circuit decreases, and anonymity increases as more work is required to identify the extra hop.

Both the geographic distance path selection strategy and the flag selection strategy reduce the size of the pool of routers available for use in path selection, so we looked at ways to get better anonymity within those strategies.

On September 21, 2009, each of the ranges used in the geographic distance path selection could use over two thirds of the available routers. Thus, this strategy is fairly safe from traffic analysis attacks that require router compromise or scanning information from the set of potential routers to work. The range with the least number of such routers is the 8000 ± 200 range, followed by the range 10000 ± 200. For the purposes of increasing the chances that the routers used are uncompromised, the ranges 400 ± 200 and 4000 ± 200 work best.

Feamster and Dingledine show in their paper, “Location Diversity in Anonymity Networks” [13] that country diversity, which is roughly approximated by this approach, is not sufficient to maintain anonymity in the face of an adversary with the ability to control ISPs. They suggest anonymous-system diversity instead as a way to fight this weakness. Murdoch and Zielinski [14] further show that internet exchanges can be used to monitor packets in Tor flows, even when anonymous-system diversity is assumed. Future work in considering geographic impact on performance should take these two studies into consideration.

On September 21, 2009, there were 23 routers with the stable flag but not the fast flag, 696 routers with the fast flag but not the stable flag, 754 routers with both flags, and 66 routers with neither. This suggests that one would gain better anonymity by choosing routers with both stable and fast flags, since it provides the biggest pool of routers from which to choose. This helps guard against traffic analysis attacks that require the attacker to have compromised at least one of the nodes or to check the latencies of every router potentially involved in the circuit, such as the congestion attack [15], [16], [17], [12] and intersection attacks [18], [19].

Another way to improve anonymity could be to use fast flag only exits, since there are more fast exits than there are stable and fast exits. Another option would be to choose only guard nodes as exits. Although there are only 113 guard nodes that are exits, these nodes are generally more trusted to be uncompromised, and the majority of them (105/113) are both stable and fast. This might also help avoid the attacks described by Bauer et. al that rely on malicious nodes misrepresenting their performance in order to encourage their use [17], which the approach of choosing specific flags is particularly vulnerable to. Bauer also suggests some strategies in his paper for better determining router bandwidth.

VII. Conclusion

We found that reducing the number of hops (and thus routers), reducing the geographic distance between routers and using the stable and fast flags helped improve performance and reliability. Reducing the number of hops from four to three improved performance by 1.2Kbps, and reducing the number of hops from three to two further improved performance by 3Kbps. Additionally, about 0.5 fewer attempts are necessary before finding a circuit with the removal of each hop, and 5% fewer circuits fail with the removal of each hop.

Reducing the geographic distance between routers improved throughput from about 5Kbps to 6.7Kbps, reduced the attempts required from 7 to 3, and reduced the percentage of failed circuits from 67% to 54%. Using the stable and fast flags improved throughput from 4Kbps to 7Kbps and reliability from 68% failure to 61% or less. In addition, the number of attempts before finding a successful circuit went from 25 attempts to fewer than 5.

Overall, the Bandwidth Weighted algorithm still had the best performance, while circuits with the stable flag only had the least number of failures, and the circuits with both stable and fast flags needed the least number of attempts before a good circuit was found. While anonymity might improve with the four hop circuit, the other strategies decreased the size of the pool of routers that an adversary would have attack to discover information about the user of the circuit.

In order to extend the work in improving performance, one would have to investigate the results of combining the strategies presented with choosing routers by weighting their bandwidths. To extend investigation into their impact on anonymity, it would first be necessary to investigate how the geographic distance algorithm performs when it takes internet exchanges into account, as well as investigating how much anonymity is lost when only two hops are available, and ways in which having four hops would benefit anonymity.

References

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

[2] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, “Anonymous connections and onion routing,” in *SP ’97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 1997, p. 44.

[3] R. Dingledine and N. Mathewson, “Tor path specification,” https://git.torproject.org/checkout/tor/master/doc/spec/path-spec.txt.

[4] P. L. Consortium, “Planetlab: An open platform for developing, deploying, and accessing planetary-scale services,” http://www.planet-lab.org.

[5] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir, “Experiences building planetlab,” in *OSDI ’06: Proceedings of the 7th symposium on Operating systems design and implementation*. Berkeley, CA, USA: USENIX Association, 2006, pp. 351–366. [Online]. Available: http://portal.acm.org/citation.cfm?id=1298455.1298489

[6] J. Poskanzer, “thttpd:tiny turbo throttling HTTP server,” http://acme.com/software/thttpd/.

[7] D. Stenberg, “curl website,” http://curl.haxx.se/.

[8] G. van Rossum, “Python website,” http://www.python.org/.

[9] T. T. Project, “Tor project website,” http://www.torproject.org/overview.

[10] M. G. C. Database, “Tor project website,” http://www.maxmind.com/app/geolitecity.

[11] R. Dingledine and N. Mathewson, “Tor directory protocol, version 3,” https://git.torproject.org/checkout/tor/master/doc/spec/dir-spec.txt.

[12] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, “Denial of service or denial of security? How attacks on reliability can compromise anonymity,” in *Proceedings of CCS 2007*, October 2007.

[13] 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.

[14] S. J. Murdoch and P. Zieliński, “Sampled traffic analysis by internet-exchange-level adversaries,” in *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, N. Borisov and P. Golle, Eds. Ottawa, Canada: Springer, June 2007.

[15] N. Evans, R. Dingledine, and C. Grothoff, “A practical congestion attack on tor using long paths,” in *Proceedings of the 18th USENIX Security Symposium*, August 2009.

[16] S. J. Murdoch and G. Danezis, “Low-cost traffic analysis of Tor,” in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[17] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, “Low-resource routing attacks against Tor,” in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, Washington, DC, USA, October 2007.

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

[19] N. Mathewson and R. Dingledine, “Practical traffic analysis: Extending and resisting statistical disclosure,” in *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, ser. LNCS, vol. 3424, May 2004, pp. 17–34.