

TorCoin

Bryan Ford, Miles Richardson, and Mainak Ghosh

Yale University, New Haven, CT

{bryan.ford, miles.richardson, mainak.ghosh}@yale.edu

Abstract. The scalability of the Tor Anonymity Network suffers from the lack of an incentive to volunteer bandwidth resources, which are required for the system to operate effectively. This paper identifies the challenges and issues involved in producing such incentives, and proposes an architectural design with acceptable trade-offs that can be realized with mostly existing technologies.

1 Introduction

The Tor network suffers from slow speeds, because of a shortage of relay nodes, unsustainably provided by altruistic volunteers. This is a well studied problem. Despite many attempts, researchers have failed to deploy a widely-adopted mechanism for compensating relay operators, while retaining anonymity of clients. [1–7].

First, we propose a radical rethinking of the economic structure of the Tor anonymity network. Specifically, we introduce the idea of *privacy service providers* (PSPs), analagous to internet service providers (ISPs) in current nomenclature. A PSP operates an independent Tor network. It charges clients for access, and pays relay operators per-megabit of bandwidth transferred.

Second, we present an anonymity-preserving software architecture to enable this economic model, without modifying the core codebase of Tor. We introduce ‘*TorCoin*’, an alternative cryptocurrency (altcoin) that relay operators can mine by transferring bandwidth. It uses the BitCoin protocol, but relies on proof-of-bandwidth instead of proof-of-computation.

TorCoin enables the economic model to work. Relay operators install software allowing them to verifiably mint TorCoins per-mb of transfer, then sell them on a *TorCoin Exchange* for cash. Each PSP operates an exchange; clients pay cash to access the network, and the PSP pays cash to relays for TorCoins.

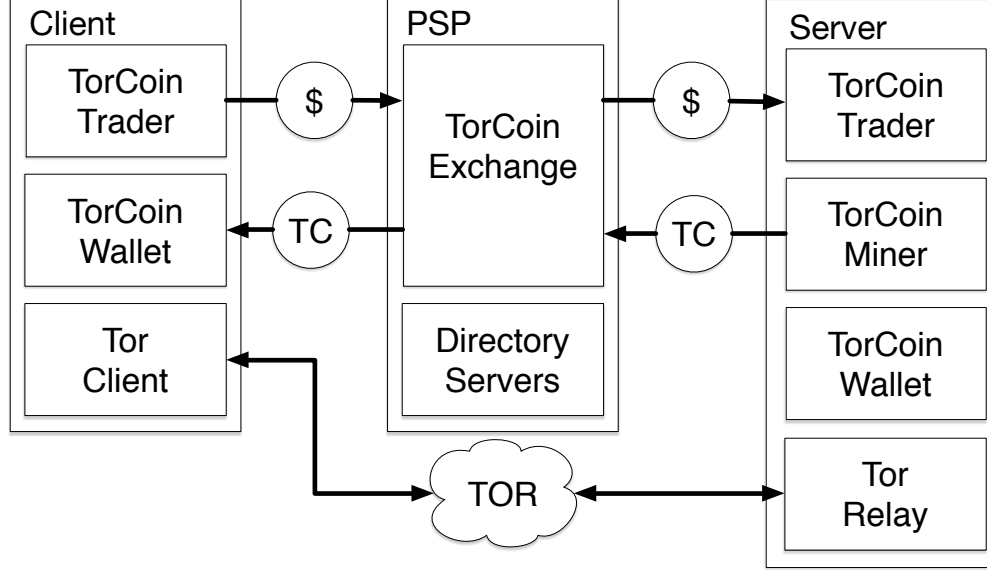
2 Economic Architecture

Todo - DRAFT NOTES

This section will detail the economic model and present any lingering questions. There will be another diagram featuring multiple clients, a PSP, and multiple hosts. The diagram below will go into the software architecture section, under a section called “system design”, followed by a sentence describing each component. Then we will go into detail on the ephemeral paths and TorCoin design, which are part of the TorCoin Minter. The TorCoin trader, wallet, and

exchange are all necessary, but not novel in terms of design, so we will spend little time on those.

TODO: Add component to Client, “TorCoin Minter” (?)



3 Proposed Architecture

We now discuss the main components necessary to realize a practical Tor incentive system while identifying some open research and development problems.

3.1 Overview

We propose a system that measures bandwidth contributed to the Tor network to produce incentive the addition of nodes to the Tor network.

The system measures the bandwidth contributed by each relay in the Tor network and rewards them with a 'TorCoin'. A Torcoin is an AltCoin that uses a bandwidth-intensive protocol as its proof-of-work. Thus, to produce a TorCoin, a relay must have transmitted a certain amount of Tor traffic.

To reduce the system's vulnerability to attackers and possible reduction of anonymity, we also utilize a system of 'Ephemeral Paths' to randomly assign relays to clients.

These TorCoins can then be traded at an exchange for other AltCoins or other goods. This forms the basis of our incentivization scheme. This is different from systems that propose differentiated service [8,9], since we do not propose to make the clients pay for access to the network. The coins are a byproduct of the usage of the system.

3.2 Ephemeral Paths

The TorCoin system adheres to the following constraints:

- No client in the group can generate its own route.
- Every resulting route has a unique public key (Route signature).
- No client in the group can know the route assigned to another client in the group.
- Any interested party can verify that a given public key represents a route assigned to a client in the current group.

Setup The Tor directory servers will create the groups using the temporal locality of the clients connecting to them, but also ensure that there is geographical and other diversity in a group. This is to ensure that adversaries cannot deterministically place themselves in a single group by connecting at the same time.

A group consists of the first $n = 1000$ or so clients that have connected to the Assigning servers. In practice, we expect the number n to be modulated so that groups are being created every 10 seconds or so. Also, we will ensure diversity in the group by ensuring that a group consists of a majority of the Assigning servers. Thus, if there are 10 Assigning Servers in the entire network, a group must consist of at least 6 of them.

Route Assignment using Neff shuffles Once all the clients have filled up the group, the Assigning Servers in the group initiate the process of route assignment.

1. Three groups of two Assigning Servers are chosen from those in the group.
2. Each group of Assigning servers uses a unique Neff shuffle to create shuffled lists of entry, middle and exit relays. The i th client in the group is thus matched up with the i th entry, middle and exit relay to form a 'route'.
3. The servers then create a route-signature for each route based on some property of each participant (client and three relays).
4. Each route-signature is then input into a cryptographic accumulator so that anyone can verify if a route belonged to a group. This accumulator is published in a publicly available log along with the timestamp of the group creation.

Each server then sends each client in the group one piece of data. Eg: The servers that assigned the entry relays send each client its own entry relay. Similarly, the middle and exit relays and the route signatures are also communicated in the same way. For each relay, the servers send an Access Control List. This is a list of all the relays and clients that the relay should accept connections from, as well as the route signatures for each.

In this way, no single server is aware of any client's entire path through the network, preserving their anonymity. Since each relay is confirmed by at least two servers, it is also robust to rogue servers.

Proof of Bandwidth - Onion Hashing Once the routes are setup, we can prove bandwidth transfer through the following protocol: Every m Tor packets, the client sends an extra packet (the Torcoin packet) containing an hash attempt likely to generate a TorCoin. Relay A gets it, generates a temporary private key

Ka (generated using the route shared key) and hashes the received packet and this key. It then forwards it to B, which does the same thing, with its own private temporary key Kb. Similarly on to C. C can now add its own Kc, and if it generates a hash with a given number of zeros, it can claim to have found a TorCoin.

Client sends to A: T0 (its hash attempt)

A sends to B : Hash(T0 + K1) = Ta # K1 is A's temporary private key.

B sends to C : Hash(Ta + K2) = Tb # K2 is Bs temporary private key.

C computes : Hash(Tb + K3) = Tc # K3 is Cs temporary private key.

C sends to B : (Tc, K3) to verify.

B sends to A : (Tc, K3, Tb, K2) to verify.

A sends to client: (Tc, K3, Tb, K2, Ta, K1) to verify.

Once the client has verified the hash, we can confirm that the data has made a round trip through the route. This completes the proof of bandwidth.

TorCoin We can then implement an AltCoin based on the proof-of-work concept in the following manner:

If (Tc = '000...')

If the client successfully verifies the hash

It adds the coin to the blockchain with the following information:

0. Timestamp of group.

1. Clients public key.

2. Route Shared key (Lets any other group member verify that the route is genuine. See accumulator.)

4. TorCoin Hash.

It then gives 1/3rd of the coin to each of the relays in the route. (If the client is rogue, can we identify and kick the client off?)

This information in the blockchain will enable any interested party to verify if the route came from a legitimate group formed by the directory servers. This can be done by taking the route signature and comparing it with the publicly available accumulators.

The properties of the Altcoin will then take over, with clients building on the blockchain with more and more TorCoins that they mine through this process.

3.3 Robustness to attack

The entire reason for constructing the elaborate ephemeral routes algorithm is to make the Torcoin system robust to attackers.

Due to the random group selection system, it is hard for attackers to deterministically place themselves in a group. To make the system even more secure, the servers can randomize group assignment instead of just taking temporal locality to be the criterion.

In addition, because the attacker needs to control all four components of a route to mint a TorCoin fraudulently, even if the adversaries control up to half the network, there is a probability of only 1/16 that an adversary client gets a

path of three colluding relays. In practice, gaining control of half of the entire Tor client and relay network is practically impossible.

A separate rate-limiting mechanism can be deployed to detect dishonest relays and remove them from the the path selection procedure. An independent verification authority, such as one based on Eigenspeed, could be used to detect these discrepancies.

4 Preliminary Results

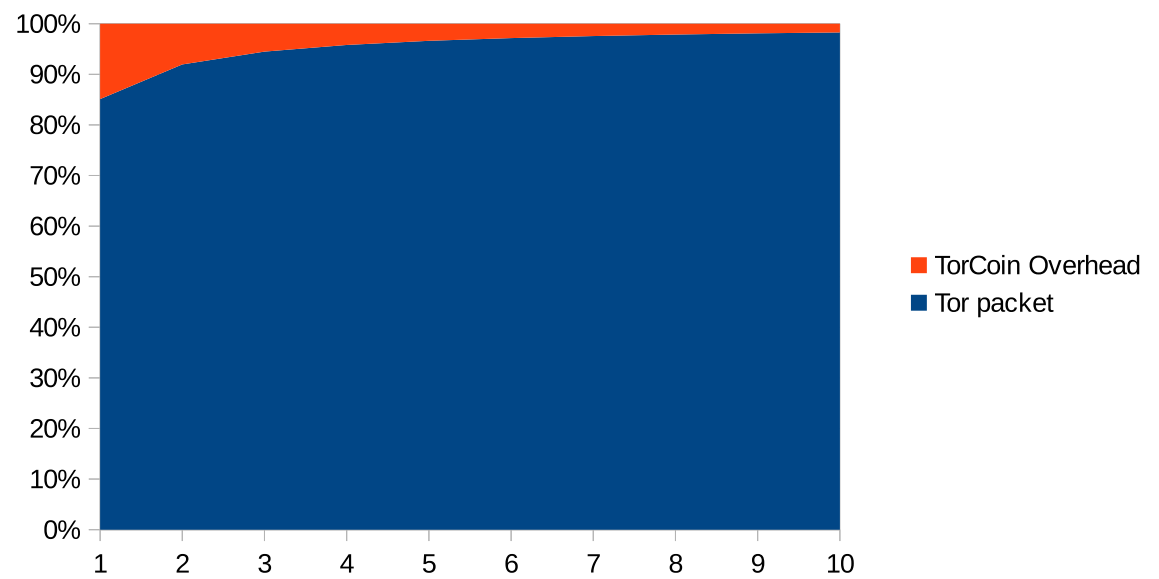
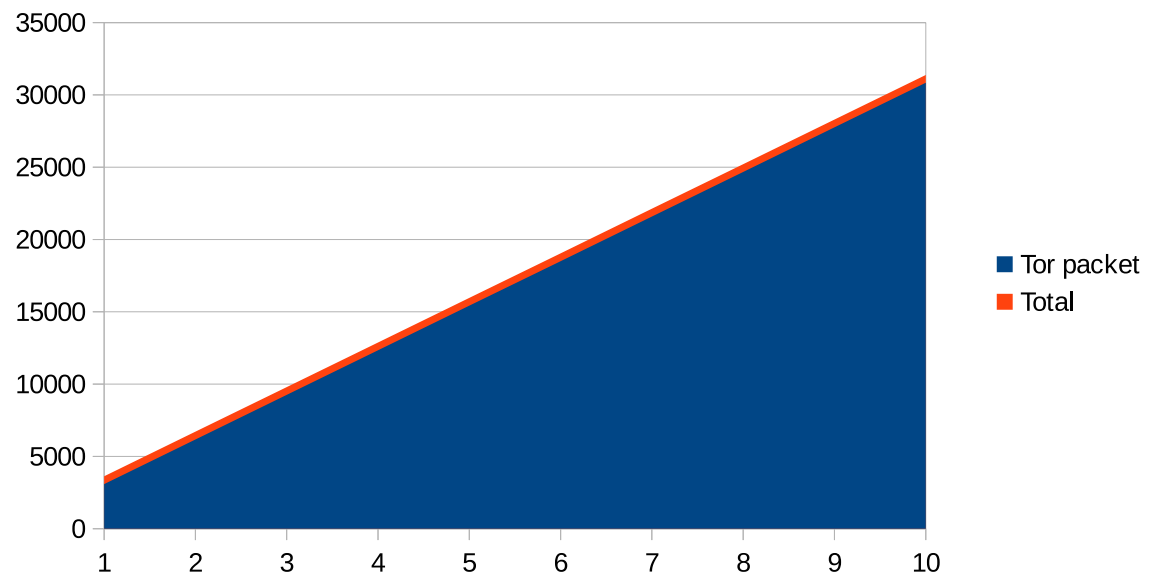
The TorCoin protocol does add a small amount of overhead to the Tor traffic. In our experimental setup, we set up a series of servers using the Python Twisted framework to simulate the passing of TorCoin generation and verification messages through a set of relays.

The total overhead from one round of successful TorCoin mining (i.e., one entire round trip from client through all the relays and back again) results in a total TorCoin packet overhead of 540 bytes. This can be broken down into:

- The initial three attempts by each relay: 42 bytes
- Verification message from Relay 3 to Relay 2: 74 bytes
- Verification message from Relay 3 to Relay 2: 138 bytes
- Verification message from Relay 3 to Relay 2: 202 bytes

Each round of TorCoin generation and verification happens only after m Tor packets have been sent. Each standard Tor cell is 514 bytes long, so each round trip on the network requires transmission of $514 * 6 = 3084$ bytes. Thus, if $m \geq 10$, the TorCoin protocol overhead is less than 2%. The value of m can be calibrated in further experimentation and as needed in order to achieve the sweet-spot of transmission efficiency and incentive maximization for relay providers.

Indeed, one can envision a tunable system where the value of m decreases during times of high usage to incentivise relay providers to temporarily provide more servers to the network.



While the Neff shuffle is complicated and requires a large number of communications between all of the servers, in practice, since the number of directory servers that will be involved in each shuffle will be relatively small (less than 10)

and are relatively fast servers with high-bandwidth connections with each other, this will not be a major bottleneck. In addition, since this is a one-time cost of connecting to the network, the users will be willing to wait for the slight time that it takes to setup the protocol, especially if the speeds increase sufficiently.

5 Related Work

PAR [1], XPay [2], Gold Star [3], BRAIDS [4], Tortoise [5], LIRA, onions for sale [7].

On the economics of anonymity [10], one-to-n scrip systems [11].

LIRA [6] is a lightweight system providing performance incentives for users to contribute bandwidth to the Tor network. It uses coins, similar to in game currency, to distribute payment. LIRA uses coins which have a tunable probability of being right, and clients can guess lottery tickets with probability, p of being right.

Eigenspeed [12] is a peer-to-peer consensus building algorithm for monitoring bandwidth over a network, specifically implemented for Tor. Unfortunately it requires a central authority for computing Principal Component Analysis operations. While we believe these operations could be decentralized, we are not interested in extending Eigenspeed. Instead, we exploit properties of the Bitcoin protocol to allow for bandwidth monitoring that is sufficient to generate payment tickets. However, as mentioned before, Eigenspeed could act as a secondary system to monitor relays and clients that seem to be producing more TorCoins than would be warranted by their speed.

6 Future Work and Conclusions

References

1. Androulaki, E., Raykova, M., Srivatsan, S., Stavrou, A., Bellovin, S.M.: PAR: Payment for anonymous routing. In Borisov, N., Goldberg, I., eds.: Privacy Enhancing Technologies: 8th International Symposium, PETS 2008, Leuven, Belgium, Springer-Verlag, LNCS 5134 (July 2008) 219–236
2. Chen, Y., Sion, R., Carbunar, B.: XPay: Practical anonymous payments for Tor routing and other networked services. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2009), ACM (November 2009)
3. Ngan, T.W.J., Dingledine, R., Wallach, D.S.: Building incentives into Tor. In Sion, R., ed.: Proceedings of Financial Cryptography (FC '10). (January 2010)
4. Jansen, R., Hopper, N., Kim, Y.: Recruiting new Tor relays with BRAIDS. In Keromytis, A.D., Shmatikov, V., eds.: Proceedings of the 2010 ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4–8, 2010, ACM (2010)
5. Moore, W.B., Wacek, C., Sherr, M.: Exploring the potential benefits of expanded rate limiting in tor: Slow and steady wins the race with tortoise. In: Proceedings of 2011 Annual Computer Security Applications Conference (ACSAC'11), Orlando, FL, USA. (December 2011)
6. Jansen, R., Johnson, A., Syverson, P.: LIRA: Lightweight Incentivized Routing for Anonymity. In: Proceedings of the Network and Distributed System Security Symposium - NDSS'13, Internet Society (February 2013)

7. Johnson, A., Jansen, R., Syverson, P.: Onions for sale: Putting privacy on the market. In: Financial Cryptography and Data Security. Springer (2013) 399–400
8. Dovrolis, C., Ramanathan, P.: A case for relative differentiated services and the proportional differentiation model. Network, IEEE **13**(5) (1999) 26–34
9. Dovrolis, C., Stiliadis, D., Ramanathan, P.: Proportional differentiated services: Delay differentiation and packet scheduling. IEEE/ACM Transactions on Networking (TON) **10**(1) (2002) 12–26
10. Acquisti, A., Dingledine, R., Syverson, P.: On the economics of anonymity. In: Financial Cryptography. Springer-Verlag, LNCS 2742. (2003) 84–102
11. Humbert, M., Manshaei, M., Hubaux, J.P.: One-to-n scrip systems for cooperative privacy-enhancing technologies. In: Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing. (2011)
12. Snader, R., Borisov, N.: Eigenspeed: secure peer-to-peer bandwidth evaluation. In: IPTPS. (2009) 9

DRAFT - April 19, 2014