Assessing the Performance of Ethereum and Hyperledger Fabric Under DDoS Attacks for Cyber-Physical Systems

Vijay Jayadev
University of Greenwich, London, UK
ka6327n@greenwich.ac.uk

Naghmeh Moradpoor*
Edinburgh Napier University, Edinburgh, UK
n.moradpoor@napier.ac.uk

Andrei Petrovski
Robert Gordon University, Aberdeen, UK
a.petrovski@rgu.ac.uk

ABSTRACT

Blockchain technology offers a decentralized and secure platform for addressing various challenges in smart cities and cyber-physical systems, including identity management, trust and transparency, and supply chain management. However, blockchains are susceptible to a variety of threats, akin to any other technological system. To assess the resilience and robustness of diverse blockchain technologies, this study evaluates their performance indicators under various attack scenarios. Therefore, this study conducts a thorough examination of multiple well-known blockchain technologies, such as Ethereum and Hyperledger Fabric, under Distributed Denial of Service attack scenarios. Ethereum, introduced as a revolutionary blockchain technology, has entirely transformed the way smart contracts and decentralized applications operate. Additionally, the innovative open source blockchain framework, Hyperledger Fabric, is intended for businesses and alliances seeking a secure and adaptable platform to develop distributed ledger applications. Hyperledger Besu, an Ethereum client with an extractable Ethereum Virtual Machine implementation designed to be enterprise-friendly for both public and private permissioned network use cases. Therefore, Ethereum and Hyperledger Fabric are utilized in this study for performance comparison. This study provides a summary of Ethereum's salient characteristics, architecture, and noteworthy influence on the blockchain and cryptocurrency ecosystem. Furthermore, it offers an overview of the main characteristics, architecture, and potential uses of Hyperledger Fabric. The blockchain's resilience against DDoS attacks is assessed by examining performance measures such as latency and throughput, which are fundamental metrics crucial for evaluating and enhancing the effectiveness of various systems, including communication protocols, databases, blockchains, and computer networks. The outcomes of these experiments show that Hyperledger Fabric has greater throughput and reduced latency, demonstrating its resistance to DDoS attacks in comparison with Ethereum. Ethereum, being a permissionless blockchain, can introduce challenges such as the potential for network congestion and scalability issues.

*Corresponding author.

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3670927

CCS CONCEPTS

• **Security and Privacy**; • **System Security**; • **Vulnerability Management**;

KEYWORDS

Ethereum Virtual Machine, Hyperledger Fabric, Distributed Denial of Service, Distributed Ledger Technology, Smart Contract

ACM Reference Format:
Vijay Jayadev, Naghmeh Moradpoor, and Andrei Petrovski. 2024. Assessing the Performance of Ethereum and Hyperledger Fabric Under DDoS Attacks for Cyber-Physical Systems. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3664476.3670927

1 INTRODUCTION

There are numerous Distributed Ledger Technology (DLT) systems currently in operation that have gained favour with users. Among them, blockchain stands out as the most well-known DLT, capable of addressing various challenges for smart cities and Cyber-Physical Systems (CPS), including data monetization and sharing, identity management, as well as trust and transparency. Blockchain tracks assets in a corporate network and enables the secure recording of transactions. The technology's name—block and chain—defines its structure. Each block in the sequence has two values: the hash of the preceding block and its own hash. This structure keeps blockchain tamper-proof since data is updated chronologically, and when data is updated, the hash value is also updated, ensuring the veracity of the data. Blockchain technology has grown in popularity in recent years due to its decentralized and secure nature. It is a game- changing technology that has attracted widespread attention and transformed numerous sectors. At its core, blockchain is a decentralized and distributed ledger system that enables multiple parties to maintain a shared database without the need for a central authority. This allows for the secure and transparent recording and verification of transactions, making it suitable for a wide range of applications. In a typical centralized system, such as a bank, a central authority controls and manages the transaction database. In a blockchain network, however, the ledger is distributed across numerous computers known as nodes that engage in a consensus mechanism to validate and agree on the ledger's status. Blockchain offers a game-changing technology that provides a decentralized and transparent method of recording and validating transactions [1].

However, while DLTs offer great features such as decentralization, transparency, immutability, efficiency, accessibility, and global reach, in most cases, the data held in the blocks has significant

value, and any security breach may result in significant reputational and/or financial damage. This includes various types of attacks, such as 51% attack, Sybil, Double Spending, Eclipse, Man-in-the-Middle, and Distributed Denial of Service (DDoS). Therefore, integrating security measures such as continuous monitoring and updates, as well as adhering to best security practices, performance assessment, and early detection, can help mitigate this risk.

The primary objective of this paper is to conduct a performance assessment of two DLTs: Ethereum, a public and permissionless ledger, and Hyperledger Fabric, an open source and permissioned ledger, due to the lack of prior research on this topic.

The research question (RQ) of the paper is:

- What are the comparative performance characteristics of Ethereum and Hyperledger Fabric in relation to DDoS attacks?

To answer the RQ, we assess the performance of Ethereum and Hyperledger Fabric, including throughput and latency, while under DDoS attacks using Hyperledger Caliper. The specific objectives of this study are as follows:

- To incorporate a thorough understanding of blockchain technology into the evaluation, considering its effects on the security and performance of the underlying architecture, consensus processes, and cryptographic protocols.
- To examine the security measures put in place in two popular Distributed Ledger Technologies (DLTs), Ethereum and Hyperledger Fabric, under Distributed Denial of Service (DDoS) attack scenarios.
- To compare the performance metrics of Ethereum and Hyperledger Fabric, such as latency and throughput, under Distributed Denial of Service (DDoS) attack scenarios to determine their resilience against such attacks.

Ethereum, Hyperledger Fabric, and Hyperledger Caliper are briefly explained as follows.

1.1 Ethereum

Ethereum is a blockchain-based platform that facilitates the establishment of a distributed network of computers capable of executing and validating 'smart contracts,' which are autonomous programs, in a secure setting [2]. As a result, Ether (ETH), the native currency of the Ethereum blockchain, can be stored on it. Currently, Ether stands as the second largest cryptocurrency. Smart contracts, operating without a central authority, facilitate transactions between parties by triggering actions when certain conditions are met. Once created, a smart contract cannot be changed. Participants have total control over and insight into transaction data due to the distributed, consistent, and authentic transaction records securely scattered around the network. Users create Ethereum accounts to send and receive transactions. Ethereum's most notable feature has earned it the tag of 'programmable money'.

1.2 Hyperledger Fabric

Hyperledger Fabric is another platform for DLT. Within the Hyperledger ecosystem, it stands as the first private DLT system, where access is restricted to a specific group or participants. Its high level of privacy is derived from its intended usage in various scenarios by both corporate and governmental organizations. Among the primary characteristics that distinguish Fabric from other DLTs is its ability to support multiple ledgers within its ecosystem [3]. Chaincode, the name of the smart contract on this technology, is a computer program loaded on the ledger and written in either Java or Go. The Chaincode handles all the interactions with the ledger data.

1.3 Hyperledger Caliper

Hyperledger Caliper, an automated performance evaluation framework, operates as a benchmark tool for Distributed Ledger Technology (DLT), enabling users to assess the performance of a DLT implementation through a predefined set of use cases. In this paper, a customized version of Hyperledger Caliper [4] is employed as a benchmarking tool to assess the performance of Ethereum and Hyperledger Fabric, utilizing Hyperledger Caliper's generated metrics such as Transactions Per Second (TPS) and Resource Use. To determine the execution time of transactions, adjustments have been made to the Hyper Ledger Calculator in this research. Furthermore, a Fabric v0.6, compatible with Hyperledger Caliper, is built, along with the deployment of the execution time within the tool. The Hyperledger Caliper architecture, for assessing the performance of various DLT, consists of four primary layers: Network Layer, Adaptation Layer, Interface & Core Layer, and Benchmark Layer. The Adaptation Layer is a crucial component of the Caliper architecture, tasked with incorporating various blockchain implementations into the assessment framework. Its primary function is to act as an intermediary between the standard northbound interfaces (NBIs) and the DLT protocol for each tested DLT platform. The Interface & Core Layer is responsible for providing various northbound DLT interfaces necessary for smart contract deployment, invocation, and querying. Additionally, this layer implements CPU and memory resource monitoring functions. The Benchmark Layer conducts stress tests on the deployed blockchain platform. Each stress test requires inputs such as test parameters and blockchain network data.

The remainder of this paper is organized as follows: Section II reviews related work in the field, Section III discusses the methodology and design of the work presented in this paper, Section IV presents the implementation, Section V shows the results of the evaluation for Ethereum and Hyperledger Fabric under different DDoS attack scenarios, and Section VI concludes the paper with directions for future work.

2 LITERATURE REVIEW

DDoS attacks occur in various domains and networks, targeting a wide range of systems and services. This includes websites and web services [13], online gaming, financial services [14], cloud services [15], IoT devices [16], government public services [17], educational institutes [18], healthcare and medical services [19], entertainment, vehicular networks [20] as well as critical infrastructure [21]. In this section, we review some papers related to attacks on DLT and methods for securing such systems against anomalies. We also include some papers that offer performance evaluations of DLTs under different metrics.

In [8], the authors use Hyperledger Caliper to evaluate the performance of two Ethereum networks: an Ethereum private network and the Ropsten testnet. They take advantage of Hyperledger Caliper, where different transactions can be tested on different blockchains. Addressing their findings, the Ethereum private network performs better than the Ropsten testnet overall. This performance is influenced by the contents of the transactions.

In [9], the authors propose an Ethereum (ETH)-based approach to securing industrial Internet systems against two attacks: Byzantine attacks and DDoS attacks from inside the system. To achieve this, they develop a credit mechanism-based Bayesian inference method and a miner selection method. To verify the effectiveness of their proposed scheme, simulation scenarios based on a smart factory is conducted. Their findings demonstrated that the proposed system is capable of identifying 90% of the false messages broadcasted by Byzantine attackers.

In [10], the authors study Ethereum transactions considering its two clients: Geth and Parity, on a private blockchain to understand the effect of utilizing different clients on Ethereum's performance. Addressing their results, Ethereum transactions are significantly faster in the Parity client compared to the Geth client while using the same system configurations.

In [11], the authors conduct a performance analysis of Hyperledger Fabric and Ethereum under varying loads in the form of transactions. They developed a methodology to evaluate blockchain and then analyse the results, showing that Hyperledger Fabric consistently outperforms Ethereum across all evaluation metrics, including execution time, latency, and throughput. Additionally, they demonstrate that Ethereum is able to handle a greater number of concurrent transactions.

In [12], the authors present a blockchain system with a credit-based consensus mechanism for the Industrial Internet of Things (IIoT). Their proposed system employs a credit-based proof-of-work (PoW) mechanism for IoT devices to ensure system security and transaction efficiency simultaneously. They develop a data authority management method to regulate access to sensor data in order to protect the confidentiality of sensitive data. Their system is built based on Directed Acyclic Graph (DAG) structured blockchains instead of Satoshi style blockchain. The experimental results demonstrate that their proposed system is secure and effective for IIoT.

Our research in this paper expands previous studies by analysing and contrasting performance metrics across multiple blockchain networks, whereas other efforts, including those referenced in [8], have primarily focused on specific blockchain platforms. Our study systematically assesses important parameters, including transaction throughput and latency, across multiple blockchain architectures.

3 METHODOLOGY & DESIGN

The attacks and technical setups of the platforms chosen for implementation are covered in this section. Ethereum and Hyperledger Fabric (HLF) are the selected ledgers. Choosing two distinct platform typologies can aid in better understanding how each functions and can be customized. Additionally, it simplifies the analysis, comparison, and determination of the most suitable method for various applications. The guidelines and ideas presented here are applicable to various use cases.

Several packages must be installed on the computer to use the HLF platform. The MacBook Air M1 laptop, which has an 8-core CPU, 16GB of RAM, and 512GB of SSD, was the node used to run the network. On Windows Subsystem for Linux (WSL)2, every command was entered into the Kali Linux terminal accessed through UTM. This paper uses the Debian OS flavour. Use the command "apt-get update && apt-get upgrade" to keep the machine up to date before beginning and installing the required tools. Docker was the first tool to be downloaded. Docker technology enables the packaging and running of software inside distinct containers. Every component of HLF, such as peers, orders, and Certificate Authority (CA), is supplied in the form of Docker images. Docker version 20.10.17 is being utilized. We also need to download the Java programming language, GO, and Node.js, as HLF utilizes them to create smart contracts. The mentioned tool versions are OpenJDK 64-bit Server VM, Java OpenJDK 17.0.4, Go version 1.13.8, and Node version 12.16.1. HLF samples, binaries, and Docker images should be installed last [5]. The Hyperledger Fabric version 2.2.5, fetched from GitHub using the command "curl -sSL https://bit.ly/2ysbOFE | bash -s – 2.2.5 1.5.2", is used in this paper. The file "network.sh" can be found by using the command "cd fabric-samples/test-network" to navigate to the test-network folder. This script enables the test network to be turned on, as well as the creation of a channel and CA. It also supports several functions, including identity registration, connection to Lightweight Directory Access Protocol (LDAP) as the user registry, issuance of Enrolment Certificates (ECerts), and certificate renewal and revocation. "./network.sh up createChannel" is the command to create the network. Deploying chaincode on the channel with a prewritten script in the "asset-transfer-basic/chaincode- javascript" file is carried out with the command "./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-javascript-ccl javascript".

We utilized the Hyperledger Besu client for Ethereum, which provides access to both public and private platforms. For this paper, we developed a private network that operates similarly to a public network, adhering to specific protocols and procedures. Hyperledger Besu is equipped with an extractable Ethereum Virtual Machine (EVM) implementation, designed with enterprise users in mind for various use cases, including both private and public permissioned networks. Moreover, it supports test networks such as Görli and Sepolia. Hyperledger Besu incorporates several consensus algorithms, including Proof of Stake (PoS), Proof of Work (PoW), and Proof of Authority (PoA). Its comprehensive permissioning methods are tailored for collaborative settings [6].

For Ethereum, Hyperledger Besu is utilized to implement the platform, which uses the Solidity language to create contracts. The existing network nodes can be verified by running the "docker ps" command in the Windows Command Prompt on a machine running Windows 10 OS, accessible through UTM. In this report, Remix IDE, a web-based Integrated Development Environment (IDE) primarily used for Ethereum smart contract, was utilised to interact with the Ethereum platform via MetaMask. MetaMask is a cryptocurrency wallet and browser extension that enables users to interact directly with the Ethereum blockchain from their web browser.

4 IMPLEMENTATION

We perform Denial of Service (DoS) attacks on both pre-existing Blockchains using three distinct methods. Since HLF and Ethereum are private and public in nature respectively, the following are the optimal test scenarios to maintain the same attack with the same values on distinct blockchains, enabling us to observe the differences or similarities between them.

- Keeping the number of users constant and varying transactions.
- Keeping transaction constant and varying users.
- Gradually varying both number of users and number of transactions.

We perform attacks on the Ethereum network and the HLF using Hyperledger Caliper, a blockchain benchmarking tool that enables users to assess the performance of a blockchain implementation using a predetermined set of use cases. Hyperledger Caliper is compatible with various blockchain solutions, including Hyperledger Besu, Hyperledger Burrow, Ethereum, HLF, FISCO BCOS, Hyperledger Iroha, and Hyperledger Sawtooth. It generates reports with a variety of performance indicators [7]. The configuration of HLF and Ethereum to be evaluated by utilizing Hyperledger Caliper is as follows.

4.1 Hyperledger Fabric Configurations

The entire configuration is divided into five steps as follows.

- Creating Caliper workspace: to use Caliper as a performance evaluator against HLF, it is necessary to create a Caliper workspace by downloading and installing the tool. Three configuration files are required for Caliper: network configuration files, user files, and a benchmark file.
- Creating Network Configuration file (networkConfig.yaml): Caliper needs the network configuration file in order to accept and process transactions on a Hyperledger Fabric network. The file can be in JSON or YAML format. The file contains information such as Name, Version, Caliper, Channels, and Organisations.
- Creating Workload file (readAsset.js): this is required to provide communication with the deployed smart contract during the benchmark session.
- Creating Benchmark Configuration file (myAssetBenchmark.yaml): in this file, the defined workload modules and benchmark rounds are referred to. It will detail how many test workers to utilize to generate the load, how many test rounds there will be, how long each round will last, how the rate control is applied to the transaction load during each round, and any settings on monitoring.
- Running Caliper Benchmark: now that we have the configuration files and test module ready, we can proceed to run the performance benchmark. We will use the Caliper CLI, which requires the path to the workspace and workspace relative paths to the network configuration file and the benchmark configuration file, to conduct the performance benchmark. The final report for each benchmark round will include the following details: Name, Success/Failure, Send Rate, (Max/Min/Average) Latency, and Throughput.

4.2 Ethereum (Hyperledger Besu)

The entire configuration is divided into three steps as follows.

- Creating Caliper workspace: Caliper workspace should be created at the same level as the Besu directory.
- Creating config.json file: this file contains details of the network, including the name of the blockchain and address details. It is required to connect the benchmark file to the existing Ethereum blockchain.
- Creating benchmark file (config.yaml): This file serves as a central configuration file, allowing users to specify parameters and configurations unique to their blockchain implementation. The config.yaml file in Hyperledger Besu encompasses numerous configurations, including privacy settings, consensus techniques, and network parameters. The blockchain network can be customized to meet individual needs by adjusting parameters such as block duration, transaction count, number of nodes, gas restrictions, and privacy group setups.

In the first scenario, the number of users remains constant at 1, 10, 20, 30, and 40, while transaction counts for each node range from 1 to 1,000,000. Initially, one user completes one transaction, followed by ten transactions at once, then 100 transactions, and so on, until reaching 10,000,000 transactions per user. The same procedures are repeated for 10, 20, 30, and 40 users. Throughout the testing process, throughput and latency are regularly observed, and the data are recorded in a tabular format. A total of 35 tests were conducted for the first scenario, focusing on throughput and latency.

In the second scenario, the number of transactions was maintained at 1, 10, 100, and 1000, with 1, 2, 5, 10, 15, and 20 nodes being used for each transaction. To obtain the values on both networks, 24 transactions were performed. This involved starting with 1 transaction using 1 node and then proceeding to 1 transaction with 2 nodes, 5 nodes, 10 nodes, 15 nodes, and finally 20 nodes. The same sequence was repeated for 10 transactions, with each set executing continuously 100 and 1000 transactions.

Finally, the testing involved increasing transactions and users simultaneously, with scenarios such as 1 user executing 1 transaction at a time, 5 users executing 5 transactions concurrently, 10 users with 10 transactions each, and continuing with 15, 20, 25, 30, 35, and 40 users. Testing was stopped due to computational requirements.

5 RESULTS

After completing the three scenarios outlined in the previous section for Ethereum and HLF using Hyperledger Caliper, we obtained data on latency and throughput. This data is tabulated and visualized in the graph below to facilitate better evaluation. The following results depict the plotted graphs for each set of experiments, showing comparability across their respective scenarios.

For instance, the results from the first scenario, where the number of users is kept constant at 1, 10, 20, 30, and 40 and transactions are varied from 1 to 1,000,000, are depicted in Fig. 1. The X-axis represents the number of transactions, while the Y-axis shows the throughput. The throughput of HLF drops after 10,000 transactions, whereas that of Ethereum drops after 100 transactions. From Fig.

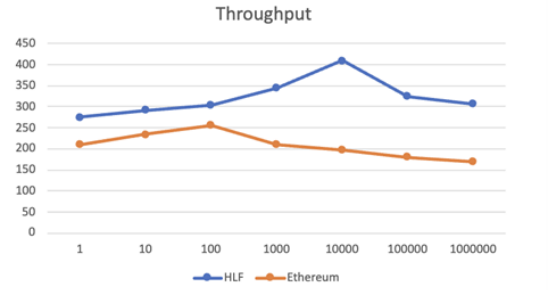

Figure 1: Ethereum vs. HLF in scenario1 (throughput vs. transaction).

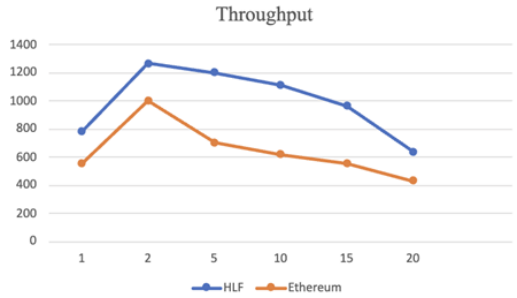

Figure 2: Ethereum vs. HLF in scenario2 (throughput vs. number of nodes).

1, we observe that the throughput of HLF is higher than that of Ethereum in all cases and shows better resilience when the number of transactions increases dramatically.

Additionally, in scenario 2, where the number of transactions was maintained at 1, 10, 100, and 1000, and with 1, 2, 5, 10, 15, and 20 nodes being used for each transaction, both HLF and Ethereum exhibit a similar pattern. However, HLF demonstrates superior throughput throughout the entire scenario, Fig 2

In scenario 3, testing involved increasing transactions and users simultaneously. This included scenarios such as one user executing one transaction at a time, five users executing five transactions concurrently, ten users with ten transactions each, and so on, up to forty users with forty transactions each. In this scenario, both HLF and Ethereum exhibit a similar pattern, but the former demonstrates higher throughput throughout the entire scenario, Fig 3

As latency represents the time taken to complete a transaction, Fig. 4 shows that Ethereum has higher latency than HLF. Latency increases suddenly after 10,000 transactions in both Ethereum and HLF, with Ethereum exhibiting higher latency compared to HLF. Similarly, in scenarios 2 and 3, HLF exhibits superior latency compared to Ethereum, as shown in Fig. 5 and Fig. 6, respectively.

In summary, Figs. 1 to 6 demonstrate that the throughput and latency of HLF are consistently higher than Ethereum across all three scenarios.

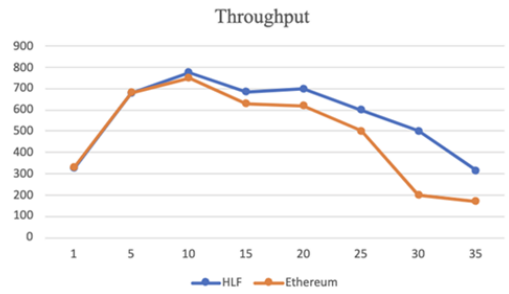

Figure 3: Ethereum vs. HLF in scenario3 (throughput vs. number of users).

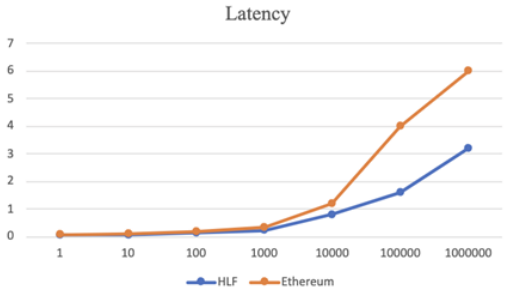

Figure 4: Ethereum vs. HLF in scenario1 (latency vs. transaction).

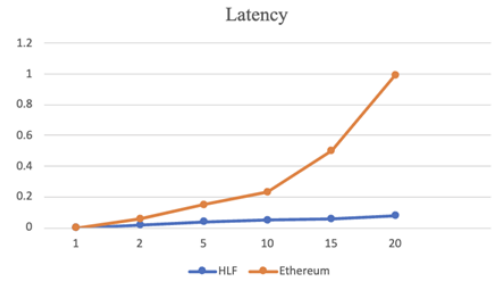

Figure 5: Ethereum vs. HLF in scenario2 (latency vs. number of nodes).

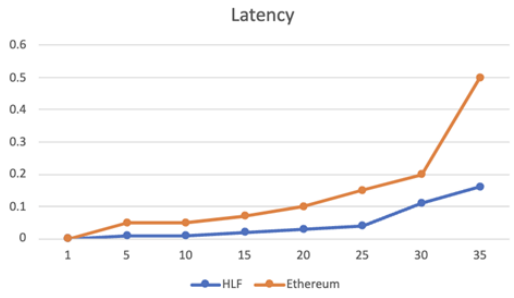

Figure 6: Ethereum vs. HLF in scenario3 (latency vs. number of users).

6 CONCLUSION & FUTURE DIRECTION

Blockchain offers a decentralized solution to address various challenges including data integrity and security, trust and transparency, smart contracts, supply chain management and security, identity management, as well as data monetization and sharing in the context of smart cities and CPS. However, blockchains are vulnerable to a range of threats, much like any other technology. In this paper, the two DLTs under investigation exhibited different performance metrics, as revealed by the captured results. HLF demonstrated greater efficiency and reduced latency, showcasing its resilience against various DDoS attack vectors. Conversely, Ethereum showcased its robustness in adversarial situations, revealing both its strengths and weaknesses. As a permissionless blockchain, Ethereum can face challenges such as network congestion and scalability issues. In contrast, HLF is designed as a permissioned blockchain, where participants are known and have defined roles. Addressing the results obtained through experiments, the answer to the RQ posed at the start of this manuscript is as follows:

- What are the comparative performance characteristics of Ethereum and Hyperledger Fabric in relation to DDoS attacks?
- Hyperledger Fabric demonstrates greater resilience to cyber attacks, such as DDoS, compared to Ethereum.

As part of our ongoing research, we aim to evaluate the performance of additional blockchain platforms using various consensus techniques and explore additional test scenarios, such as determining the maximum number of transactions that a particular blockchain platform can process with optimal efficiency. Additionally, we plan to experiment with other network configurations, such as scaling the number of nodes, to assess how node scalability impacts the performance of a blockchain platform.

REFERENCES

[1] Nakamoto, S, "Bitcoin: A Peer-to-Peer Electronic Cash System.", [online] Available at: https://assets.pubpub.org/d8wct41f/31611263538139.pdf.

[2] AWSBlockchain. [online] Available at: https://aws.amazon.com/blockchain/whatis-ethereum/.

[3] Li, D., Wong, W. E., & Guo, J. , "A survey on blockchain for enterprise using hyperledger fabric and composer", in 2019 6th International Conference on Dependable Systems and Their Applications (DSA) (pp. 71-80). IEEE.

[4] GitHub. (2023). hyperledger/caliper. [online] Available at: https://github.com/hyperledger/caliper.

[5] Yli-Huumo, J., Ko, D., Choi, S., Park, S. and Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. PLOS ONE, 11(10), p.e0163477. doi: https://doi.org/10.1371/journal.pone.0163477.

[6] Hyperledger Besu. [online]Available at: https://www.hyperledger.org/projects/besu [Accessed 20 Jul. 2023].

[7] S. S. Smith, "Emerging Technologies and Implications for Financial Cybersecurity", International Journal of Economics and Financial Issues, 10(1), 27, 2020.

[8] Choi, W., & Hong, J. W. K., "Performance evaluation of ethereum private and testnet networks using hyperledger caliper", In 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), (pp. 325-329), 2021.

[9] Yang, X., Gao, Q., Basin, M. V., Li, H., & Peng, X. , "An ETH-based approach to securing industrial Internet systems against mutinous attacks", Information Sciences, 655, 119904, 2024.

[10] Rouhani, S., & Deters, R. , "Performance analysis of ethereum transactions in private blockchain", In 2017 8th IEEE international conference on software engineering and service science (ICSESS) (pp. 70-74). IEEE, 2017.

[11] Pongnumkul, S., Siripanpornchana, C., & Thajchayapong, S., "Performance analysis of private blockchain platforms in varying workloads", In 2017 26th international conference on computer communication and networks (ICCCN) (pp. 1-6). IEEE.

[12] Huang, J., Kong, L., Chen, G., Wu, M. Y., Liu, X., & Zeng, P., "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism", IEEE Transactions on Industrial Informatics, 15(6), 3680-3689, 2019.

[13] Prasetyo, S. E., Haeruddin, H., & Ariesryo, K., "Website Security System from Denial of Service attacks, SQL Injection, Cross Site Scripting using Web Application Firewall", Antivirus: Jurnal Ilmiah Teknik Informatika, 18(1), 27-36., 2024

[14] Kannan, Y. , "Impact of Internet of Things (IoT) devices on Network Security at Financial Institutions", Authorea Preprints, 2024.

[15] Shafi, M., Lashkari, A. H., Rodriguez, V., & Nevo, R. , "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS) Dataset and Cloud Intrusion Traffic Characterization", Information, 15(4), 195, 2024.

[16] Kumar, A., & Singh, D. , "Detection and prevention of DDoS attacks on edge computing of IoT devices through reinforcement learning.", International Journal of Information Technology, 16(3), 1365-1376, 2024.

[17] Gaie, C., "Enhancing the Efficiency and the Security of e-Government: The French Case Study of Human Resources Applications. In Transforming Public Services—Combining Data and Algorithms to Fulfil Citizen's Expectations", (pp.223-239). Cham: Springer Nature Switzerland, 2024.

[18] Singh, B., & Kumar, B., "A Comprehensive Analysis Of Key Factors Causing Various Kinds Of Cyber-Attacks In Higher Educational Institutes", Journal of Research Administration, 6(1), 2024.

[19] Sripriyanka, G., & Mahendran, A., "Securing IoMT: A Hybrid Model for DDoS Attack Detection and COVID-19 Classification", IEEE Access, 2024.

[20] Verma, A., Saha, R., Kumar, G., Conti, M., & Kim, T. H., "PREVIR: Fortifying Vehicular Networks against Denial of Service Attacks", IEEE Access, 2024.

[21] Srivastava, A., Saini, P. K., Tiwari, S., Sawan, V., & Garg, N., "Securing SCADA System from DDoS Attack", In 2024 2nd International Conference on Computer, Communication and Control (IC4) (pp. 1-6). IEEE, 2024.