

On Designing Performant Peer-Peer Overlay Networks for Web 3.0

Arunesh Mishra and Adi Kancherla
Piccolo Labs

Abstract—Many experts believe that decentralized applications will be the core paradigm for Web 3.0. To become real, these applications need to run on a fast network. Since most networks provide an open participation model, the system doesn't have control over how the network topology grows. In this work, we conduct an empirical analysis of the network layer that supports decentralized applications and blockchains. We present a set of design principles that p2p overlay networks should adhere to in general. The talk will also discuss empirical results from data collected on IPFS, Ethereum and a couple of other blockchain projects to compare and contrast their design and performance. Our goal is to foster an open discussion so that the network layer can scale the applications that run on top of it.

I. INTRODUCTION

WE are in the midst of building a new Internet. Blockchain networks like Ethereum have popularized the idea of fully decentralized, owner less applications that are run on an open network of untrusted but incentivized nodes without the oversight of a central authority. There are many other projects such as Dfinity, ThunderCore, Golem, etc who are creating the *world computer*. This radical computing platform has the potential to solve many of the issues around data privacy and the paradox of *user-trust* in an attention economy [1] incentivized to sell eye-ball time to advertisers.

In order to realize these benefits, decentralization requires an open participation model, that is, anybody in general should be allowed to participate on the network. This brings forth new and exciting technical challenges for the Academic and Research community to tackle. One such challenge is the need for a fast network that connects the participating nodes, miners or users and their devices. This p2p overlay network [2] is a building block for blockchain protocols or core blockchain operations, such as consensus, sharding, replication, ledgers etc. All of the communication messages use the overlay network and thus, it becomes important to understand how such a network should be designed.

This short writeup discusses ongoing work as part of the Piccolo Project ¹. The goal of the Piccolo project is to build a fast decentralized open data network to support blockchain applications by offloading the storage and querying of structured data to a separate fast network. As a part of this effort, we are evaluating the network layer designs used by existing projects in the blockchain space order to understand a few aspects, namely: (i) what are the typical overlay topologies (and their properties) for blockchain networks and (ii) what are the routing, lookup, message transfer latencies, costs and

inefficiencies if any. This work will help us design a better p2p overlay which is critical to support production quality decentralized applications if we are to make that future a reality.

The rest of this writeup is organized as follows. We present a brief summary of the related p2p overlay network layers in Section II. Subsequently in Section III we present an overview of our research methodology into existing p2p overlay designs that support popular blockchains and applications today. Finally, we conclude in Section IV.

II. RELATED WORK

There has been extensive research on p2p overlay networks over the last 15 years. Napster [3] was one of the first popular services that provided much of the original inspiration for p2p systems although its database was centralized. DNS is an example of a widely deployed distributed and largely decentralized key-value database that powers every lookup and interaction on the Internet [4]. DNS relies on special root servers to bootstrap the lookup protocol. The Freenet [5], [6] and the Gnutella [7] p2p systems were popular in the previous decade for file sharing. Both systems were designed for sharing of large files over a longer duration of time. Content reliability including lookup reliability and network latency goals were necessary in this environment.

The second generation of p2p systems include research driven projects such as Chord [8], Content Addressable Network (CAN) [9], Pastry [10], Tapestry [11] and Kademlia.

Other notable systems include Viceroy [12] which provides logarithmic hops through nodes with constant degree routing tables. SkipNet [13] uses a multidimensional skip-list data structure to support overlay routing, maintaining both a DNS-based namespace for operational locality and a randomized namespace for network locality. Other overlay proposals such as Koorde [14] and Naor et al [15] attain lower bounds on local routing state but oversimplify some of the other features.

The third generation of p2p research includes building applications on top of these DHT systems, validating them as novel infrastructures or tuning them for specific use cases. For example, applications such as PAST [16] and SCRIBE [17] are built on top of Pastry. Decentralized file storage application project OceanStore [18] was built on top of Tapestry, while CFS [19] was build on top of Chord. FarSite [20] uses a conventional distributed directory service and could be built on top of Pastry. Another example of an overlay network is the Overcast System [21], which provides reliable single-source multicast streams.

¹See <https://piccolo.network/paper> for a detailed discussion of the Piccolo network and database layer.

Given the wide diversity of research in the p2p overlay community, it is a bit surprising that the blockchain community has largely only used one or two variants of the choices available. IPFS uses Kademlia [22] which is implemented as a part of *libp2p* project. Many other network layers have made this choice, while some resort to a more simplistic gossip-protocol style designs.

III. RESEARCH METHODOLOGY

The goal of our study is to understand the architecture of some of the popular blockchain networks, such as Ethereum (Whisper protocol), IPFS, Neo, EOS alongwith some of the emerging projects in this space. Our approach includes documenting their overlay network properties according to the features listed the following subsection. Further we have built a "network crawler" utility, similar to traceroute, that can crawl the layer 4 topology of these networks to further understand them through certain metrics (discussed below).

Overall, we find that most blockchain networks and decentralized projects pay very little attention to the design details for the p2p overlay layer. This is expected as the majority of the focus has been on speed, fast consensus and smart contract execution to name a few challenges. However, a well designed network layer has demonstrated improvements at the application layer metrics in past work [11], [16], [18], [20]. We will present results from these various networks alongside insights on any bottlenecks and possible improvements both including short-term fixes and long-term architectural improvements. In the next subsection, we discuss some of the aspects of the routing layer that are relevant to decentralized applications in general.

A. P2p Overlay Design principles

Some of the interesting and relevant aspects of a p2p overlay routing system would include:

- *Location independent routing*: Location independent routing refers to a class of techniques for locating objects based on content rather than their location, while attempting to find the shortest path possible to reach such objects. For example, in IPFS [23] a hash of the block is used to identify and refer to the content. The hash is also used to locate the content using the Kademlia routing protocol. Depending on the application, such as content lookup versus synchronizing across consensus shards, this is a
- *Deterministic node mapping*: It should be possible to route messages or lookup objects and services in the network regardless of where the lookup originates. The result of the distributed routing algorithm should be deterministic.
- *Overlay topology quality metrics*: Overlay metrics such as routing stretch and stress help understand how inefficient a route is. For example, the stress metric helps understand the amount of duplication on a single network link as a result of inefficiencies in the overlay routing. Other metrics such as network diameter and graph topology metrics help understand how the network

would behave in adverse scenarios, such as likelihood of a partition.

- *Load balancing*: It might be possible to leverage the network layer for implicit load balancing by using a set of nodes as representatives for a given content addressable object. This might largely be a function of the service that the network layer supports but might be critical to the overall application as well. For instance, a file system can use a network layer that can easily balance load for a popular object.
- *Dynamic membership*: How the routing layer adapts to churn. Nodes are expected to disappear without notice in an open network environment. The service layer and the routing layer has to include redundancy in order to gracefully cope with such failures. The probability of key nodes disappearing which can lead to network partition should be minimized. In our study, we look at the possibility of network partitions as a way to understand robustness in design.
- *Tolerance to Byzantine behavior*: Can the network can tolerate malicious and intentional disruption. While correlated failure modeling can provide some approximation to intentional disruption, this attack vector is something that has not been studied in previous p2p network research. Most overlay protocols are not evaluated from this perspective of denial of service attacks, in essence, understanding what type of attack would need to be mounted to bring the network down (or cause a partition) and whether such an attack can be detected (or mitigated).

B. Data collection

Similar to how traceroute works on the Internet by sending pings with sequentially increasing TTL values, one can create a similar "route crawling" algorithm that can compute p2p overlay topologies for blockchain networks. For example, with a prefix-based routing scheme like Kademlia, a crawler can send route requests to a set of carefully determined content or host IDs which can help it maintain an updated view of the network. This data can be combined in real-time with multiple such observers in the network. Note that the physical location (layer 3) of these crawlers has no significance on the measurement outcome, essentially a single machine can run a few hundred of these crawlers that can be uniformly distributed in the routing space.

We will present results from crawling the top blockchain networks. For some of the upcoming chains, we will also present results from their respective testnets which we have been provided access to for research purposes.

IV. CONCLUSION

The overall goal of our research is to cultivate an understanding in the blockchain community for the importance of p2p overlay network design. We do this in two steps: (i) The first step is to present an empirical study of existing networks, their performance and understanding of their limitations and bottlenecks if any, (ii) bring forth lessons from extensive research done in the past decade to the blockchain

network design in light of the empirical study. This will help researchers and blockchain engineers design a better network backbone to support fully decentralized applications.

REFERENCES

- [1] P. van Niekerk, "Can platforms ever be trusted in an attention economy?" <https://www.fipp.com/news/features/can-platforms-ever-trusted-attention-economy>.
- [2] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE COMMUNICATIONS SURVEY AND TUTORIAL*, 2004.
- [3] "Napster," <https://en.wikipedia.org/wiki/Napster>.
- [4] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," in *Symposium Proceedings on Communications Architectures and Protocols*, ser. SIGCOMM '88. ACM, 1988.
- [5] I. Clarke, "A distributed decentralised information storage and retrieval system," *Master's thesis, Univ. of Edinburgh*, 1999.
- [6] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*. Springer-Verlag, 2001.
- [7] "Gnutella p2p network," <https://en.wikipedia.org/wiki/Gnutella>.
- [8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. ACM, 2001.
- [9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. ACM, 2001.
- [10] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*. Springer-Verlag, 2001.
- [11] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, Jan 2004.
- [12] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proceedings of the Twenty-first Annual Symposium on Principles of Distributed Computing*, ser. PODC '02. ACM, 2002.
- [13] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "Skipnet: A scalable overlay network with practical locality properties," in *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, ser. USITS'03. USENIX Association, 2003.
- [14] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Peer-to-Peer Systems II*. Springer Berlin Heidelberg, 2003.
- [15] M. Naor and U. Wieder, "A simple fault tolerant distributed hash table," in *Peer-to-Peer Systems II*. Springer Berlin Heidelberg, 2003.
- [16] P. Druschel and A. Rowstron, "Past: a large-scale, persistent peer-to-peer storage utility," in *Proceedings Eighth Workshop on Hot Topics in Operating Systems*, 2001.
- [17] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel, "Scribe: The design of a large-scale event notification infrastructure," in *Networked Group Communication*, J. Crowcroft and M. Hofmann, Eds. Springer Berlin Heidelberg, 2001.
- [18] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An architecture for global-scale persistent storage," *SIGPLAN Not.*, 2000.
- [19] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with cfs," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, ser. SOSP '01. ACM, 2001.
- [20] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer, "Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs," in *Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '00. ACM, 2000.
- [21] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr., "Overcast: Reliable multicasting with an overlay network," in *Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation - Volume 4*, ser. OSDI'00. USENIX Association, 2000.
- [22] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. Springer-Verlag, 2002.
- [23] J. Benet, "Ipfes - content addressed, versioned, p2p file system," <https://ipfs.io/ipfs>.

Arunesh Mishra: Dr. Mishra is an Entrepreneur and a Researcher in the area of Distributed Systems, Networking and Security. He has published in top ACM and IEEE Conferences including a best paper award at ACM Mobicom. More at <https://arunesh.github.io>

Adi Kancherla: Mr. Kancherla is an ex-Google engineer with a graduate degree from University of Wisconsin, Madison. He has experience in building scalable database backends to support high availability, resilience in the presence of failures and real-time performance.