Building Incentives into Tor*

Tsuen-Wan "Johnny" Ngan[1], Roger Dingledine[2], and Dan S. Wallach[3]

[1] Google Inc.
[2] The Tor Project
[3] Department of Computer Science, Rice University

Abstract. Distributed anonymous communication networks like Tor depend on volunteers to donate their resources. However, the efforts of Tor volunteers have not grown as fast as the demands on the Tor network. We explore techniques to incentivize Tor users to relay Tor traffic too; if users contribute resources to the Tor overlay, they should receive faster service in return. In our design, the central Tor directory authorities measure performance and publish a list of Tor relays that should be given higher priority when establishing circuits. Simulations of our proposed design show that conforming users receive significant improvements in performance, in some cases experiencing twice the network throughput of selfish users who do not relay traffic for the Tor network.

1 Introduction

Anonymizing networks such as Tor [16] and Mixminion [11] aim to protect users from traffic analysis on the Internet. That is, they help defeat attempts to catalog who is talking to whom, who is using which websites, and so on. These anonymity systems have a broad range of users: ordinary citizens who want to avoid being profiled for targeted advertisements, corporations who do not want to reveal information to their competitors, and law enforcement and government agencies who need to interact with the Internet without being noticed.

These systems work by bouncing traffic around a network of relays operated around the world, and strong security comes from having a large and diverse network. To this end, Tor has fostered a community of volunteer relay operators. This approach can provide sustainability (the network doesn't shut down when the money runs out) and diversity (many different groups run relays for many different reasons), but it can also be a weakness if not enough people choose to operate relays to support the network's traffic.

In fact, the number of Tor users keeps growing [30], while a variety of factors discourage more people from setting up relays; some want to save their bandwidth for their own use, some can't be bothered to configure port forwarding on their firewall, and some worry about the possible consequences from running a relay. This growing user-to-relay ratio in turn hurts the service received by all users, leading to a classic "tragedy of the commons" situation [24].

* This research was funded, in part, by NSF grants CNS-0524211, CNS-0509297, and CNS-0959138. The first author did part of this work while at Rice University.

Worse, not all users are equal. While Tor was designed for web browsing, instant messaging, and other low-bandwidth communication, an increasing number of Internet users are looking for ways to anonymize high-volume communications. We did an informal measurement study by running a Tor exit relay at our institution, and we confirmed McCoy et al.'s results [32]: the median connection coming out of our relay looked like web browsing traffic, but the median *byte* looked like file-sharing traffic.

The Tor designers argued in 2005 [17] that having too much load on the Tor network should be self-correcting, since low bandwidth and poor performance would drive away users until the users that remain have acceptable performance. Instead, performance has remained bad for many users. We suggest this disparity is because different activities have different tolerance for bad performance: users of interactive applications like web browsing give up before the file-sharers, who are less sensitive to waiting hours for their work to complete.

How can we get more users to relay traffic? There are three common approaches to encouraging people to offer service in the p2p design space: building community, making it easier to run relays, and providing improved performance in exchange for service. So far Tor has focused most on the first two approaches, attracting people who believe in the need for anonymous communications to run relays. Tor now has over 1500 relays pushing over 1GBit/s of aggregate traffic [31], but it still has not kept up with demand. On the other hand, an accounting scheme for tracking nodes' performance and rewarding nodes who perform well would seem to be at odds with preserving anonymity.

This paper shows how to strike a balance between these seemingly conflicting goals. We propose a solution where the central Tor directory authorities measure the performance of each relay and construct a list of well-behaving relays. Relays obtain this list from the authorities during normal updates. To allow relays to be treated differently, traffic from relays in the list is marked as high priority by other relays and receives better treatment along the whole circuit.

The rest of the paper is organized as follows. Sect. 2 provides background on Tor. Sect. 3 investigates exactly which behaviors we need to incentivize. Sect. 4 describes our proposed design, and Sect. 5 presents simulation results showing our design improves performance for listed relays, even as traffic from other users increases. We discuss the results in Sect. 6, and evaluate the new risks our design introduces, the most notable of which is that we end up with two anonymity sets: the group of well-behaving relays and the group of other users and relays. We review related works in Sect. 7, and conclude in Sect. 8.

2 Background

The Tor network is an overlay network of volunteers running *Tor relays* that relay TCP streams for *Tor clients*. Tor lets its users connect to Internet destinations like websites while making it hard for 1) an attacker on the client side to learn the intended destination, 2) an attacker on the destination side to learn the client's location, and 3) any small group of relays to link the client to her destinations.

To connect to a destination via Tor, the client software incrementally creates a private pathway or *circuit* of encrypted connections through several Tor relays, negotiating a separate set of encryption keys for each hop along the circuit. The circuit is extended one hop at a time, and each relay along the way knows only the immediately previous and following relay in the circuit, so no single Tor relay knows the complete path that each fixed-sized data packet (or *cell*) will take. Thus, neither an eavesdropper nor a compromised relay can see both the connection's source and destination. Clients periodically rotate to a new circuit to complicate long-term linkability between different actions by a single user.

The client learns which relays it can use by fetching a signed list of Tor relays from the *directory authorities*. Each authority lists the available relays along with opinions on whether each relay is considered reliable, fast, and so on. Clients base their decisions on the consensus (i.e. the majority of authority opinions). Each authority's signing key comes with the Tor software, so Tor clients can't be tricked into using an alternate network run by an attacker. Authorities also provide a way for Tor users to synchronize their behavior; since anonymity loves company, users that make decisions based on similar information will blend together better [15]. A more detailed description of the Tor design can be found in its original design document [16] and its specifications [14].

Anonymity designs can be divided into two classes based on their goals: *high-latency* and *low-latency*. High-latency designs like Mixmaster [34] and Mixminion [11] can take hours to deliver messages, but because messages mix with each other they can withstand quite powerful attackers. These designs are not suitable for web surfing, which would be untenable with long latencies.

Tor chooses to build a practical and useful network, then try to achieve good security within these constraints. To that end, Tor doesn't batch or reorder messages at each hop. This choice means that Tor circuits are vulnerable to *end-to-end correlation attacks*: an attacker who can measure traffic at both ends of the circuit can link them [10, 28]. A variety of other anonymity-breaking attacks become possible because of Tor's requirement to remain useful for low-latency communications [26, 29, 35, 36, 41, 44].

Because Tor aims to resist *traffic analysis* attacks (attacks that try to pick the communicants out of a large set of participants) but does not aim to protect against correlation attacks (attacks that watch two suspected endpoints to confirm the link), we have some flexibility in what design changes we can propose. As long as we don't introduce any attacks that are worse than the correlation attacks, we are still within Tor's threat model.

3 Incentive Goals

Relayed traffic is traffic forwarded from a Tor client or Tor relay to another relay within the network. Choosing to relay traffic can provide better anonymity in some cases: an attacker who controls the user's next hop would not be able to know whether the connection originated at the user or was relayed from somebody else. But the exact details of the potential anonymity improvement

are not well-understood even by the research community. Therefore they are hard to communicate to users, so any potential perceived gains do not outweigh the costs of setting up relaying and providing bandwidth to others.

Tor relays may also opt to serve as exit relays. *Exit traffic* is forwarded from a Tor relay to somewhere outside the Tor network, as well as return traffic from outside back into Tor. While there are theoretical anonymity improvements similar to those for relaying traffic, as well as potential legal advantages for the relay operator from not necessarily being the originator of all traffic coming from the relay's IP address [20], in practice the destination website and the user's ISP have no idea that Tor exists, and so they assume all connections are from the operator. Some ISPs tolerate abuse complaints better than others. This hassle and legal uncertainty may drive users away from running as an exit relay.

Beyond creating incentives to relay traffic inside the Tor network and to allow connections to external services, we also need to consider the *quality* of the traffic (e.g., the latency and throughput provided, and the reliability and consistency of these properties). Since Tor circuits pass over several relays, the slowest relay in the circuit has the largest impact.

4 Design

Our solution is to give a "gold star" in the directory listing to relays that provide good service to others. A gold star relay's traffic is given higher priority by other relays, i.e., they always get relayed ahead of other traffic. Furthermore, when a gold star relay receives a high priority connection from another gold star relay, it passes on the gold star status so the connection remains high priority on the next hop. All other traffic gets low priority. If a low priority node relays data through a gold star relay, the traffic is still relayed but at low priority. Traffic priority is circuit-based. Once a circuit is created, its priority remains the same during its entire lifetime.

We can leverage Tor's existing directory authorities to actively measure the performance of each individual relay [42] and only grant those with satisfactory performance the gold star status. This measurement can include bandwidth and latency of the relayed traffic for that relay. By measuring the bandwidth through the Tor network itself, the directory authorities can hide their identity and intent from the Tor relays. This method of anonymously auditing nodes' behavior is similarly used in other systems [19, 38, 45].

Due to variations of the network conditions and the multi-hop nature of Tor, it may take multiple measurements to get accurate results. Therefore, we use a "k out of n" approach, where a relay has to have satisfactory performance for k times out of the last n measurements to be eligible for gold star status. At that point, it becomes a policy issue of who gets a gold star. We assign a gold star to the fastest 7/8 of the nodes, following the current Tor design in which the slowest one-eighth of Tor relays are not used to relay traffic at all. Of course, relays may choose not to give priority to gold star traffic. But in this case, they would most likely become the slowest nodes in the measurements and would

not earn a gold star. The directory authorities can then distribute the gold star status labels along with the network information they presently distribute.

The behaviors we most need to encourage will vary based on the demands facing the Tor network. Since there are enough exit relays currently, the design and analysis in this paper focuses on incentives for relayed traffic. However, our approach of giving priority to traffic from the *most useful* relays means that we can adapt the definition of "useful" as needed: e.g. we could vary the required threshold in the measurement tests above or require new tests such as verifying that exit traffic is handled correctly. We would then only need to publish the desired policy; users desiring higher priority for their traffic would then decide whether to follow the policy. We consider exit traffic testing more in Section 6.2.

The effectiveness of this design depends on the accuracy of the measurements, which in turn depends on the measurement frequency. Frequent measurements increase our confidence, but they also place a burden on the network and limit the scalability of the measuring nodes. Snader and Borisov [46] suggest an alternate approach to learning bandwidth, where every relay reports its own observations about the other relays, and the directory authorities use the median vote. If we used this approach, gold stars could then be assigned based on having a high enough median vote; but we note that active measurements would still be needed for verifying other properties such as exit traffic.

5 Experiments

Here we show simulations of the effectiveness of our "gold star" incentive scheme against different scenarios, including varying amounts of load on the Tor network, and varying strategies taken by simulated nodes (e.g., selfish vs. cooperative).

5.1 Experimental apparatus

We built a packet-level discrete event simulator that models a Tor overlay network. The simulator, written in Java, was executed on 64-bit AMD Opteron 252 dual processor servers with 4GB of RAM and running RedHat Enterprise Linux (kernel version 2.6.9) and Sun's JVM, version 1.5.0.

We simulate every cell at every hop. Each node, particularly simulated BitTorrent clients, can easily have hundreds of outstanding cells in the network at any particular time. Simulating 20 BitTorrent clients and 2000 web clients consumes most of the available memory. To keep the client-to-relay ratio realistic, we could only simulate Tor networks with around 150 relays.

For simplicity, we assumed the upstream and downstream bandwidth for all relays is symmetric, since the forwarding rate of any relay with asymmetric bandwidth will be limited by its lower upstream throughput. We also simplify relays by assuming they take no processing time. The cooperative relays (which reflect the altruists in the current Tor network) have a bandwidth of 500KB/s. The latency between any two nodes in the network is fixed at 100 ms.

Our simulations use different numbers of simplified web and BitTorrent clients to generate background traffic. Our web traffic is based on Hernández-Campos et al. [25]'s "Data Set 4," collected in April 2003 [48]. Our simplified BitTorrent clients always maintain four connections, and upload and download data at the maximum speed Tor allows. They also periodically replace their slowest connection with a new one, following BitTorrent's standard policy. We assume that the external web or BitTorrent servers have unlimited bandwidth. The different relay traffic types are:

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.

All of our simulations use ten directory authorities. To assign the gold star status, every minute each directory authority randomly builds a circuit with three Tor relays and measures its bandwidth by downloading a small, known 40KB file from an external server. The bandwidth measurement is recorded and attributed to only the middle relay in the circuit. (In a genuine deployment, the entry and exit nodes would be able to determine that they were being measured by virtue of being connected to known measurement nodes and could thus change their behavior in response.) To obtain a gold star, we require Tor relays to successfully relay traffic at least two times out of the last five measurements (i.e. $k = 2$ and $n = 5$ from Section 4). A relay is defined as successful if the directory authority can receive the correct file within a reasonable amount of time.

For our results, we describe the observed network performance in terms of "download time" and "ping time." Download time describes the time for nodes to download a 100KB file from an external server. Ping time describes the round-trip latency for that same external server. (This external server is assumed to have infinite bandwidth and introduce zero latency of its own.) Both measures are important indicators of how a Tor user might perceive Tor's quality when web surfing. For contrast, a Tor user downloading large files will be largely insensitive to ping times, caring only about throughput.

5.2 Experiment 1: Unincentivized Tor

First, we want to understand how Tor networks behave when demand for the network's resources exceeds its supply. We simulated 50 cooperative relays, 50

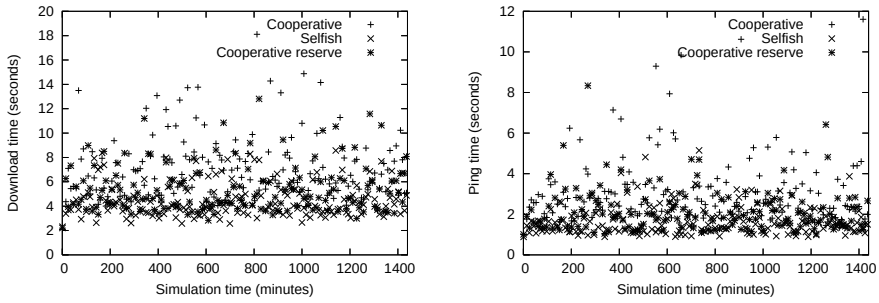

Fig. 1. Average download and ping time over time when no incentive scheme is in place and heavy traffic (20 BitTorrent clients and 2000 web clients). Both download and ping time show significant variation, regardless of relay type.

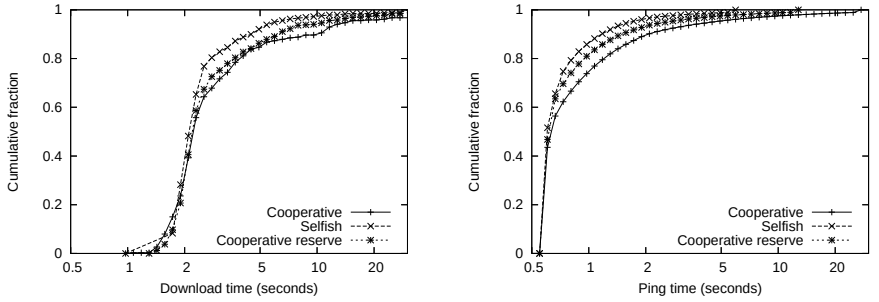

Fig. 2. Cumulative download and ping time when no incentive scheme is in place and heavy traffic (20 BitTorrent clients and 2000 web clients). Performance for all relay types is similar, although selfish relays do somewhat better in the worst case.

selfish relays, and 50 cooperative reserve relays with heavy background traffic (20 BitTorrent clients and 2000 web clients).

Figure 1 plots the *average* download and ping time for each relay type. Each plotted point is the average for 50 raw samples. Despite this, the variation among the data points suggests that the network performance is highly variable.

To get a better view of the distribution of download times and ping times, we use cumulative distribution functions (CDFs). Figure 2 represents the same data as Fig. 1, albeit without any of the data-point averaging. The x-axis represents download time or ping time and the y-axis represents the percentage of nodes who experienced that particular download or ping time *or less*.

While the ideal download time for all relay types in this experiment is 0.8 seconds (six network roundtrip hops plus transmission time), all relay types rarely achieve anywhere close to this number. Figure 2 shows that roughly 80% of the attempted downloads take more than two seconds, regardless of a node's policy. Approximately 10% of cooperative relays take longer than ten seconds. Selfish nodes, in general, do better in the worst case than cooperative nodes, but observe similar common-case performance.

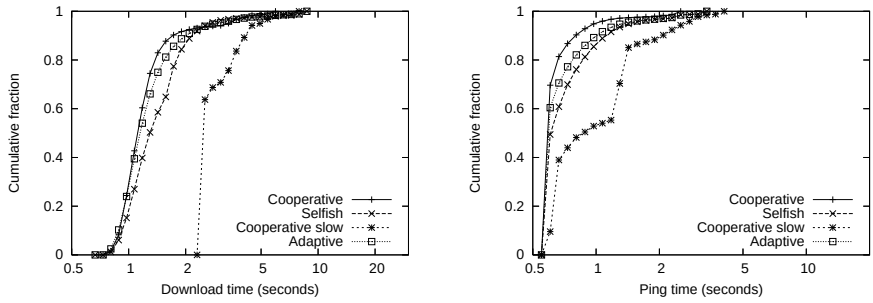

Fig. 3. Cumulative download and ping time with the gold star scheme and no background traffic.

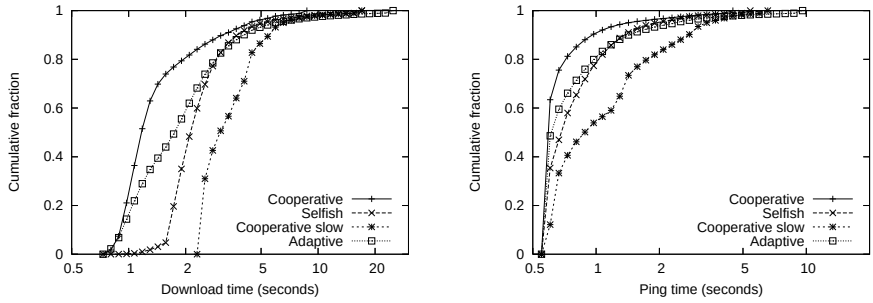

Fig. 4. Cumulative download and ping time with the gold star scheme and heavy background traffic (20 BitTorrent clients and 2000 web clients). Cooperative nodes maintain their performance, while selfish and adaptive nodes suffer.

5.3 Experiment 2: Gold stars

Our first experiment represents the present-day situation in the Tor network and is clearly unsatisfactory. This second experiment measures the effectiveness of our "gold star" mechanism in addressing this concern. This time, our simulation consists of 40 cooperative relays, 40 selfish relays, 40 cooperative slow relays, and 40 adaptive relays. These variations, relative to the first experiment, also allow us to see whether slower cooperative nodes still get the benefits of a gold star, and whether adaptive nodes can be more effective than purely selfish nodes. Figures 3 and 4 show the cumulative download and ping time with no background traffic and heavy background traffic, respectively.

Our results are striking. Cooperative nodes maintain their performance, regardless of the level of background traffic in the overlay. When there is no background traffic, they slightly outperform the selfish and adaptive nodes, but once the traffic grows, the cooperative nodes see clear improvements. For example, under heavy background traffic, 80% of the cooperative nodes see download times under two seconds, versus roughly 2.5 seconds for the selfish and adaptive nodes.

Our experiment also shows that the adaptive policy does not defeat the gold star mechanism. Adaptive nodes will experience better performance while

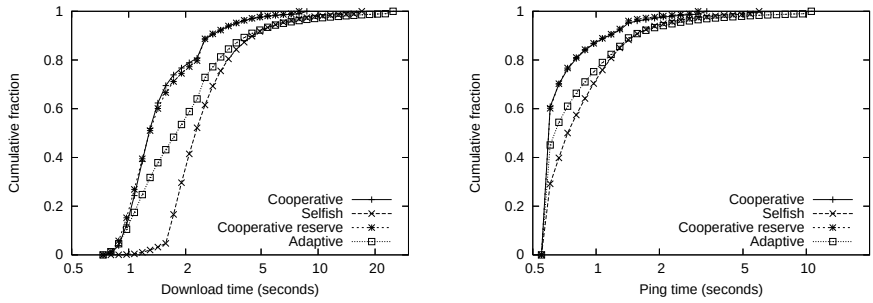

Fig. 5. Cumulative download and ping time with the gold star scheme and heavy background traffic (20 BitTorrent clients and 2000 web clients). Cooperative reserve relays, which replaced the cooperative slow relays, have similar performance to fully cooperative relays.

they have a gold star, but their benefit only splits the difference between the cooperative and selfish policies, roughly in proportion to the effort they are spending to maintain their gold star.

Cooperative slow nodes are always relatively slow due to their limited available bandwidth. However, like their fast counterparts, they experience stable performance as the background load on the Tor network increases. This demonstrates that the gold star policy can effectively reward good behavior, regardless of a node's available bandwidth.

We conducted a further experiment, replacing the cooperative slow nodes with cooperative reserve nodes, representing a possible rational response to the gold star mechanism. These nodes use 10% of their bandwidth for relaying and earning a gold star, reserving 90% of their bandwidth for their own needs. Figure 5 shows the results of this experiment. Both kinds of cooperative nodes observe identical distributions of bandwidth and latency. Selfish and adaptive nodes suffer as the background traffic increases. This experiment shows, unsurprisingly, that nodes need not be "fully" cooperative to gain a gold star. In an actual Tor deployment, it would become a policy matter, perhaps an adaptive process based on measuring the Tor network, to determine a suitable cutoff for granting gold stars (see Sect. 6.1).

5.4 Experiment 3: Alternating relays

This experiment considers a variation on the adaptive strategy used in the previous experiments. Alternating nodes will toggle between the cooperative and the selfish strategies on a longer timescale—four hours per switch. This experiment uses 50 such alternating relays with 50 cooperative relays and with heavy background traffic (20 BitTorrent clients and 2000 web clients).

Figure 6 shows the average download and ping time for both relay types over time. During the periods where the alternating relays are cooperative, they receive service of a similar quality as the full-time cooperative nodes. However,

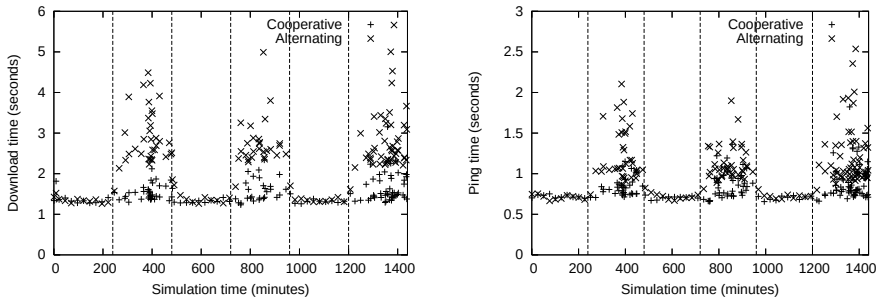

Fig. 6. Average download and ping time with relays that alternate between being cooperative and selfish. This experiment is with the gold star scheme in place and heavy background traffic (20 BitTorrent clients and 2000 web clients). Dotted lines show the times at which the alternating relays switch. The performance of alternating relays gets worse whenever they switch to being selfish, while performance for cooperative relays only suffers a little.

once the alternating relays switch to become selfish, their download times quickly increase, representing the same quality of service that would be observed by a selfish node. Note that while the cooperative nodes do observe lower quality of service (after all, fully half of the Tor nodes stopped relaying any data), they still do much better than their selfish peers. Our gold star system robustly responds to changes in node behavior.

5.5 Experiment 4: Pair-wise reputation

Last, we investigated a variation on our gold star design where individual circuits are not labelled as being low or high priority. Rather, each circuit inherits its priority from the status of the previous relay. That is, a low-priority node routing traffic through a gold-star node will experience delays getting the gold-star node to accept the traffic, but the traffic will have high priority in its subsequent hops. This alternative design has significant improvements from an anonymity perspective, because traffic at a given hop does not give any hint about whether it originated from a low-priority or high-priority node. However, our experiment showed selfish nodes clearly outperforming their cooperative peers. The results are shown in Appendix A.

6 Discussion

Our experiments show that our "gold star" technique is effective at giving higher priority to users who contribute to the Tor network. Nonetheless, a variety of questions remain about the policy that should be used for assigning gold stars and how the policy may affect the behavior of strategic Tor users.

6.1 Strategic users

Our proposed incentive scheme is not perfectly strategy-proof, in the sense that users can earn a gold star without providing *all* of their network capacity for the use of the Tor network (as in the "cooperative reserve" policy discussed in Sect. 5.3). This balance creates a variety of possible strategic behaviors.

Provide borderline or spotty service. A relay needs to provide only the minimal amount of bandwidth necessary to gain the gold star. Of course, if every user provided this amount, Tor would still have vastly greater resources than it does today. Next, because the bandwidth policies are determined centrally, the required minimum bandwidth for obtaining a gold star could be adjusted in response to network congestion. Strategic nodes would then adjust the capacity they offer, making more bandwidth available whenever they are needed.

Only relay at strategic times. Strategic users might provide relay services only when the "local" user is away, and thus not making demands on the Tor network. Such behavior is not disincentivized by our approach, as it still provides scalable resources to the Tor network. However, any users following such behavior may be partially compromising their anonymity, as their presence or absence will be externally observable.

Share a relay among several users. Several users could share a single entry relay into the Tor network, thus inheriting its gold star benefits without providing any additional bandwidth to the Tor network. In fact, we may even want to support this design, so users can run a fast relay at a colocation facility and then reap the rewards from their slower cable-modem or DSL Tor client. To allow the client to inherit the reputation of the server, the relay could be configured to give high priority to connections from a given set of IP addresses or Tor identity keys. On the other hand, multiple users that use a shared entry point must be able to trust one another. Lacking such trust, their desire for personal anonymity would incentivize them to run individual Tor relays.

Accept traffic only from known relays. In our design the directory authorities do their measurements anonymously via Tor, so all audits will come from other listed Tor relays. Thus a strategic relay could get away with giving poor performance (or no performance at all!) to connections from IP addresses not listed in the directory. One answer is that some of the measurements should be done through unlisted relays, perhaps by gathering a large pool of volunteer Tor users to help diversify the audit sources. Another answer is to turn this vulnerability around and call it a feature—another reason that users should want to get listed as a good relay.

Forward high-priority traffic as low-priority. A relay who correctly forwards traffic can still cheat by changing the priority on incoming traffic. The measuring authorities should build high-priority test circuits back to a trusted relay, to see if the circuit arrives with the expected high-priority status.

6.2 The audit arms race

Some attacks outlined above involve relays that provide some level of service but not quite as much as we might prefer. The response in each case is a smarter or more intensive measurement algorithm so the directory authorities can more precisely distinguish uncooperative behavior.

To see why this won't be an arms race between increasingly subtle cheating and increasingly sophisticated audits, we need to examine the incentives for ordinary users. Based on informal discussions with Tor relay operators, the most challenging part of setting up a Tor relay is configuring the software, enabling port forwarding in the firewall, etc. Compared to this initial barrier, the incremental cost of providing a bit more bandwidth is low for most users. As long as our audit mechanism correctly judges whether the user relays any traffic at all, we're verifying that the user has performed the most personally costly step in setting up a relay. We expect that the diminishing returns a strategic relay gets in saving bandwidth as we progress down the arms race will limit the complexity required for the auditing mechanism.

Measuring whether a relay is forwarding traffic adequately within the network is only one step. We could also extend our auditing techniques to measure whether an exit relay is in fact correctly forwarding exit traffic. We could thus incentivize exit traffic in the same way we incentivize relay traffic.

One concern in any measurement scheme over Tor is that the anonymity of Tor hides which node in an overlay route may have been responsible for degrading the quality of service. We could potentially "charge" all of the nodes in the route, but this could lead to "collateral reputation damage" for innocent nodes. An adversary may even strategically target a node for such damage. This ability to influence reputation can assist in anonymity-breaking attacks [6, 19].

6.3 Anonymity implications

Anonymity metrics like entropy [12, 43] apply to high-latency systems like Mixminion, where a global attacker aims to narrow the set of suspects to a small anonymity set (or to a probability distribution that has low entropy). With low-latency systems like Tor, most attacks either fail or reduce the entropy to zero. Thus these systems instead measure their security by the probability that a non-global attacker will be in the right positions in the network to launch an attack [47]. One key factor in this metric is the number of relays in the network.

Assuming the Tor network starts out with a small number of gold star relays, whenever a Tor relay receives a high priority cell, it knows with absolute certainty that the cell must have originated from a relay having a gold star. With so few gold star relays, the presence of high priority traffic greatly reduces the number of possible sources for that traffic. Worse, the set of users with a gold star is made public, further simplifying the attack.

We believe this tradeoff would still attract many gold star relays, though. First, altruists who don't use Tor as a client would still be the early adopters, as

predicted by Acquisti et al. [1] and as observed in the current Tor network. Low-sensitivity users would come next; many users who care more about performance than anonymity would be enticed into running Tor relays. The number of gold star nodes in the system should therefore increase over time, reducing the extent to which the presence of prioritized traffic gives away information to an attacker. We speculate that the growing anonymity set of gold star relays, along with the improved performance from being in the group getting priority traffic, would ultimately be enough to push many mainstream users into setting up relays.

We leave one major obstacle for future work: even if the anonymity set of gold-star users is large, changes in the set over time allow *intersection attacks* on anonymity. That is, if an attacker can link connections by a gold-star user (for example by tracking cookies or logins to a website), this attacker can narrow down which relay is running and has a gold star whenever such a connection occurs. One solution might be to make the gold star status persist a while after the relay stops offering service, to dampen out fluctuations in the anonymity sets. Depending on the rate of churn in the Tor network, this period might need to be a week or more, which means we then need to reevaluate the balance questions from this section. A more radical change would be to move to an ecash based service where getting high priority is less related to whether you're running a good relay at the time [2] – but we note that even in these more complex designs where there are multiple plausible reasons for a given user to get higher priority, users that earn their priority by relaying traffic can still be attacked [33].

6.4 The economics of attracting more relays

The experiments in Section 5 show that our design creates significant incentives for users to run Tor relays. As we attract more relays, the Tor network grows larger. Thus the anonymity that can be achieved increases for both the relays and the clients. As we attract more relays, the overall capacity in the network grows too. In fact, if we get a large enough network, the performance will improve compared to the currently deployed Tor network not only for the users who choose to run relays, but also for the users who don't!

If enough users do choose to run relays that there is excess network capacity, then the observable performance difference between high priority traffic and regular traffic might be insufficient to get more relays, or even to keep all of the current relays. If such a problem were to occur, one additional possibility would be to reserve bandwidth for high-priority traffic [40], effectively throttling low-priority traffic and creating a larger incentive for users to get a gold star. The downside to such an approach, of course, is that Tor performance would "needlessly" suffer for low-priority Tor users.

This discussion of the economics of our incentive strategy leaves out many details. We should start by analyzing the various equilibria and deriving utility functions for various user classes. We leave this investigation to future work.

7 Related Work

7.1 Incentives in anonymous communication networks

Real-world anonymizing networks have operated on three incentive approaches: *community support*, *payment for service*, and *government support*. (Discussion of the funding approaches for research and development of anonymity designs, while related, is outside the scope of this paper.) The Tor network right now is built on community support: a group of volunteers from around the Internet donate their resources because they want the network to exist.

Zero-Knowledge Systems' Freedom network [7] on the other hand was a commercial anonymity service. They collected money from their users, and paid commercial ISPs to relay traffic. While that particular company failed to make its business model work, the more modest Anonymizer [3] successfully operates a commercial one-hop proxy based on a similar approach. PAR [2] proposes a micropayment model where clients pay coins for each circuit, and relays can use these coins for service of their own or convert them into actual payments; however, its dependency on an ecash bank means it remains a theoretical design. Lastly, the AN.ON project's cascade-based network was directly funded by the German government as part of a research project [4]. Unfortunately, the funding ended in 2007, so they are exploring the community support approach (several of their nodes are now operated by other universities) and the pay-for-play approach (setting up commercial cascades that provide more reliable service).

Other incentive approaches have been discussed as well. Acquisti et al. [1] argued that high-needs users (people who place a high value on their anonymity) will opt to relay traffic in order to attract low-needs users — and that some level of free riding is actually beneficial because it provides cover traffic. It is unclear how well that argument transitions from the high-latency systems analyzed in that paper to low-latency systems like Tor.

7.2 Incentives in other peer-to-peer networks

Incentives for applications. Incentive schemes have been proposed for several p2p applications. BitTorrent [8], one of the pioneers, facilitates large numbers of nodes sharing the effort of downloading very large files. Every node will have acquired some subset of the file and will trade blocks with other nodes until it has the rest. Nodes will preferentially trade blocks with peers that give them better service ("tit-for-tat" trading). Scrivener [37] addresses a more general problem, where nodes are interested in sharing a larger set of smaller files.

In a storage network, nodes share spare disk capacity for applications such as distributed backup systems. Ngan et al. [38] proposed an auditing mechanism, allowing cheaters to be discovered and evicted from the system. Samsara [9] enforced fairness by requiring an equal exchange of storage space between peers and by challenging peers periodically to prove that they are actually storing the data. Tangler [50] required users to provide resources for a probation period before they are allowed to consume resources.

Reputation systems. Resource allocation and accountability problems are fundamental to p2p systems. Dingledine et al. [13] survey many schemes for tracking nodes' reputations. In particular, if obtaining a new identity is cheap and positive reputations have value, negative reputation could be shed easily by leaving the system and rejoining with a new identity. Friedman and Resnick [21] also study the case of cheap pseudonyms, and argue that suspicion of strangers is costly. EigenTrust [27] is a distributed algorithm for nodes to securely compute global trust values based on their past performance. Blanc et al. [5] suggest a reputation system for incentivizing routing in peer-to-peer networks that uses a trusted authority to manage the reputation values for all peers, comparable to our use of directory authorities.

Trading and payments. SHARP [22] is a framework for distributed resource management, where users trade resources with trusted peers. KARMA [49] and SeAl [39] rely on auditor sets to track the resource usage of each node in the network. Golle et al. [23] considered p2p systems with micro-payments, analyzing how various user strategies reach equilibrium within a game theoretic model.

Tradeoff between anonymity and performance. If the number of gold star relays in the network is small, sending gold star traffic may result in reduced anonymity, albeit better performance. This introduces another dimension of traffic control. In our design a gold star relay is not required to send its own traffic at high priority; it may choose to send it at a low priority for better anonymity. This tradeoff is similar to the idea in Alpha-mixing [18], where the sender can use a parameter to choose between better anonymity and lower latency.

8 Conclusions

This paper proposes an incentive scheme to reward Tor users who relay traffic. Our simulations show that we can correctly identify nodes who cooperate with our desired policies, and they achieve sizable performance improvements, particularly in Tor's current situation where the Tor network is saturated with traffic. While we reduce anonymity for cooperative ("gold star") nodes because any high priority traffic must have originated from a gold star node, we create significant performance incentives for many users to join the Tor network as relays, which improves both performance *and* anonymity.

Once our technique is ready to be deployed on the live Tor network, both pragmatic and theoretical concerns remain. For example, we cannot predict future demand on the Tor network, nor can we predict the extent to which firewalls or ISP bandwidth policies might interfere with Tor or otherwise disincentivize users from relaying Tor traffic. We should also investigate the extent to which the centralized Tor management nodes might be able to coordinate their network measurements and agree on optimal incentivization policies as network conditions evolve.

References

1. Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the economics of anonymity. In *Proceedings of the 7th Annual Conference on Financial Cryptography (FC '03)*, Gosier, Guadeloupe, January 2003.
2. Elli Androulaki, Mariana Raykova, Shreyas Srivatsan, Angelos Stavrou, and Steven M. Bellovin. PAR: Payment for anonymous routing. In *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 219–236, Leuven, Belgium, July 2008.
3. The Anonymizer. `http://www.anonymizer.com/`.
4. Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, Coventry, UK, July 2000.
5. Alberto Blanc, Yi-Kai Liu, and Amin Vahdat. Designing incentives for peer-to-peer routing. In *Proceedings of the 24th IEEE INFOCOM*, Miami, FL, March 2005.
6. Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? How attacks on reliability can compromise anonymity. In *Proceedings of CCS 2007*, October 2007.
7. Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000. `http://osiris.978.org/~brianr/crypto-research/anon/www.freedom.net/products/whitepapers/Freedom_System_2_Architecture.pdf`.
8. Bram Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.
9. Landon P. Cox and Brian D. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proc. 19th ACM Symp. on Operating System Principles (SOSP '03)*, Bolton Landing, NY, October 2003.
10. George Danezis. The traffic analysis of continuous-time mixes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *LNCS*, pages 35–50, Toronto, Canada, May 2004.
11. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.
12. Claudia Díaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
13. Roger Dingledine, Michael J. Freedman, and David Molnar. Accountability measures for peer-to-peer systems. In *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly and Associates, November 2000.
14. Roger Dingledine and Nick Mathewson. Tor protocol specification. `https://www.torproject.org/svn/trunk/doc/spec/tor-spec.txt`.
15. Roger Dingledine and Nick Mathewson. Anonymity loves company: Usability and the network effect. In *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, Cambridge, UK, June 2006.
16. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of 13th USENIX Security Symposium*, San Diego, CA, August 2004. Project web site: `https://www.torproject.org/`.
17. Roger Dingledine, Nick Mathewson, and Paul Syverson. Challenges in deploying low-latency anonymity. Technical Report 5540-265, Center for High Assurance Computer Systems, Naval Research Laboratory, 2005.

18. Roger Dingledine, Andrei Serjantov, and Paul Syverson. Blending different latency traffic with alpha-mixing. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, UK, 2006.
19. Roger Dingledine and Paul Syverson. Reliable MIX cascade networks through reputation. In *Proceedings of the 6th Annual Conference on Financial Cryptography (FC '02)*, Southampton, Bermuda, March 2002.
20. Electronic Frontier Foundation. Tor: Legal FAQ for Tor server operators. `https://www.torproject.org/eff/tor-legal-faq.html`.
21. Eric Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.
22. Yun Fu, Jeffrey S. Chase, Brent N. Chun, Stephen Schwab, and Amin Vahdat. SHARP: An architecture for secure resource peering. In *Proc. 19th ACM Symp. on Operating System Principles (SOSP '03)*, Bolton Landing, NY, Oct 2003.
23. Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for sharing in peer-to-peer networks. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, FL, October 2001.
24. Garrett Hardin. The tragedy of the commons. *Science*, 162, 1968. Alternate location: `http://dieoff.com/page95.htm`.
25. Félix Hernández-Campos, Kevin Jeffay, and F. Donelson Smith. Tracking the evolution of web traffic: 1995–2003. In *Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Orlando, FL, October 2003.
26. Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-Tin. How much anonymity does network latency leak? In *Proceedings of the 14th ACM Conference on Computer and Communication Security*, Alexandria, VA, October 2007.
27. Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 2003.
28. Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew K. Wright. Timing attacks in low-latency mix-based systems. In *Proceedings of the 8th Annual Conference on Financial Cryptography (FC '04)*, Key West, Florida, February 2004.
29. Marc Liberatore and Brian Neil Levine. Inferring the Source of Encrypted HTTP Connections. In *Proceedings of the 13th ACM conference on Computer and Communications Security (CCS 2006)*, pages 255–263, Alexandria, VA, October 2006.
30. Karsten Loesing. Evaluation of client requests to the directories to determine total numbers and countries of users. Technical report, The Tor Project, June 2009. `https://torproject.org/projects/metrics`.
31. Karsten Loesing. Measuring the Tor network from public directory information. Technical report, 2nd Hot Topics in Privacy Enhancing Technologies (HotPETs 2009), Seattle, WA, USA, August 2009.
32. Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the Tor network. In *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, Leuven, Belgium, July 2008.
33. Jon McLachlan and Nicholas Hopper. On the risks of serving whenever you surf: Vulnerabilities in Tor's blocking resistance design. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2009)*. ACM, November 2009.
34. Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster protocol — version 2. IETF Internet Draft, July 2003. `http://www.abditum.com/mixmaster-spec.txt`.

35. Steven J. Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA, May 2005.
36. Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by Internet-exchange-level adversaries. In *Proceedings of Privacy Enhancing Technologies Symposium (PET 2007)*, Ottawa, Canada, June 2007.
37. Animesh Nandi, Tsuen-Wan "Johnny" Ngan, Atul Singh, Peter Druschel, and Dan S. Wallach. Scrivener: Providing incentives in cooperative content distribution systems. In *Proceedings of the ACM/IFIP/USENIX 6th International Middleware Conference (Middleware 2005)*, Grenoble, France, November 2005.
38. Tsuen-Wan "Johnny" Ngan, Dan S. Wallach, and Peter Druschel. Enforcing fair sharing of peer-to-peer resources. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, CA, February 2003.
39. Nikos Ntarmos and Peter Triantafillou. SeAl: Managing accesses and data in peer-to-peer sharing networks. In *Proceedings of the 4th IEEE International Conference on P2P Computing*, Zurich, Switzerland, August 2004.
40. Andrew M. Odlyzko. Paris metro pricing for the Internet. In *ACM Conference on Electronic Commerce*, pages 140–147, 1999.
41. Lasse Øverlier and Paul Syverson. Locating hidden servers. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 2006.
42. Mike Perry. TorFlow: Tor Network Analysis. Technical report, 2nd Hot Topics in Privacy Enhancing Technologies (HotPETs 2009), Seattle, WA, USA, August 2009.
43. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
44. Vitaly Shmatikov and Ming-Hsui Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In *Proceedings of the 11th European Symposium On Research In Computer Security (ESORICS 2006)*, Hamburg, Germany, Sept 2006.
45. Atul Singh, Tsuen-Wan "Johnny" Ngan, Peter Druschel, and Dan S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In *Processings of IEEE INFOCOM*, Barcelona, Spain, April 2006.
46. Robin Snader and Nikita 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.
47. Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.
48. The Distributed and Real-Time Systems Research Group, UNC. Data for the UNC HTTP traffic model.
`http://www.cs.unc.edu/Research/dirt/proj/http-model/`.
49. Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gün Sirer. KARMA: A secure economic framework for p2p resource sharing. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.
50. Marc Waldman and David Mazières. Tangler: A censorship resistant publishing system based on document entanglements. In *Proceedings of the 8th ACM Conference on Computer and Communication Security (CCS 2001)*, Philadelphia, Pennsylvania, November 2001.

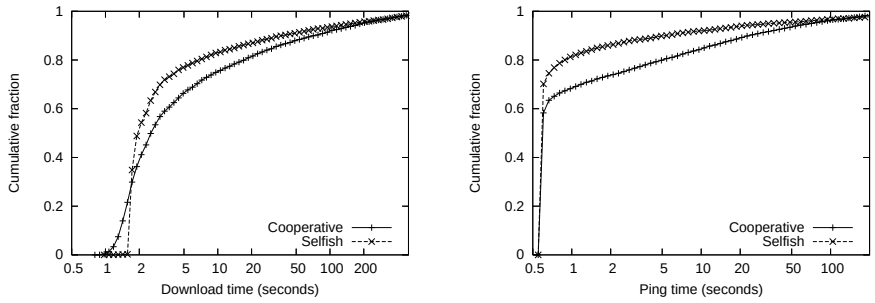

Fig. 7. Cumulative download and ping time with the pair-wise reputation design and heavy traffic (20 BitTorrent clients and 2000 web clients). Four relay types (cooperative, selfish, cooperative reserve, and adaptive) are simulated, although only the performance of the former two are shown, as the latter two behave similarly to cooperative relays.

A Experiment 4: Pair-wise reputation

In this experiment, we investigated a variation on our gold star design, where individual circuits are not labelled as being low or high priority. In this variation, a low-priority node routing traffic through a gold-star node will experience delays getting the gold-star node to accept the traffic, but the traffic will have the gold-star priority in its subsequent hops. This alternative design has significant improvements from an anonymity perspective, because traffic at a given hop doesn't give any hint about whether it originated from a low-priority or high-priority node. However, this design might fail from an incentives perspective, since there is less incentive for a node to earn its own gold star.

In this experiment, we again simulate a network with 40 relays for each relay type: cooperative, selfish, cooperative reserve, and adaptive. For clarity, Fig. 7 only shows the download and ping time for cooperative and selfish relays, as the performance for cooperative reserve and adaptive relays is very close to that for cooperative relays.

This experiment shows selfish nodes clearly outperforming their cooperative peers. This indicates that the gold star strategy requires a transitive property, i.e., each hop of a circuit must inherit the gold star status of the previous hop. Otherwise, selfish nodes will outperform their cooperative peers and there will be no incentive for cooperation.