Strategic Latency Reduction in Blockchain Peer-to-Peer Networks

Weizhao Tang
Carnegie Mellon Univeristy & IC3
USA
wtang2@andrew.cmu.edu

Lucianna Kiffer
ETH Zurich
Switzerland
lkiffer@ethz.ch

Giulia Fanti
Carnegie Mellon Univeristy & IC3
USA
gfanti@andrew.cmu.edu

Ari Juels*
Cornell Tech, IC3, & Chainlink Labs
USA
juels@cornell.edu

CCS CONCEPTS

• **Security and privacy** → *Network security*; • **Networks** → *Network measurement*; **Network algorithms**.

KEYWORDS

blockchains, P2P networks, strategic manipulation

ACM Reference Format:
Weizhao Tang, Lucianna Kiffer, Giulia Fanti, and Ari Juels*. 2023. Strategic Latency Reduction in Blockchain Peer-to-Peer Networks. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '23 Abstracts), June 19–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3578338.3593572

1 INTRODUCTION

[1] Most permissionless blockchains today run on peer-to-peer (P2P) communication networks due to their flexibility and distributed nature. These benefits of P2P networks typically come at the expense of network performance, particularly the latency of message delivery. Historically, this has not been a bottleneck because most permissionless blockchains are currently performance-limited not by network latency, but by the rate of block production and transaction confirmation, both of which are superlinear in network latency and dictated by the underlying consensus mechanism. For this reason, network latency has not traditionally been a first-order concern in blockchain P2P networks, except where substantial latency on the order of many seconds or even minutes raises the risk of forking.

Emerging concerns in blockchain systems, however, are beginning to highlight the importance of *variations in fine-grained network latency*—e.g., on the order of milliseconds—among nodes in blockchain P2P networks. These concerns are largely independent of the underlying consensus mechanism. They revolve instead around strategic behaviors that arise particularly in smart-contract-enabled blockchains, such as Ethereum, where decentralized finance (DeFi) applications create timing-based opportunities for financial gain. Key examples of such opportunities include:

- **Arbitrage:** Many blockchain systems offer highly profitable opportunities for *arbitrage*, in which assets are strategically sold and bought at different prices on different markets (or at different times) to take advantage of price differences. Strategic agents performing arbitrage can obtain an important advantage through low latency access to blockchain transactions.
- **Strategic transaction ordering:** In many public blockchains, strategic agents (who do not validate blocks) can profit by ordering their transactions in ways that exploit other users—a phenomenon referred to as *Miner-Extractable Value* (MEV) [2]. Small network latency reductions can allow an adversary to observe victims' transactions before competing strategic agents do.
- **Improved block composition:** Miners (or validators) rely on P2P networks to observe user transactions, which they then include in the blocks they produce. Which transactions a miner includes in a block determines the fees it receives and thus its profits. Therefore the lower the latency a miner experiences in obtaining new transactions, the higher its potential profit. [2]
- **Targeted attacks:** The security of nodes in blockchain P2P networks may be weakened or compromised if adversaries discover their IP addresses, e.g., through denial-of-service (DoS) attacks.

In order to reduce latency in practice, nodes may rely on proprietary, *cut-through* networks such as bloXroute [1] to learn transactions or blocks quickly. An agent in a blockchain P2P network, however, can also act strategically by means of *local actions*, namely choosing which peers a node under its control connects to.

Peri: A recent protocol called Perigee [3] leverages agents' ability in blockchain P2P networks to control their peering to achieve network-wide latency improvements. In the Perigee protocol, every node assigns each of its peers a latency score and periodically tears down connections to peers with high scores. Over time, Perigee causes nodes to remain connected to low-latency peers, while replacing other peers with random new ones. These sort of strategies can be adopted by *individual strategic agents*. We adopt a set of latency-reduction strategies called ***Peri***[3] that modify Perigee with

[*] The author contributed to this work in his Chainlink Labs role.
[1] The full paper of this abstract can be found at https://doi.org/10.1145/3589976.

[2] Receiving blocks from other miners quickly is also beneficial. To reduce forking risks, however, miners or validators often bypass P2P networks and instead use cut-through networks to communicate blocks to one another.
[3] A mischievous winged spirit in Persian lore. We explicitly do not consider the design of Peri to be a main contribution, as the design is effectively the same as Perigee.

SIGMETRICS '23 Abstracts, June 19–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0074-3/23/06.
https://doi.org/10.1145/3578338.3593572

optimizations tailored for the individual-agent setting. Their main difference is in their goals – Perigee optimizes systemic network performance, while Peri advantages one node over other nodes.

We show that small differences in transaction latencies can help an adversary distinguish between paths to a victim—and ultimately even discover the victim's IP address. There is a strong incentive, therefore, for agents to minimize the latency they experience in P2P networks. **In this work, we initiate the study of strategic latency reduction in blockchain P2P networks**.

1.1 Types of network latency

In our investigations, we explore two types of latency that play a key role in blockchain P2P networks: *direct* and *triangular latency.*

Direct latency refers to the latency with which messages reach a listener node from one or more vantage points—in other words, source latency. We consider two variants:

- *Direct targeted latency* is the delay between a transaction being pushed onto the network by a *victim* node and an agent node receiving the transaction.
- *Direct global latency* node refers to targeted latency for an agent node averaged over all source victim nodes in the network.

Triangular latency is a second type of latency that corresponds to a node's ability to inject itself between a pair of communicating nodes. We consider two forms of triangular latency:

- *Triangular targeted latency* refers to the ability of an agent's node v to "shortcut" paths between a sender s and a receiver r. That is, suppose s creates a transaction m that is meant to reach r. This latency measures the difference between the delay on path $s \rightarrow v \rightarrow r$ and the smallest delay over all paths from $s \rightarrow r$ not involving v. Such a negative latency implies that v can inject a competing transaction m' to take m's place.
- *Triangular global latency* refers to the triangular relative targeted latency averaged over all source-destination pairs in the network.

2 MAIN RESULTS

We model the P2P network $\mathcal{N}$ as a (possibly weighted) graph $(\mathcal{V}, \mathcal{E})$. Edge weights represent the physical distance traversed by a packet traveling between a pair of nodes (e.g., this can be approximated in practice with traceroute). Each individual node $v \in \mathcal{V}$ can create a message m and broadcast it to the network. m traverses all the nodes in $\mathcal{V}$ following the network protocol, which allows an arbitrary node to forward m upon receipt to a subset of its neighbors. A participant of the P2P network has two classes of identifiers:

Class 1. **(Network ID)** An ID assigned uniquely to each of its nodes (such as enode ID in Ethereum P2P network).

Class 2. **(Logical ID)** An ID that is used to identify the creator or owner of a message (such as the public key of a wallet).

We assume a mapping $\texttt{NID} : \mathcal{W} \rightarrow \mathcal{V}$ exists from a logical ID to a network ID, where $\mathcal{V}$ and $\mathcal{W}$ denote the Network and Logical ID spaces, respectively.

We consider an adversary who aims to reduce latency in order to gain profit and/or threaten network security. It is capable of inserting one agent node a into the network, which can maintain at most k peer connections at a time. a can behave arbitrarily during relaying (for instance, blocking messages). Through a, the adversary observes the network traffic consisting of transactions, where each transaction has the logical ID of its sender. It is assumed to not know the mapping $\texttt{NID}$, and cannot peer with a transaction sender $v = \texttt{NID}(w)$ knowing only the logical ID w. Given the network ID v, however, the adversary a can peer with the target node.

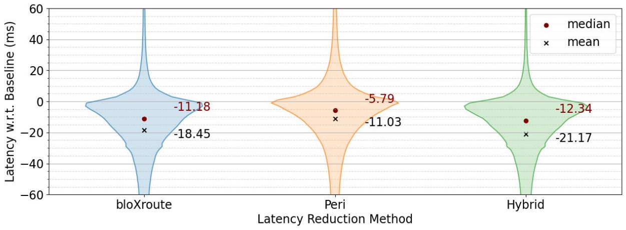

Figure 1: PDFs of global latency distributions (difference in ms from Baseline latency).

Direct Latency Reduction. Through experiments on the Ethereum mainnet, we compare the performances of Peri, bloXroute and hybrid (Peri + bloXroute) in reducing direct global latency. As Fig.1 shows, Peri achieves over half the reduction observed for bloXroute for both global and targeted latency, while hybrid provides a free 15% extra boost. We also show that Peri can discover the Network ID on the testnet of a victim node 7× more frequently than baseline approaches. Note that Peri achieves this performance with only **local** knowledge and control over the network.

Triangular Latency Reduction. To minimize triangular targeted latency, the agent only needs to peer with the source and destination, which reduces to the Network ID discovery problem. As for the triangular global latency, it is not interesting to reduce the aggregated latency directly, so we introduce a new proxy metric **advantage**, which captures the number of source-destination pairs that can be frontrun, given the adversary's choice on the agent node's peers. We prove the NP-hardness of finding the optimal peer set, but introduced a greedy algorithm that uses global network knowledge. We show by simulation that Peri performs as well as the greedy algorithm with only local information.

Impossiblity of strategy-proof peering protocols. Within a theoretical model, we prove that *strategy-proof P2P network design is fundamentally unachievable*: For any default peering algorithm used by nodes in a P2P network, as long as the network experiences natural churn and a target node is active, a strategic agent can always reduce direct targeted latency relative to agents following the default peering algorithm. Specifically, with probability at least $1 - \varepsilon$ for any $\varepsilon \in (0, 1)$, an agent can connect directly to a victim in time $O(\varepsilon^{-1} \log^2(\varepsilon^{-1}))$.

REFERENCES

[1] [Accessed Apr. 2022]. BloXroute website. https://bloxroute.com/. ([Accessed Apr. 2022]).

[2] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 910–927.

[3] Yifan Mao, Soubhik Deb, Shaileshh Bojja Venkatakrishnan, Sreeram Kannan, and Kannan Srinivasan. 2020. Perigee: Efficient peer-to-peer network design for blockchains. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*. 428–437.