Next-Generation Distributed Hash Tables

SAIDU SOKOTO, City, University of London, United Kingdom
MICHAŁ KRÓL, City, University of London, United Kingdom
VLADIMIR STANKOVIC, City, University of London, United Kingdom
ETIENNE RIVIÈRE, UCLouvain, Belgium

Distributed Hash Tables (DHTs) serve as the backbone of numerous modern decentralized systems like the InterPlanetary File System (IPFS) and Ethereum. As these systems evolve and expand, there is a growing need to enhance and optimize their underlying network support. In response to these challenges, we embark on the development of a new class of DHTs, marked by efficiency, security, and suitability for real-world deployments. We achieve this by making changes in the routing procedures, incorporating latency-aware routing, and harnessing recent hardware advancements.

1 INTRODUCTION

Over the past two decades, numerous Distributed Hash Tables (DHTs) have been proposed [13]. Today, they serve as a fundamental component in numerous decentralized systems such as the InterPlanetary File System (IPFS), Ethereum, and Libp2p [3]. Libp2p is of particular interest as it provides a versatile networking stack that harnesses the Kademlia DHT [9] for peer discovery and routing. The library is a de facto choice used for building peer-to-peer (P2P) applications [2]. Critical operations such as peer discovery and content fetching performed by DHTs underscore an urgent need for security and efficiency enhancements.

With the ever-growing landscape of Web3 and decentralized systems, a solution capable of addressing vulnerabilities inherent in current DHT protocols [12] will enable decentralized systems to meet the evolving demands of the users and applications that rely on them. For instance, emerging use cases such as Data Availability Sampling (DAS) in Ethereum not only require secure P2P networks but also impose stringent timing constraints, e.g. the completion of queries within a 4 second window [8].

2 PROBLEM STATEMENT

Despite the changing requirements of decentralized systems, DHTs have shown limited evolution, leading to two persistent problems: *(i)* efficiency and *(ii)* security. Lookup operations often take a couple of seconds - a sharp contrast from sub-second, cloud-based content fetching [11]. This variance can be attributed to the delay encountered in traversing the DHT. In terms of security, numerous attacks persist, including: eclipse attacks, whereby benign nodes are shielded from the real network; route poisoning attacks, where nodes erroneously direct traffic to wrong or malicious nodes [12]; and content censorship attacks using Sybil identities [10].

Existing solutions focus on one property in isolation without thoroughly investigating the impact on the other. For instance, network coordinate systems like Vivaldi [6] prioritize performance enhancements but may compromise on security. Nodes may simply provide false information about their coordinate, thus facilitating eclipse attacks. Conversely, approaches such as S/Kademlia [4] focus on improving security but may incur performance penalties through the introduction of intensive cryptographic operations, such as proof of work to counter Sybil attacks. Additionally, some prior work rely on assumptions that may not align with real-world conditions, such as assuming the existence of a social graph [7], that was never deployed by any large-scale system.

These observations underscore the complexities of balancing performance, security, and practicality. Evidently, a holistic and practical solution to overcoming these challenges effectively in the presence of millions of nodes is needed. Achieving this will represent a key step towards achieving decentralization, censorship resistance, and an improved user experience.

3 PROPOSAL

First, we measure existing decentralized systems (e.g., IPFS and Ethereum) to understand how DHTs are used, identify their weaknesses and existing data flows. The results will drive the design of our DHT and act as realistic workloads for future evaluation.

We develop a next-generation DHT by modifying the core protocol. This includes changes in how nodes interact and find new peers, along with efficient node replacement policies in the routing table. We focus on secure network coordinates to enable latency awareness routing [5]. Furthermore, we implement new query types directly supporting emerging use cases such as DAS [8].

We also harness recent hardware advancements, such as increased memory and trusted execution environments (TEE). For instance, increasing the number of peers stored in the routing table can significantly speed up the routing. Assuming a 30 bytes record size, the memory required to store all 15,000 Ethereum nodes is only 439 KB. Furthermore, we investigate how the presence of a few semi-trusted nodes running TEE can mitigate the risk of DHT attacks without additional overhead.

We perform simulations to evaluate the proposed DHT's performance under controlled conditions and large-scale experiments on a testbed such as Grid5k.[1] Furthermore, our ongoing consultations with industry partners will offer insight into practical aspects of the protocol and its potential deployment. Mitigation of potential vulnerabilities along with rigorous security analysis will determine the project's success in providing security improvements.

4 PRELIMINARY RESULTS

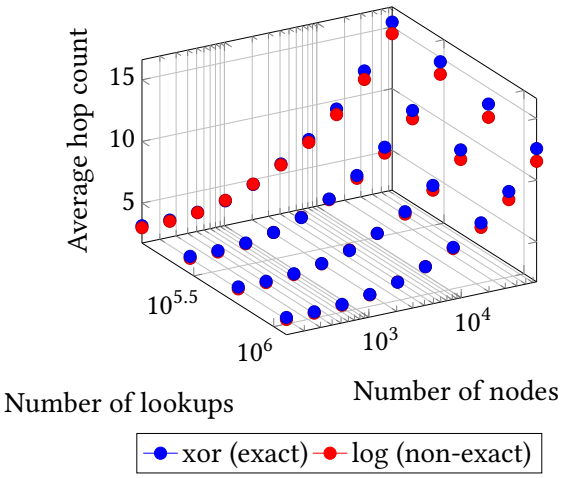
Our research involves investigating modifications implemented by industrial systems, such as Ethereum and IPFS to inform our design. To this end, we have conducted a comprehensive comparison between the original Kademlia [9], the most widely used DHT, and its Ethereum implementation [1].

The main difference between these two implementations lies in their routing strategies. Ethereum's implementation iterates through the k-buckets (used to organize routing tables), gathering neighbor candidates based on their logarithmic distances to the key (against the xor distance used in the vanilla Kademlia). Interestingly, this key itself is not provided as part of the query. This modification enhances security by making it more challenging to execute Eclipse attacks, but its performance impact was unknown.

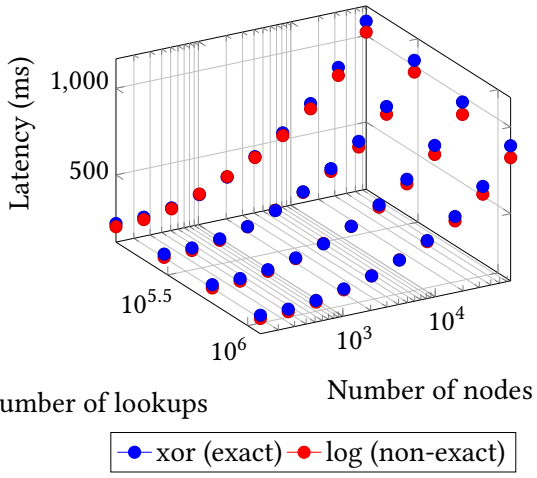
To assess the impact of these routing strategy differences, we simulated both implementations on a Peersim based event-driven simulator,[2] focusing on key performance metrics such as latency and overlay hop count. We also compared them over a range of lookups ranging from 150,000 to 1,200,000 lookup operations and considered various node counts: from 128 to 65,536, as illustrated in the 3D plot in Figure 1. Throughout this evaluation, we focused on the protocols' fundamental aspects and avoided unnecessary complexities that add no value to their functionality.

Across these metrics, we observed a consistent trend with the Ethereum implementation outperforming the vanilla Kademlia as the number of nodes and lookups increased. However, this difference was not drastic, as both latency and

[1] https://www.grid5000.fr/w/Grid5000:Home
[2] Available at https://github.com/datahop/kademlia-simulator.



(a) Hop count.



(b) Latency comparison.

Fig. 1. Relationship between hops and latency for different numbers of lookups and nodes under two lookup methods.

hop count remained within 8% of each other. These findings suggest that specific changes in the lookup process can improve both security and performance. This holds significant potential and can be used to advocate for the adoption of these Ethereum-specific changes in other systems.

5 CONCLUSION AND FUTURE RESEARCH

Our preliminary experimental results shed light on alternative Kademlia lookup approaches. Even a seemingly simple change such as not disclosing a key during a lookup operation, can yield security and performance improvements. Moving forward, our next steps involve testing an architecture incorporating all the parts outlined in our proposal. This will leverage available and ongoing measurement results within a simulation environment. We also plan to conduct real-world testing while actively collaborating with industrial partners to ensure a seamless integration should the project

prove successful. Our multifaceted approach will help gauge success based on latency improvements against established benchmarks. In the spirit of open-source research, we are committed to making all our datasets, implementations, and artifacts readily accessible to the scientific community.

REFERENCES

[1] 2023. Discv5. GitHub. https://github.com/ethereum/devp2p/blob/master/discv5/discv5-wire.md

[2] 2023. IPFS. Online. https://ecosystem.ipfs.tech

[3] 2023. Libp2p. Online. https://libp2p.io

[4] Ingmar Baumgart and Sebastian Mies. 2007. S/Kademlia: A practicable approach towards secure key-based routing. In *2007 International Conference on Parallel and Distributed Systems*. 1–8.

[5] Miguel Castro, Peter Druschel, Y Charlie Hu, Antony Rowstron, et al. 2002. *Exploiting network proximity in peer-to-peer overlay networks*. Technical Report. Technical Report MSR-TR-2002-82, Microsoft Research.

[6] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. 2004. Vivaldi: A decentralized network coordinate system. *ACM SIGCOMM Computer Communication Review* 34, 4 (2004), 15–26.

[7] George Danezis, Chris Lesniewski-Laas, M Frans Kaashoek, and Ross Anderson. 2005. Sybil-resistant DHT routing. In *Computer Security–ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings 10*. Springer, 305–318.

[8] Michał Król, Onur Ascigil, Sergi Rene, Etienne Rivière, Matthieu Pigaglio, Kaleem Peeroo, Vladimir Stankovic, Ramin Sadre, and Felix Lange. 2023. Data Availability Sampling in Ethereum: Analysis of P2P Networking Requirements. *arXiv preprint arXiv:2306.11456* (2023).

[9] Petar Maymounkov and David Mazières. 2002. *Kademlia: A Peer-to-Peer Information System Based on the XOR Metric*. Lecture Notes in Computer Science, Vol. 2429. Springer Berlin Heidelberg, Berlin, Heidelberg, 53–65.

[10] Srivatsan Sridhar, Onur Ascigil, Navin Keizer, François Genon, Sébastien Pierre, Yiannis Psaras, Etienne Rivière, and Michał Król. 2023. Content Censorship in the InterPlanetary File System. *arXiv preprint arXiv:2307.12212* (2023).

[11] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. 2022. Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 739–752.

[12] Guido Urdaneta, Guillaume Pierre, and Maarten Van Steen. 2011. A survey of DHT security techniques. *Comput. Surveys* 43, 2 (Feb 2011), 8:1–8:49.

[13] Hao Zhang, Yonggang Wen, Haiyong Xie, Nenghai Yu, et al. 2013. *Distributed hash table: Theory, platforms and applications*. Springer.