Resolving Performance Issues in the Tor Network while Maintaining Anonymity

Michael J. Keen

Ball State University

Spring 2021

Abstract

Since its introduction to the general public in 2002, the Tor network and its creators have helped to kindle meaningful conversations about Internet surveillance and tracking of users by third parties, such as Internet service providers, government agencies, or web companies[7]. Tor has also emerged as an invaluable tool for people in authoritarian regimes around the world who seek to share or access unfiltered information while also avoiding government persecution[7]. However, despite the gains in Internet freedom it has been able to advance in its nearly 20 years of existence, it is not a perfect product, and it never was. Specifically, the network has long suffered from shortcomings in speed and performance, which stem from a variety of causes such as infrastructure design and malicious attacks. This paper will discuss several modern, prominent causes related to performance issues of the Tor network, as well as how these problems might be remedied without sacrificing anonymity to enable a safe, smoother experience for all users.

Section 1: Introduction

The Tor network is a volunteer-run system of relays (also known as onion routers and/or nodes) that enable users to initiate anonymous communications on the Internet.[1] This is made possible by a networking technique known as onion routing. In an onion routing implementation, a user's traffic passes through three random Tor relays before reaching its intended destination. The actual content of the traffic is enclosed in multiple layers of encryption, akin to the layers of an onion (hence the inspiration behind the procedure's name). As the traffic passes through each relay, a single layer of encryption is discarded. By the time the traffic exits the third and final relay, it will be fully decrypted. Nonetheless, due to the removal of the encrypted layers by the three relays, the true origin of the traffic is unknown to the destination site. And because the traffic is fully encrypted when it is passed to the first relay, the latter never has knowledge of the traffic's ultimate destination. Therefore, in theory, no one relay knows anything about the traffic besides its previous and next hop. This makes it exceptionally difficult for a third party to trace specific activity back to any one entity on the Tor network, therefore concealing a user's location and identity[1]. Because of this, the Tor network has emerged as a critical asset for the privacy-conscious, as well as individuals in autocratic nations and regions who wish to circumvent government-imposed censorship on the Internet and access or post content that will not be moderated by the regime[7]. It has also helped to contribute to the wider discussion regarding personal privacy, censorship, and technological surveillance by entities such as government agencies or web companies who may not necessarily respect users' desire for anonymity, impartial news, or self-expression.

However, the anonymity and privacy that the Tor network offers is often hindered by its performance issues, some of which are as old as Tor itself. From issues with bandwidth on relays, to the small number of relays available on the network, to congestion attacks that can further exacerbate existing bottlenecks, a satisfactory Tor user experience is heavily contingent upon avoiding these as much as possible. However, far too many Tor users encounter performance issues on a regular basis, making the network undesirable for use in the best case, and impractical for use in the worst case. This effectively cripples Tor's anonymity and privacy, as users who discontinue using Tor because of performance also forfeit the protection it offers. Addressing the root causes of the performance issues, while at the same time maintaining anonymity, would help to convince users to keep using Tor, and thus allow them to continue to benefit from the privacy and anonymity protection it offers.

Section 2: Problems in Performance

Performance issues have been a consistent problem for the Tor network since its creation. It is a common source of frustration for many Tor users, and it has consistently inhibited the adoption of Tor by more mainstream Internet users. The problem is further amplified by the network's architecture: because a Tor client must select three different relays and proceed to route its web traffic back and forth between said relays, the process for creating and maintaining a stable connection is much more complex than it is for Internet-enabled devices that do not use Tor. This results in an increased likelihood of a bottleneck in at least one part of the network exchange, which can lead to slower speeds for end-users. Furthermore, while there are many Tor clients who use the network each day, the number of relays available to service them is significantly smaller. To make matters worse, many of the relays that are available have only limited bandwidth to offer to clients. If they are forced to service more clients than their resources allow, then Tor clients who use them can expect to see speeds further reduced, which can lead to unreliable connections. Network performance can also suffer as a result of actions by malicious users, who may seek to exploit weaknesses in the Tor network and execute attacks such as congestion attacks or low-bandwidth attacks as a means of slowing down the network. The following subsections will discuss each issue in greater detail.

Section 2-1: Relay Bandwidth

One of the factors in Tor's performance issues stems from the bandwidth capacity of its relays. More specifically, Tor's load balancing algorithm is susceptible to obtaining inaccurate bandwidth values from said relays, which in turn stifles a Tor client's ability to find a relay with suitable bandwidth. As reported by Snader et al., the current algorithm relies on bandwidth values that are communicated to the network by each relay.[1] To prevent a relay from declaring an unrealistically high bandwidth capacity, the current Tor network architecture rejects any value that exceeds a specified upper bound value, which is presently at 10 MB/s[1]. However, despite this rule being in place, the Tor network is still susceptible to inaccurate reporting of bandwidth values by relays. There is currently no mechanism for independently verifying the bandwidth values that are publicly disclosed by a relay; each Tor client must simply accept them as accurate. This could potentially allow a malicious relay to drastically overstate its bandwidth, which would entice a large number of Tor clients to use it in their paths. According to Snader et. al., this is also a security risk; since the malicious relay is shown to have a high bandwidth capacity, it increases the likelihood that it could be selected as both the first and last relays in a Tor client's path, thus threatening the end user's anonymity[1]. Even in more benign cases where relays attempt to give a veracious report, many of these relays tend to overestimate their bandwidth capacity due to the perpetually fluctuating state of the Tor network. In both cases, a relay can become overloaded due to the number of client connections it must support, ultimately resulting in slower, more unreliable connections for many of the end users who use it in their paths. Furthermore, Snader et. al. show that this problem has been steadily growing worse since 2007, even though more relays have continuously been added over that time period[1]. They also show that not all Tor users necessarily desire a high level of anonymity, particularly if it involves sacrificing performance. Though some Tor users, such as those who live in censorship regimes, might still require stronger anonymity, other users in less hazardous circumstances might prefer to focus more on performance, especially if they are only interested in basic web browsing or other low-bandwidth, low-latency activities.

Section 2-2: Number of Relays

The Tor network is a free service to use, as anyone with a viable Internet connection is able to acquire the necessary software and access it with no monetary obligations. Nonetheless, for the network to function as intended, it depends on a volunteer group of relay operators who provide bandwidth to the network community at their own expense. By providing these resources, this group allows the network to sustain its scalability and thus allow it to remain practical for use among its clients. However, this operation model is jeopardized if too many new users decline to operate relays as the network grows larger. The number of users on Tor has grown significantly since 2007, which has helped to contribute to the rising popularity of the network in general[5]. Though the number of relays has also grown over this period, it has not grown on the same scale as the number of new users. In 2009, it was estimated that there were close to 100,000 active Tor clients. However, there were less than 2,000 relays on the Tor network that were available to service them[8]. Furthermore, both experienced and new Tor users face increasing obstacles to setting up a viable Tor relay, such as correctly configuring port forwarding on their routers, bandwidth allocation, possible legal implications (if operating an exit relay), etc. Because of this, some users simply reject the prospect of relay operation outright, finding that it contains too many added responsibilities and liabilities with too little reward. This mindset among too many Tor users severely restricts the network's ability to scale itself to support both existing users and new users. In the long term, this can result in consistently inadequate performance for many users, making Tor unusable even for low-bandwidth activities. Though Tor's creators have insisted that performance problems related to limited relays will eventually fade as the network sheds users, Ngan et al. have found that even in this instance, performance does not usually improve for the users who are left[5]. This is likely because users of low-bandwidth activities, such as basic web browsing and messaging, are more conscious of poor performance than users of high-bandwidth activities, such as those who download large files and stream video or music. Thus, the low-bandwidth users are more likely to make up a larger share of departing users. According to Panchenko et al., this also presents a security issue for the Tor users who remain, since the strength of Tor's anonymity has a direct dependence on the number of regular users[2]. Thus, in order to adequately address this issue, the Tor network should create a mechanism that will encourage users to set up and operate relays, even if it comes at a cost.

Section 2-3: Congestion Attacks

Though performance issues in the Tor network usually have benign origins, this is not always the case. According to Evans et al., Tor has been vulnerable to attacks involving congestion of network relays since at least 2005[4]. At that time, it was shown that an attacker was able to locate the relays for a particular Tor client (though not necessarily in the order of the path) and overload them, thus causing the end user's network performance to suffer[4]. Though this approach is no longer feasible due to the growth in the number of relays, the attack itself can still be implemented with some changes. This is possible in part because, prior to the publication of Evans et al.'s work, Tor server-side software did not place any restrictions on the number of relays that a client can include in their path. Thus, if a Tor client was modified to connect to an arbitrarily large number of relays in lieu of the usual three, then there was no mechanism on the Tor network to flag this and reject the connection[4]. Though this particular vulnerability has since been addressed, it is noted that other vulnerabilities have a similar implementation.

To initiate the attack described by the authors, one must first perform a timing attack on the target, thus capturing network activity and acquiring knowledge about the timing of requests.

The attacker must then log the times that each request appears, allowing them to identify the request sequence. After this, the attacker must modify the sequence in such a way that they are able to overload the Tor relays and determine the first relay in the target's path, thus completing the attack. Evans et al. state that rather than use a simulation, they decided to execute this attack on the actual Tor network[4]. They began their experiment by setting up their own malicious exit relay on the Tor network as well as their own clients (for ethical reasons, they did not involve actual clients that were outside of their control). To record the request sequence, the "victims" were made to execute a malicious JavaScript file that sent roughly one ICMP packet each second. These packets were detected by a server that was configured to listen for them specifically[4]. The "victim" then chose two regular Tor relays as its guard and middle relays, but used the malicious relay as the exit relay. To overload the connection, the researchers then set up both an HTTP server and client, as well as an altered Tor client that was configured to exploit the Tor network's lack of relay limits by forming an excessively long path through the relays that were to be congested. This was implemented by repeatedly routing the target relays as successor relays in the path, which was intended to accelerate the strain on their bandwidth[7]. The altered client was then made to download a large stream of random data by accessing the HTTP server through the HTTP client. While one end of the path was connected to the Tor client via the HTTP client, the other end was connected to the HTTP server, thus creating a passage for the large, random stream of data. The authors note that while this is one way of implementing the attack, it can also be done if the attacker has sufficient bandwidth to create a series of smaller Tor paths through the targeted relays and use them both to retrieve data from the HTTP server simultaneously. To determine how much the network had slowed down since the attack began, the researchers twice calculated the elapsed time between when the "victim" sent an ICMP packet to the malicious exit relay; they performed this calculation both before and after the altered Tor client began requesting the data stream from the HTTP server. Four separate trials were run to confirm the accuracy, with the results showing that the attack was effective (Figure 1)[4]. Thus, it presents a very real threat to the infrastructure of the Tor network.

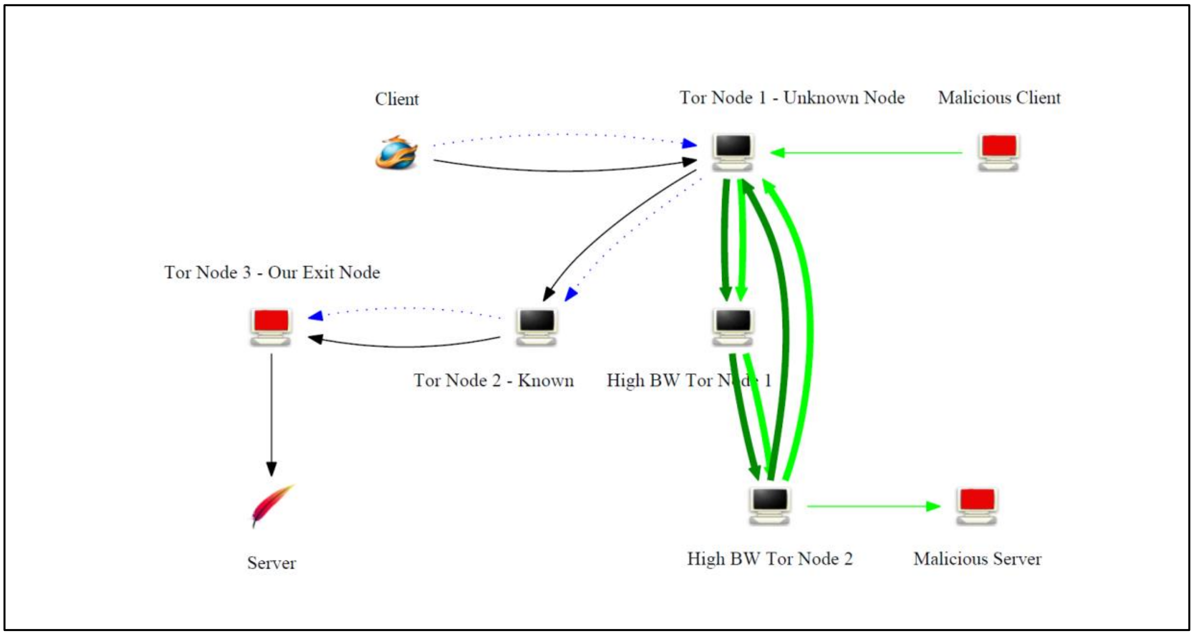


***Figure 1**: Attack as described by Evans et al. The light green/dark green path is the path chosen by the malicious client, and the left path with the malicious exit relay is the path chosen by the victim. The victim's connection is slowed due to the malicious client's attack on the first relay, which it shares with the victim.*

Section 3: Proposed Solutions

Though the factors at the heart of Tor's performance issues are significant, they can be remedied with the appropriate solutions. Even more so, the authors also address anonymity in their solutions, stating that they will either have no effect on, or could possibly increase, each user's anonymity. Though the described solutions are not immaculate, they nonetheless show that steps are being taken to address the above problems, and further research will be beneficial for strengthening the solutions.

Section 3-1: Tunable Path Selections

To address the issues with relay bandwidth, such as inaccurate reporting, Snader et al. propose revising Tor's load balancing algorithm. They state that replacing it with an "opportunistic bandwidth measurement" approach would, among other things, allow a Tor client to interact with relays on Tor before deciding whether to include a given relay in the client's path. More importantly, the new algorithm would offer Tor clients a mechanism for independently measuring router bandwidth, which would remove the need for relay-reported bandwidth values[1]. This is a preferable alternative for two reasons: it would eliminate the possibility of malicious low-bandwidth relays attacking Tor clients, and it would also enable a more precise prediction of relay performance, since benign relays may also misrepresent their bandwidth (albeit unintentionally). It would also help to ease the burden on overloaded relays by evaluating their diminished capacity and directing clients to less used relays, which would result in greater utilization and a significantly improved worst case for Tor clients[1]. Furthermore, these changes can be achieved without a substantial sacrifice of anonymity for most users.

To implement this functionality, Snader et al. propose that each Tor relay adopt "opportunistic monitoring", which would allow the relay to keep records of recent bandwidth performance from the relays around it[1]. This would be more efficient in comparison to other mechanisms such as probing, which would itself use the relay's limited bandwidth. Opportunistic monitoring, in contrast, would gather bandwidth data during the regular exchanges that each relay has with its surrounding relays, and thus would not require any additional resources. Each relay would then upload its data to a directory server, which would then communicate the information to Tor clients; directory servers are currently responsible for collecting each relay's bandwidth values. It is suggested that it would be beneficial for directory servers to aggregate the data, as this would also increase the accuracy of the relay's bandwidth relative to the Tor network's advertised bandwidth[1]. It would also help to use bandwidth more efficiently by directing clients away from higher capacity relays that may be overloaded, and towards lower capacity relays that might not have as much traffic.

Snader et al. tested their propositions by conducting a series of experiments both in the real Tor network as well as in a simulated network. In their tests, they downloaded a 1 MB file using a Tor path that included a high bandwidth exit relay and a normal Tor client. However, the entry and middle relays would be selected by the propositional framework. Their results in the real Tor network showed that the changes could improve a client's performance by up to 7 percent, even when the client's settings indicated a preferred focus on anonymity rather than performance. When the settings were changed to favor the highest level of performance at the cost of anonymity, over 85 percent of the clients were able to download the 1 MB file in less than one minute[1]. The results for this experiment are shown in Figure 2.

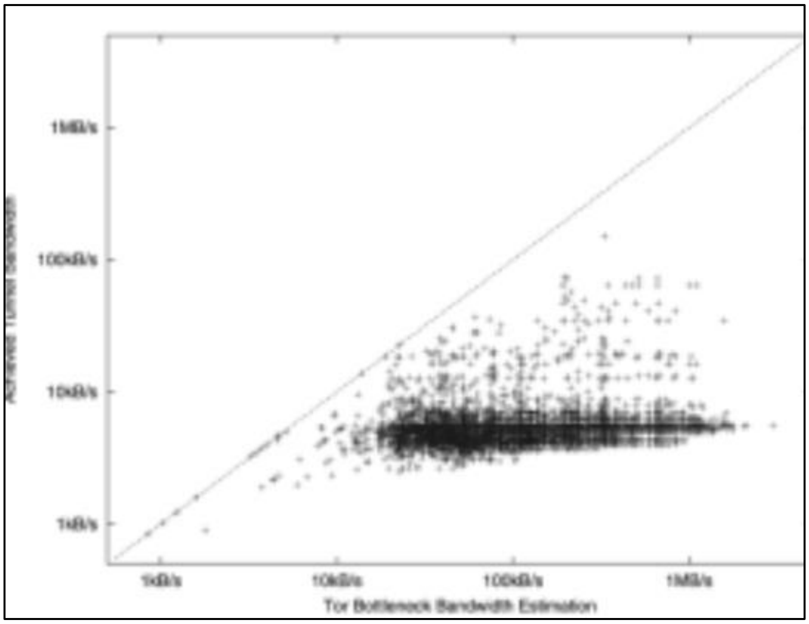


***Figure 2**: Results of experiment in the real Tor network. Compares Tor's current bandwidth measurements vs. the bandwidth obtained with the new mechanism.*

In the simulated Tor network, the authors also studied how the overall infrastructure might be affected if all clients were using the proposed changes. In this environment, they configured most of the clients to prefer maximum performance, with a smaller number favoring maximum anonymity and an even smaller number selecting an equal amount of both performance and anonymity. Their results for this experiment indicated that, if the changes were implemented on the real Tor network, then some users who opt to focus on maximum performance may inadvertently end up receiving slower speeds than users who focus both on performance and anonymity[1]. The authors attributed this outcome to bottlenecks that occurred at higher bandwidth Tor relays. They also revealed that users who preferred maximum anonymity had noticeably worse results than users who chose to focus on both performance and anonymity. Nonetheless, the results show that generally, the new algorithm would benefit the average Tor user with both improved performance and anonymity. The results for the simulated Tor network are shown in Figure 3.

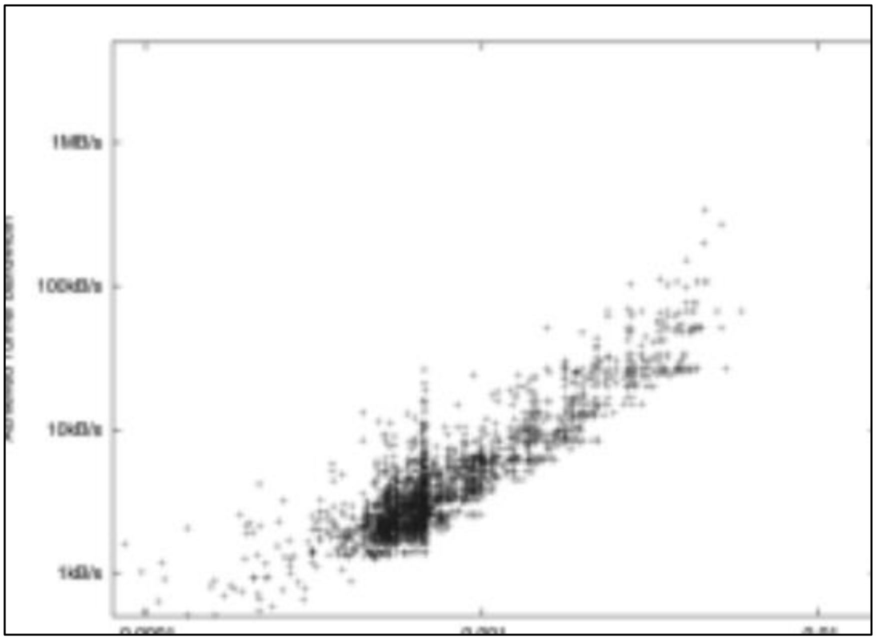
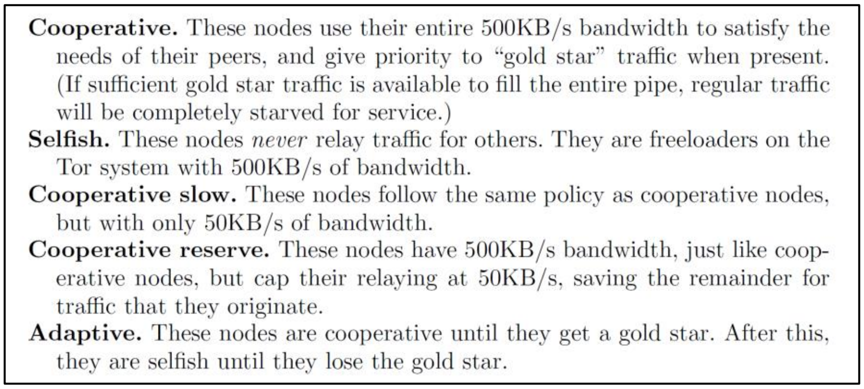


Figure 3: Results for the simulated Tor network. Compares bandwidth at various levels in ascending order.

Section 3-2: Relay Operation Incentives

One approach for persuading more Tor users to relay traffic, and thus increasing the amount of bandwidth available in the Tor network, is adding incentives for relay operators. However, the most effective design for an incentive scheme remains a matter of debate among researchers. One such design, introduced by Ngan et al., proposes a scheme that they call the "gold star" scheme[3]. Under this design, any Tor user who provides at least one relay with acceptable bandwidth that is available to other users will be eligible to receive an attribute known as a "gold star." This would allow any relay who receives the operator's own Tor traffic to mark it as high priority and forward it before any other traffic. The gold star attribute also stays with the operator's traffic for the duration of its time in the Tor network, ensuring that the traffic will continue to be designated as high priority in succeeding relays. The authors also contend that if the system were to be implemented in a certain way, anonymity would be maintained. In their model, Tor's directory servers were responsible for measuring each relay's bandwidth and conferring the gold star status when appropriate. When the directory servers transmit this data to the surrounding relays, they will obscure the identity of the given relay and keep it anonymous[3]. To test their scheme, the authors deployed a simulated Tor network and populated it with a total of 160 relays. These relays exhibited a variety of different behaviors that the researchers anticipated would be found on the actual Tor network; these behaviors are detailed in Figure 4.



Cooperative. These nodes use their entire 500KB/s bandwidth to satisfy the needs of their peers, and give priority to "gold star" traffic when present. (If sufficient gold star traffic is available to fill the entire pipe, regular traffic will be completely starved for service.)

Selfish. These nodes *never* relay traffic for others. They are freeloaders on the Tor system with 500KB/s of bandwidth.

Cooperative slow. These nodes follow the same policy as cooperative nodes, but with only 50KB/s of bandwidth.

Cooperative reserve. These nodes have 500KB/s bandwidth, just like cooperative nodes, but cap their relaying at 50KB/s, saving the remainder for traffic that they originate.

Adaptive. These nodes are cooperative until they get a gold star. After this, they are selfish until they lose the gold star.

Figure 4: Behaviors of relays in the simulated Tor network.

The relay behaviors in the experiment consisted of 40 cooperative, 40 selfish, 40 cooperative-slow, and 40 adaptive relays. The simulation also consisted of ten directory servers, each of which would build a random relay path to test each relay's bandwidth at least once per minute[3]. The results of the experiment found that the gold star scheme was highly effective in providing adequate performance for the cooperative relays. Though the advantage was only minimal in comparison to selfish and adaptive relays during periods of low traffic, the difference became much more distinguishable as the traffic grew[3]. Furthermore, even though cooperative slow relays had much more limited bandwidth to offer, they were still adequately rewarded with better performance. This indicates that it would be an effective scheme no matter how much available bandwidth a relay is able to offer[3].

However, other researchers, such as Jansen et al., have criticized the gold star system, stating that despite the benefits it is able to offer cooperative-slow relays, they are overshadowed by those given to cooperative relays, which have more bandwidth. The group has also scrutinized gold star's anonymity protection, finding that despite Ngan et al.'s claims, the system would only protect the anonymity of less than 65 percent of Tor users[5]. To address these shortcomings, Jansen et al. present an alternative scheme known as BRAIDS. In the BRAIDS system, they propose the creation of an entity, which they call "a centralized, partially-trusted, offline bank", that would be responsible for monitoring relay bandwidth and assigning rewards to relays who fulfill the requirements[5]. Nevertheless, all relays will remain anonymous to protect the identity of the operator, and the rewards given for operation will not have any identifying information that links them to their owner. The bank will reward relays by issuing one-time, authenticated tickets that the operators can then redeem to receive a specified category of higher-end performance. This allows users to opt for either low-latency or high-latency performance, the latter of which is best suited for large downloads and the former for basic web browsing/email. In addition, Jansen et al. state that no matter which option is picked, the system provides complete anonymity protection for all users on the network, thus resolving one of the gold star scheme's most significant shortcomings[5]. To test BRAIDS's ability to deliver on both low and high-latency performance, Jansen et al., like Ngan, deployed a simulated Tor network and populated it with a variety of high-latency nodes, such as file sharing clients and relays, as well as low-latency nodes such as web clients. Overall, the experiment consisted of about 20,000 web clients and file sharing nodes[5]. The results from the experiment indicated that all categories of services, specifically low-latency, saw a notable improvement in performance. The difference in performance between users without a relay and those with a relay was also significant, therefore providing a tangible incentive for users to run relays should BRAIDS be adopted by Tor. Thus, Jansen et al. believe that it would be a suitable alternative to the gold star scheme if the Tor network were to introduce incentives for relay operation.

Section 3-3: Limit on Relays in Tor Circuits

As mentioned in section 2-3, though Tor clients are configured to use only three relays in their paths (alternatively named circuits) by default, this can be modified by the Tor client's end user. Furthermore, prior to Evans et al.'s work, there was no mechanism in the Tor network for limiting the number of relays that a Tor client could use in its circuit, leaving it defenseless against attackers who attempted to construct extraordinarily long circuits. It was due to the public disclosure of the vulnerability that subsequent versions of Tor server software restricted circuit paths to 8 relays[4]. It achieves this by forcing Tor relays to monitor how many other relays are added to a given path, and to block the end user's traffic if it exceeds the 8-relay limit. However, even with these changes, the network remains vulnerable to more creative attackers, as they could exit the network completely via an exit relay and reconnect back to it as a new client. Evans et al. contend that, if conducted in sufficient numbers, the result could have the same congestion effect as an extraordinarily long Tor circuit. The attacker could also use a series of proxy servers, unlisted Tor relays, or other entities to circumvent the new restrictions and mount a practical congestion attack. A solution for this vulnerability would require a dramatic overhaul of the Tor network's foundational implementation, specifically its mechanisms for protecting user anonymity. The authors do not offer a solution for this type of attack, but recommend that it be researched further.

As a remedy in the interim, Evans et al., recommend that Tor users protect themselves by lengthening their relay circuit length to between 4 and 6 relays, and possibly by alternating between said values. This would make it more difficult for an attacker to target a potential victim due to the increase of time and bandwidth required for discovering the victim's full Tor circuit. However, the authors also acknowledge that such a maneuver would be a double-edged sword, as it would drastically increase bandwidth requirements not just for the attacker, but the Tor network as a whole due to the additional strain longer circuits would have on the relays. It would likely also increase the attack surface for other vulnerabilities in Tor. Thus, further research for a suitable remedy of this issue is likely warranted.

Section 4: Conclusion

The Tor network has served as an essential privacy tool for users around the world for almost 20 years. From acting as an aid to circumvent censorship in authoritarian regimes and regions, to helping other users avoid technological surveillance by government agencies or web companies, Tor has helped to define a golden standard for Internet freedom and privacy in the 21st century. However, performance-related issues threaten to brand Tor as unusable or undesirable to a large segment of the world population, thus rendering its anonymity and privacy protections useless. This paper has talked about some of the most significant causes behind Tor’s performance issues, and how they might be addressed without compromising anonymity. In some cases, the described solutions can even increase anonymity, making them especially ideal for adoption by Tor. Though some of the solutions are not perfect, they nonetheless represent a step in the right direction, and further research may yield information that can be used to address any shortcomings. Ensuring that Tor is able to perform at the highest level possible while maintaining its strong anonymity will help to encourage new and current users alike to use it, and thus create a smooth, secure experience for all users.

References

1. Snader, Robin, et al. “Improving Security and Performance in the Tor Network through Tunable Path Selection.” IEEE Xplore, IEEE, 2 Sept. 2010, ieeexplore.ieee.org/abstract/document/5560675.
2. Panchenko, Andriy, et al. “Improving Performance and Anonymity in the Tor Network.” IEEE Xplore, IEEE, 11 Jan. 2013, ieeexplore.ieee.org/abstract/document/6407715.
3. Ngan, Tsuen-Wan, et al. “Building Incentives into Tor.” Springer Link, Springer Link, 2010, link.springer.com/chapter/10.1007/978-3-642-14577-3_19.
4. Evans, Nathan S, et al. “A Practical Congestion Attack on Tor Using Long Paths.” USENIX, USENIX, Jan. 2009, www.usenix.org/legacy/event/sec09/tech/full_papers/evans.pdf.
5. Jansen, Rob, et al. “Recruiting New Tor Relays with BRAIDS.” Research Gate, Research Gate, 8 Nov. 2010, www.researchgate.net/publication/221609099_Recruiting_new_tor_relays_with_BRAIDS.
6. Schneier, Bruce. “The Tor Project: Privacy & Freedom Online.” *Tor Project*, The Tor Project, www.torproject.org/about/history/.