Strategic Latency Reduction in Blockchain Peer-to-Peer Networks

IC3

Follow

The Initiative for CryptoCurrencies and Contracts (IC3)

--

Listen

Share

By Weizhao Tang, Lucianna Kiffer, Giulia Fanti, and Ari Juels

TL;DR

In traditional financial systems, time equals money and network latency

— i.e., the time for messages to travel in a network — has an outsized impact. Recently, network latency has become critical in blockchain peer-to-peer (P2P) networks as well. Among other impacts, low-latency connections in P2P networks can advantage arbitrageurs by giving them the ability to exploit the trades of other users for financial gain.

In this blog post, we summarize a recent paper of ours that explores P2P network latency. We present Peri, a practical strategy that selects peers with low latencies from a local view of the P2P network. We demonstrate how strategic agents

, i.e., self-interested P2P network actors, can use Peri to manipulate network latency to their advantage.

Our experiments suggest that Peri, which is cost-free, can in some situations achieve latency comparable to that of paid cut-through-network services such as bloXroute. Peri can alternatively boost the effectiveness of those services. Additionally, Peri can in some cases reveal the IP addresses of P2P targets that would otherwise be hidden, raising concerns about vulnerability against DoS attacks. In practice, such DoS concerns can be mitigated by frequently cycling IP addresses. Our work is a first step toward guidance around appropriate setting of cycling intervals.

Conversely — and unfortunately — we prove under a simple network model that in permissionless P2P networks, it is impossible to completely evade the strategies studied in our work.

Introduction

The introduction to Michael Lewis's Flash Boys

, a popular exposé on high-frequency trading on Wall Street, recounts thelaying of a fiber-optic cable from Chicago to the New York area (New Jersey). This cable shaved (one-way) communication latency down by a mere 1.5 milliseconds (ms) — from 8ms to 6.5ms — at the whopping cost of $300 million. Yet by the time Flash Boys

was published, a competing project was already underway: Microwave towers that would further reduce the latency tojust over 4ms.

In financial systems, speed matters. Time equals money, in the sense that traders can profit from fast generation and transmission of their transactions. What's true on Wall Street is also true in blockchain systems, as shown in research on [miner-extractable value

(MEV)](https://arxiv.org/abs/1904.05234). Arbitrage in blockchain systems — and MEV in general — are now widely recognized phenomena.

Largely overlooked in the study of MEV and blockchain arbitrage, however, is the kind of network-level competition discussed in Flash Boys

. Blockchains make critical use of peer-to-peer

(P2P) networks in order to broadcast transactions and blocks. A key question about these networks has remained unanswered: Is it possible through strategic manipulation of peer connections for a node to reduce its latency in receiving (and transmitting) transactions in blockchain P2P networks

?

Latency in P2P networks matters for several reasons. Low latency can advantage traders in on-chain trades. It can also help them with arbitrage between blockchains and off-chain systems. P2P-network latency also matters for security. As we'll explain, if an adversary can lower its latency to the extent of directly connecting to a victim's node, the adversary can learn the victim's IP address for use in denial-of-service (DoS) attacks.

Cut-through networks — including paid services such as bloXroute — offer fast connections within P2P networks analogous to the fiber-optic and microwave connections used in traditional finance. But a question still remains: Are there alternative, strategic behaviors by which nodes can reduce P2P latency without cost or can enhance the latency reduction obtained with use of cut-through networks?

Our research shows that there are. We also show that in theory, there is no way to design a P2P network that resists strategic manipulation.

Blockchain P2P networks

P2P networks in permissionless blockchain systems consist of a collection of nodes known, of course, as peers

. P2P networks are themselves permissionless, meaning that anyone can add or remove her own peer at any time. Each node is represented by what is called its enode ID

, which includes the node's IP address and TCP and UDP ports. A node has a number of slots

designated for connection to other peers in the network. (The Geth client for Ethereum has 50 slots by default.) When a node first joins the network, it discovers peers with open slots and randomly samples them to make connections and fill its own slots. (The Ethereum P2P network uses the Kademlia distributed hash-table (DHT) protocol for this purpose.) When slots open as a result of lost peers, a node will refill its slots in the same manner. Thus the global

topology of the network is determined by the local

actions of nodes.

Transactions are propagated through the P2P network via a broadcast or flooding protocol

. Upon receiving a transaction and verifying its validity, a node will transmit it to a random subset of its peers. Thus, when a node originates a transaction, i.e., introduces it into the network, it propagates throughout the network thanks to the help of peers.

Peri/gee

The randomized peer selections made by nodes in the manner described above can result in a network topology that is globally suboptimal, i.e., results in relatively slow transmission of transactions. Random graph topologies have good properties in terms of strong connectivity robust against connection failures, but can result in poor and heavily-skewed latencies between pairs of nodes. For example, nodes may end up connecting to peers in geographically distant places, resulting in slow transaction broadcast.

In view of this problem, Mao et al. introduced a schemein a 2020 paper to improve the systemic

performance of P2P networks, i.e., reduce latencies for all nodes. Their system, called Perigee

, employs a simple idea. In Perigee, nodes actively choose peers based on observed latencies

. When a node receives a message (in Perigee, a block) from a set of neighboring peers, it observes the time of receipt from each peer that sent it. The node then drops

peers that are slow to deliver the message, i.e., are on paths that impose relatively high latency. The node then refills its peer slots with randomly selected fresh peers. In this way, a node in Perigee retains connections to neighbors that provide low latency (are well connected), while exploring connections to new neighbors that might further improve its connectivity.

Perigee significantly improves (block) broadcast in a P2P network, reducing latency in the authors' simulations by some 33%. Despite its benefits, however, Perigee isn't currently used in P2P networks.

Given that Perigee isn't in current use, our work explores a variant of Perigee with a rather different goal: for a node unilaterally to gain a latency advantage over other nodes

. We introduce a scheme for this purpose called Peri[1]

. Peri is an algorithm in which a single, strategic

node adopts a Perigee-like algorithm, dropping connections to peers that evidence relatively high latencies. (Peri also has a

number of technical differences from Perigee, including a focus on transactions of strategic relevance, blocklisting of previously seen peers, etc.)

[1] Peri is a mythological, mischievous winged creature in Persian lore. The name Peri also, of course, pays homage to Perigee.

Flavors of latency

One important facet of our work is an exploration (and formalization) of the different types of latency potentially relevant to an arbitrageur (or other strategic network participant). Let's refer to our arbitrageur as Alice. Often Alice is interested in the transactions coming from a particular node

in the P2P network — let's refer to its owner as Bob. Bob's node may be broadcasting easy-to-arbitrage transactions or acting as a broadcast point for many transactions — e.g., an Infura node. We refer to this case as targeted latency

. Also of interest is global latency

, meaning the latency from nodes around the network as a whole.

Latency can also be measured in terms of the time it takes for Alice to receive a transaction from Bob. We call this direct

latency. But if Alice is an arbitrageur, she may be interested in front-running

Bob's transactions. In other words, when she sees a transaction T_B from Bob, her goal may be to get her own transaction T_A to the node of a miner before Bob's. (For example, suppose Bob's transaction T_B buys a particular token, Fabulous Wealth Token (FWT), Alice knows the price will rise as a result and may want to slip her own buy order T_A for FWT first.) We call this triangular latency

.

Since it's hard to measure triangular latency, our experimental focus has been on direct latency.

How much does Peri reduce latency?

In experiments with Peri, we first examined direct global latencies. We conducted a series of experiments lasting about a month (from 18 February to 16 March 2022), in which we analyzed the latencies of over 6 million transactions. We compared Peri with a "baseline" (ordinary Geth) node, as well as with a "bloXroute" node using bloXroute's $300 / month Professional Plan, and a "hybrid" node using both Peri and bloXroute. Compared with the baseline node, the bloXroute node experienced transaction-receipt latencies (direct, global latencies) reduced by 18.45 ms on average. Peri achieved a reduction of 11.09 ms on average. The best was the hybrid node, which further reduced latency over the bloXroute node by 15%. The upshot is that in our experiments, Peri achieves at no cost latency reductions half as good as bloXroute's, and can also serve to boost bloXroute's performance.

We also studied direct targeted latency. Setting up our own node and sending 6 million transactions would have cost more than $12 million. So we instead conducted a limited experiment in which we took as our victim (Bob in the above example) the node of the Chainlink operator Chainlayer. (Chainlayer graciously identified to us the transactions it broadcast on the network.) We found that in this case, Peri achieved latency reductions statistically equivalent to those of the bloXroute and hybrid nodes — about a latency reduction of 14% of the end-to-end latency measured by ping (roughly 11ms).

Finding victims with Peri

Peri is not just relevant for reducing latency–it can also be used in some cases to connect to (or discover the IP address of) a node based only on the operator's public cryptocurrency address. This can have serious security implications; if an attacker discovers a node's IP address, they can launch denial-of-service attacks, eclipse a node, or censor their transactions.

To quantify this risk, we ran additional experiments evaluating how often Peri connects directly to a victim node compared to baseline peering strategies. In experiments on the Ethereum Rinkeby testnet, we set up a victim node that broadcasted

transactions 3 times per minute. We discovered thatPeri found the victim 67.1% of the time, which is 7 times as frequent as the default node.

However,

these results appear to depend on the frequency with which the victim node broadcasts transactions. For example, we also tried to learn the IP address of our Chainlink operator node, which transacts about once an hour. In these experiments, Peri found the victim only 5% of the time, which was comparable with the baseline node. A key takeaway is that nodes that transact more frequently are more vulnerable to IP identification attacks

. This risk can be mitigated by changing IP addresses periodically, though such a solution may not be practical for nodes who need a stable connection.

Impossibility result

These results raise an important question: can we design a P2P network that is robust to strategic latency manipulation attacks? Our results suggest that the answer is no. Under a stylized model of network dynamics, we analyze the time it takes for an attacker Alice to peer with a victim Bob. We find that to peer with Bob (or any other target) with probability 1-ε, Alice needs time at most quasilinear in ε^{-1}. In that sense, it is impossible to design a fully strategy-proof peering protocol in unstructured blockchain P2P networks.

Conclusion

So what can we do as system designers? Since our results suggest that “fixing” the P2P layer is impossible, more research is needed to ensure that strategic P2P latency manipulations do not adversely affect fairness or privacy at the application layer.

For now, it is useful for node operators to understand both the risks and the opportunities associated with strategic peering. For example, our results show that Peri is a powerful peering strategy for reducing network latency. At the same time, an attacker can use Peri to discover IP addresses of nodes, and potentially launch a DDoS attack. This risk can be mitigated by periodically refreshing a node’s IP address or by broadcasting transactions from multiple nodes in different locations. Further research is needed to understand how effective these measures are, and if there exist more powerful alternatives to Peri.

For more details, please check out our [paper

](https://arxiv.org/abs/2205.06837).

Acknowledgements

We wish to thank Lorenz Breidenbach from Chainlink Labs and Peter van Mourik from Chainlayer for their gracious help in the setup for our direct-targeted-latency experiments.

This material is based upon work by Giulia Fanti supported by the Air Force Office of Scientific Research under award number FA9550–21–1–0090, Chainlink Labs, and IC3.

Ari Juels contributed to this project in his role with Chainlink Labs. Lucianna Kiffer contributed to this work under a Facebook Graduate Fellowship.