# How to design a Fast Peer-Peer Overlay Network

Arunesh Mishra and Adi Kancherla
*Picolo Labs*

***Abstract*—**

## 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.

They have further evolved into a computing and decentralized app (dapp) platform [1]. This radical computing paradigm has the potential to become the new Internet with both data and compute being fully decentralized.

Decentralization has a lot of benefits [] which requires an open participation model, that is, anybody in general should be allowed to participate on the network. This brings forth new exciting technical challenges for the Academic and research community to tackle. One such challenge to consider is the need for a fast network that connects the participating nodes, miners or users and their devices. The p2p overlay network [2] is a building block for all of the protocol or blockchains' operations, such as consensus, sharding, replication, ledgers etc. All of the messages use the overlay network and thus, it becomes important to understand how such a network should be designed.

This is an short summary of the resear

## 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 reliabilty including lookup reliability and network latency goals were necessary in this enviroment.

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 namesapce 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.

## 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, Bitcoin 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

Arunesh Mishra: https://arunesh.github.io/

possible to reach such objects. For example, in IPFS [22] a hash of the block is used to identify and refer to the content. The hash is also use 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.
- *Dynamic membership:* How the routing layer adapts to churn.
- *Tolerance to Byzantine behavior:* Whether the network can tolerate malicious and intentional disruption. This is something that has not been studied in previous p2p network research.

[23]

## IV. Summary of Contributions

## V. Conclusion

The conclusion goes here.

## Appendix A
## Proof of the First Zonklar Equation

Appendix one text goes here.

## Appendix B

Appendix two text goes here.

## Acknowledgment

The authors would like to thank...

## References

[1] A. Reyes, "Ethereum – the world computer," https://medium.com/@reyesale/ethereum-the-world-computer-fb7b58948280.
[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 on overlay network," in *Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation - Volume 4*, ser. OSDI'00. USENIX Association, 2000.
[22] J. Benet, "Ipfs - content addressed, versioned, p2p file system," https://ipfs.io/ipfs.
[23] C. Gentry, "A fully homomorphic encryption scheme," http://cs.au.dk/~stm/local-cache/gentry-thesis.pdf.

**Arunesh Mishra** Dr. Arunesh Mishra is an Entrepreneur and a Researcher in the area of Distributed Systems, Networking and Security. He has a PhD in the area of Wireless networking and Security which includes 30+ papers in top ACM and IEEE Conferences including a best paper award at ACM Mobicom. He has written proposals for NSF grants which have been awarded and also holds 22+ patents (20 pending). He built Googles location computation

algorithms that serve 3 Billion users every day by self-learning from crowd-sourced data. He created the tech stack for 4 product teams during his 10 year tenure at Google including a demo at Google I/O.

**Adi Kancherla**