International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Implementation of Parallel Proof of Work Algorithm for Blockchain Using SimBlock Simulator

Ms. Priyanka, Dr. Ritu Makani

Submitted: 29/01/2024 **Revised:** 07/03/2024 **Accepted:** 17/03/2024

Abstract: The distributed, peer-to-peer network-based technology known as blockchain is used to maintain decentralized databases without the need for centralized or third-party systems, which are becoming increasingly popular these days. Although blockchain offers immutability and security for blockchain transactions, it also has limitations because there are no reliable methods for confirming the operations of large blockchain networks. This study covered blockchain network simulators mostly used for blockchain simulation, such as Blocksim, VIBES, Blockchain Demo, Blocksim-Net, and SimBlock. Unlike the existing simulators, SimBlock may easily modify the node's behaviour on the blockchain if necessary to enable investigating the node's behaviour's impact on the blockchain. SimBlock is an easily configurable collection of nodes with some blockchain network settings that aid in simulating the peer-to-peer network of several types of blockchain, such as Ethereum and Bitcoin. SimBlock also facilitates the visualization of node behaviour and block propagation. We analyzed the use of better neighbour strategies and the implications of the relay network in this study using the blockchain simulator SimBlock.

Keywords*: Blockchain, Simulator, peer-to-peer, SimBlock, decentralized.*

1. Introduction

One of the main technologies behind cryptocurrencies is blockchain, which is currently receiving increased attention for use in a variety of applications outside of cryptocurrency. Blockchain offers several benefits, such as the ability to manage the ledger decentralized and easily accessible by all nodes, as well as the inability for third parties to alter previously received block data. Blockchain's use in cryptocurrency and other sectors is growing because of some of its strongest qualities.

Blockchain technology has several advantages, which is why it has become so successful. Initially, immutable ledgers—which cannot be modified or altered by nature—are produced using blockchain technology. Transactions are non-modifiable once they are created and recorded [16]. Blockchain's reliance on decentralized control, where all participating nodes' resources are utilized, is another essential feature. As a result, a single point of failure is resolved [15]. Thirdly, users' identities are well protected by blockchain. Fourth, the mitigation of the SPOF issue has led to greater security in blockchain technology [17]. Last but not least, blockchain allows involved nodes to promptly and cooperatively process transactions [18].

Due to its peer-to-peer scalable architecture, blockchain characteristics can present certain challenges. These are:

- It is challenging to obtain information about the entire blockchain network, unless the nodes themselves supply it, in the large-scale public network. Until we have all the knowledge, we cannot experiment.
- In the blockchain's small-scale private network, we are unable to investigate specific network behaviour.

Planning experiments on a practical and large-scale blockchain network is challenging for these primary reasons. However, adopting a large number of nodes results in significant network costs in addition to empirical circumstances, and network design modification is difficult. We employ the blockchain network simulator SimBlock to address these issues. We illustrate some of these through simulation, such as investigating a better neighbour choice approach

[1]*Research Scholar, Department of CSE, GJU S&T, Hisar and Assistant Professor of Computer Science, CBLU, Bhiwani, Haryana, India*
[2]*Associate Professor, GJU S&T, Hisar, Haryana, India*

and evaluating the relay network's effects. so that we may easily conduct blockchain research using the SimBlock simulator.

1.1 Need for Modelling and Simulation

The performance of complex systems can be analyzed and predicted with the use of modeling and simulation. Simulators imitate the behaviour of these systems, capturing important aspects of them and enabling experimentation without actual implementation [11]. The network layer, consensus layer, data layer, execution layer, and application layer are the five layers that make up a typical blockchain system [12][13][14]. These layers are complicated. Consequently, testing and assessing real-world blockchain systems' functionality can be difficult. Therefore, blockchain simulation is frequently a good substitute for the following two key reasons. It first relieves the financial and computational load associated with implementing and testing blockchain technologies. Secondly, it facilitates the assessment of blockchain performance across diverse scenarios and parameter setups.

2. Research Methodology

An investigation of studies related to blockchain simulators is the goal of this paper's systematic mapping study [19]. To cover more ground than just a cursory review of current blockchain emulators, this work employs a methodical mapping strategy. Stated differently, a systematic mapping review offers analytical techniques that critically assess the blockchain simulator literature in addition to aiding in the focus of subject investigation on certain questions. Our ability to recognize and chart significant research avenues will also be facilitated by the study's results. The five phases of systematic mapping utilized in this investigation are depicted in Figure 1.

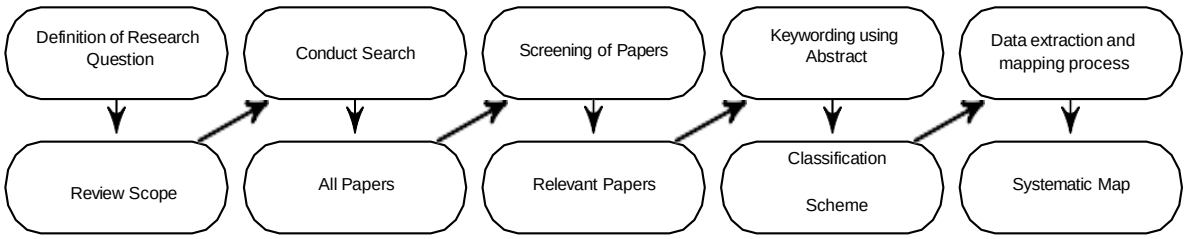

Figure. 1. Phases of the methodical mapping investigation

3. SimBlock

To advance blockchain research, a distributed system group at the Tokyo Institute of Technology built it. It supports the peer-to-peer blockchain networks, or nodes, that are found on a single computer system in blockchains like Litecoin and Bitcoin. Since the SimBlock simulator is event-driven, events include the creation of a block as well as the sending and receiving of messages. This simulator allows us to easily apply the algorithm for choosing neighboring nodes. The increasing likelihood of block formation and the regeneration of the block allow for the simple determination of the block's creation time. Block mining requires a large amount of processing power, and it is simple to mimic a large network with a large number of nodes. By default, the block is created together with the simulated blockchain network using the Proof of Work probability assumption. By adjusting the block creation parameters, we can utilize alternative algorithm proofs of the stack for simulation. Furthermore, SimBlock functions as a visualizer for depicting block behaviour and node propagation. SimBlock simulates the behaviour of individual nodes in a blockchain network over an extended period. This makes it possible to evaluate the security and performance of various blockchain configurations without having to put them into use on an actual network.

Figure 2 depicts the message exchange and block generation. A node that receives an INV message replies with a GETDATA message and waits to obtain the block if it doesn't already have it. A block containing a large amount of data is not sent needlessly when such a protocol is used.

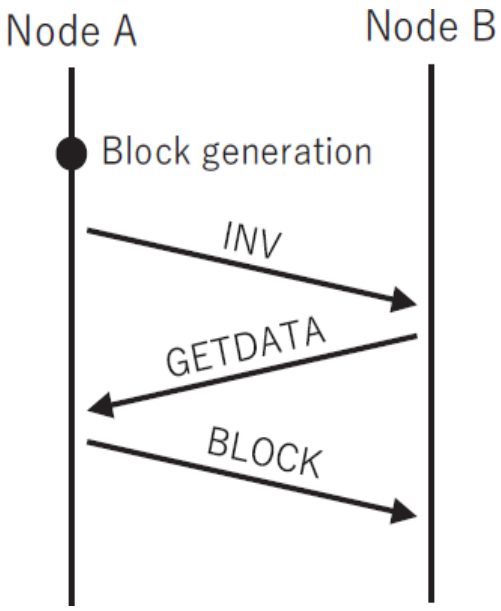

Figure 2 Block propagation protocol between two nodes

To sum up, SimBlock is an invaluable resource for scholars, programmers, and anybody else curious about how blockchain networks behave. Through the process of simulating various scenarios and setups, you can obtain important insights into how this innovative technology functions.

3.1 Functioning of SIMBLOCK:

- **Event-driven simulation:** Being an event-driven simulator, SimBlock works by modeling discrete events like as the production of new blocks, the propagation of transactions, and consensus voting. Every event is timestamped and causes particular actions to be taken on the impacted nodes. With this method, the timing and interdependence between various events in the blockchain network may be accurately modelled.
- **Node behaviour customization:** In your simulation, you can specify how each node behaves. This encompasses variables like processing power, network throughput, and application of consensus algorithms. Malicious nodes can also be introduced to look for security flaws.
- **Network topology:** The topology of the blockchain network, including the quantity, location, and interconnections of its nodes, can be customized using SimBlock. This affects network resiliency and block propagation time, among other things.
- **Consensus algorithm modeling:** Convergence methods supported by SimBlock include Byzantine Fault Tolerance, Proof-of-Work (PoW), and Proof-of-Stake (PoS) (BFT). For each sort of blockchain you are modeling, you can select the proper algorithm and examine its performance in various scenarios.
- **Data collection and visualization:** SimBlock gathers information on several measures, including network latency, block production rate, and transaction throughput, while the simulation is running. Afterward, you can use the visualization tools that are offered to see this data and have a better understanding of how the simulated network behaves overall.

3.2 3.2 Benefits of using SimBlock:

- **Experimentation:** Evaluate various consensus algorithms and blockchain settings without jeopardizing real-world installations.
- **Performance analysis:** Examine variables like latency, throughput, and scalability in different network scenarios.
- **Security evaluation:** Examine possible weak points and evaluate defenses against intrusions.
- **Research and development:** Create and enhance novel blockchain algorithms and protocols.

3.3 Design and Features: We took into account the

block, node, and network parameters in SimBlock.

3.3.1 Parameters of Block

- **Size of Block:** In Blockchain, a block's real size depends on the node that generated it.
- **Block generation interval:** The time between blocks in a blockchain depends on the blocks it generates.

3.3.2 Parameters of Node

- **Total nodes:** Every node in the present Blockchain network that is engaged.
- **Block generation capacity:** Every block in a network has a unique capability for producing new blocks. The block generation difficulty can be simply ascertained by adding the created block capacity for each network node and the required value for the block interval generation. The computational power of a block is what determines its capacity in Proof of Work.
- **Position of the node:** Within the network, a node's position is determined by its geographical location.
- **Total neighbour nodes: T**he total number of nodes connected to each specific node that is operational at the moment.

3.3.3 Network Parameters

- **Network Bandwidth:** Both the downstream and upstream bandwidth for each block region are used to calculate the overall bandwidth. To transfer a message from Block Region X to Block Region Y, for instance, the bandwidth between the two regions is measured as the minimum bandwidth rate of the upstream value of Block Region X and the downstream value of Block Region Y.
- **The Network propagation delay:** By averaging the block propagation delays between different block locations, the propagation delay of the network is ascertained.

  Two fundamental criteria are often preferred when determining the precise arrival time of a block message: the node's bandwidth and the propagation delay between two nodes. The precise bandwidth values of both sites and the message size are used to compute the transmission time. The propagation delay and the transmission time are added to determine the message-receiving time. Wherein the management of every neighbour node is handled by a unique class. Every time a transmission is made from one transmission node to the receiving node, the management class of the node next to it calls the associated function. to alter this management class function and thus the neighbour node's selection algorithm.

4. METHODOLOGY

4.1 Evaluation and Difficulty Adjustment

Similar to the current simulator discussed by Gervais et al. [4], The simulator conducted an evaluation, and a comparative experiment was conducted in a similar setting. This simulator's main goal was to determine how resistant it was to double-spending attacks while varying the block size and generation interval. Therefore, altering the block's properties, like as its size, is straightforward; but, altering the algorithm that establishes the network's topology or a node's consensus process is more challenging.

This conclusion is supported by the experiment carried out by Gervais et al. on renewing the simulator's variables. Using models of the present environments for Dogecoin, Litecoin, and Bitcoin, we calculated the fork occurrence rate and the time it would take for the created block to reach half of the network's connected nodes. Gervais et al. (2015) measured the total number of nodes, block size, and node dispersion by location in the blockchain. By rebuilding their network at the time, Yusuke et al. selected the propagation delay and bandwidth of each of the six regions: Asia, Europe, Japan, North America, Australia, and South America. According to the bandwidth, actual information, and propagation delays between the various regions, which were used for all the nodes, the entire network was divided into these six regions. Table 1 represents the block parameters:

Parameters	Dogecoin	Litecoin	Bitcoin
# Of the nodes	600	800	6000
Block interval	1min	2min30sec.	10min
Block size	8 KiB	6.11 KiB	534KiB

# Of the connection	Miller et al.'s distribution estimates [3]
Geographical Distribution	Distribution in line with the real blockchain
Bandwidth	
Propagation delay	Sixth-regional bandwidth and propagation delay

Table 1

Gervais et al.'s Simulator parameters [10]

The Miller et al. observation [3] provides a total number of node connections. Yusuke et al.

[1] simulated the selection of a random node from the entire network and fixed it as a neighbour node until up to 10,000 blocks had been created. The results of Gervais et al., genuine data, and SimBlock results as determined by Yusuke et al. are shown in Table II [1]. The prior simulator and that of Gervais et al. can simulate the chain of blocks with outstanding accuracy whenever the outcome is close to the discovered value. These SimBlock-simulated values were quite identical to the values Gervais et al. determined for identical network features.

Sim-Block ensures that the hash rate and block structure remain constant during the simulation procedure by initializing them. It does not represent the true environment of a blockchain network, where its difficulty varies according to how long it took to mine prior blocks after all N blocks had been mined. For example, the degree of difficulty in Bitcoin varies with each block of 2016. To avoid abrupt changes to complexity over a short period, complexity adjustments cannot increase or decrease by more than four times the present level. Every digital currency has a different difficulty modification mechanism, with different adjustment criteria or computationally challenging calculations. Consequently, we proposed merging SimBlock with the existing Bitcoin difficulty modification method.

	Bitcoin	**Litecoin**	**Dogecoin**
Block interval	10 min	2.5 min	1 min
Measured tMBP	8.70s	1.02s	0.98s
Gervais et al. tMBP	9.42s	0.86s	0.83s
SimBlock tMBP	8.94s	0.85s	0.82s
Measured rs	0.41%	0.27%	0.62%
Gervais et al. rs	1.85%	0.24%	0.79%
SimBlock rs	0.58%	0.30%	0.80%

Table II

Comparison of the median block propagation (tMBP) and fork rate(rs)in real and simulated blockchain networks [1]

However, the "difficulty" indicator is not used by Bitcoin itself. A user can assess "how challenging it is to mine a block at the time" using this metric. The great majority of cryptocurrencies use a Proof-of-Work technique to verify that their variable factor "target" is regularly modified, ensuring that the block creation frequency is at the scheduled duration (for Bitcoin, 10 minutes). Every network node experiences "retargeting" independently and independently of the others. The objective is specified to guarantee that the network's current hash rate can accomplish the anticipated block generation interval. It is more difficult to mine a whole block when it comes to the target. Target is a 32-bit "compact" representation of a 256-bit unsigned integer, which can be displayed in hexadecimal notation and is difficult to completely comprehend. The general difficulty management method is demonstrated by Algorithm 1 in the most recent version of Bitcoin: For each block that is formed,

check to see if it has been mined before the latest difficulty modification in 2016.

Next, find out how long it takes to mine a block for the blocks that were mined in 2016. Divide the time it took to mine the 2016 blocks by 4 if it was less than what was expected; if it was higher, multiply the time by 4. This process restricts the adjustment in order to avoid sudden shifts in difficulty. To get at the current target, multiply the current block by the actual time required to mine 2016 blocks.

To determine the new goal, divide the new target by the anticipated time needed to mine 2016 blocks. The lowest difficulty level in Bitcoin is 1, hence a maximum aim is a number that equals that number. This will ensure that there will be no difficulty level lower than 1. This will guarantee that the difficulty level never drops below 1. If it is lower than 1, then it is simply set as 1 rather [7]. Equation 1 provides an equation for the newly established goal (Ti+1).

$$T_{i+1} = T * \frac{\sum_{i=1}^{N} x_i}{N*B} \quad (1)$$

where B is the estimated duration for mining a block, xi is the amount of time required to mine block I, and T is the present objective. The interval for difficulties or retargeting is represented by the letter N. As we mentioned earlier, T and Equation 1 are applied internally by Bitcoin to ensure that a block is generated once every 10 minutes. As a result, Equation 2 may be used to determine the complexity D using the target T.

D = Tg/Tc (2)

where Tg is the target of the genesis block (block #0) and Tc is the current target. [8].

Algorithm 1 Bitcoin's general difficulty adjustment algorithm

1: target Timespan = estimated duration of block mining (seconds)* N

2: If (currentBlockHeight+1) % N is equal to zero, then

3: total Interval = elapsed time to mine N blocks 4: if total Interval < target Timespan then

5: total Interval = target Timespan / 4 6: end if

7: if total Interval > target Timespan then 8: total Interval = target Timespan * 4

9: end if

10: new Target = getOldTarget (existing Block) 11: new Target = new Target * total Interval

12: new Target = new Target/target Timespan 13: if new Target > maximum goal then

14: New target = maximum goal 15: end if

16: end if

Equation 3 is utilized rather than Equation 1 to get the recommended result because the complexity of estimating the true time needed to mine a block was previously established in Sim Block's initial execution. The difficulty is directly accessed in this approach, which sets it apart from others. The further difficulty (D_{i+1}) for our suggested SimBlock implementation is given in Equation 3.

$$D_{i+1} = D*\frac{N*B}{\sum_{i=1}^{N} x_i} \quad (3)$$

Further evidence that the Litecoin difficulty adjustment method is the same as the Bitcoin one comes from Equation 1. Unlike Bitcoin and Litecoin, which update their difficulty every 2016 block, Dogecoin updates it every single block. Litecoin is expected to take less time to mine a block than Dogecoin, at about 2.5 minutes. The complicated adjustment algorithms for Litecoin and Dogecoin are likewise featured in SimBlock because both of these cryptocurrencies are identically simulated.

4.2 Hash rate

In Sim Block's initial implementation, the global hash rate is fixed at activation and remains that way throughout the simulation period [6]. Even if each simulated node is given a distinct hash rate by Equation 1, the time it takes to find and mine a block in real life will differ. However, the overall simulation result won't change significantly because the network's overall hash rate doesn't change over time. The predicted hash rate for the actual Bitcoin networks between July 2019 and January 2023 is displayed in Figure 3. This is not a realistic representation of the condition of a distributed ledger system in the real world, where the hash rate varies regularly. Once every N blocks, the first choice regarding increasing or decreasing the hash rate is made. On the simulator, the user is also permitted to keep their hash rate steady. Given a Gaussian distribution with variable degrees of control, users can choose to raise, lower, or stabilize the hash rate.

Users can select from two ratios: a hash rate rise ratio that indicates whether the rate at which hashing occurs will increase (or decrease, if needed) and a hashing rate change ratio that displays the proportion of the hash rate change. Algorithm 2 illustrates the change in each node's hash rate. When a floating-point value chosen at random from a Gaussian distribution is less than or equal to the hash rise in rate ratio, the nodes' current hash rate is increased by a percentage of the rate of hash changing ratio; if not, it is decreased by the same percentage. Thus, the introduction of better mining equipment as well as the inclusion or removal of miners from the network can be precisely portrayed in the simulation.

Algorithm 2 Hash rates rising/falling

if the hash rate increase ratio >=random number then

return (existing hash rate * (1 + hash rate change ratio))

else

return (existing hash rate / (1 + hash rate change ratio))

end if

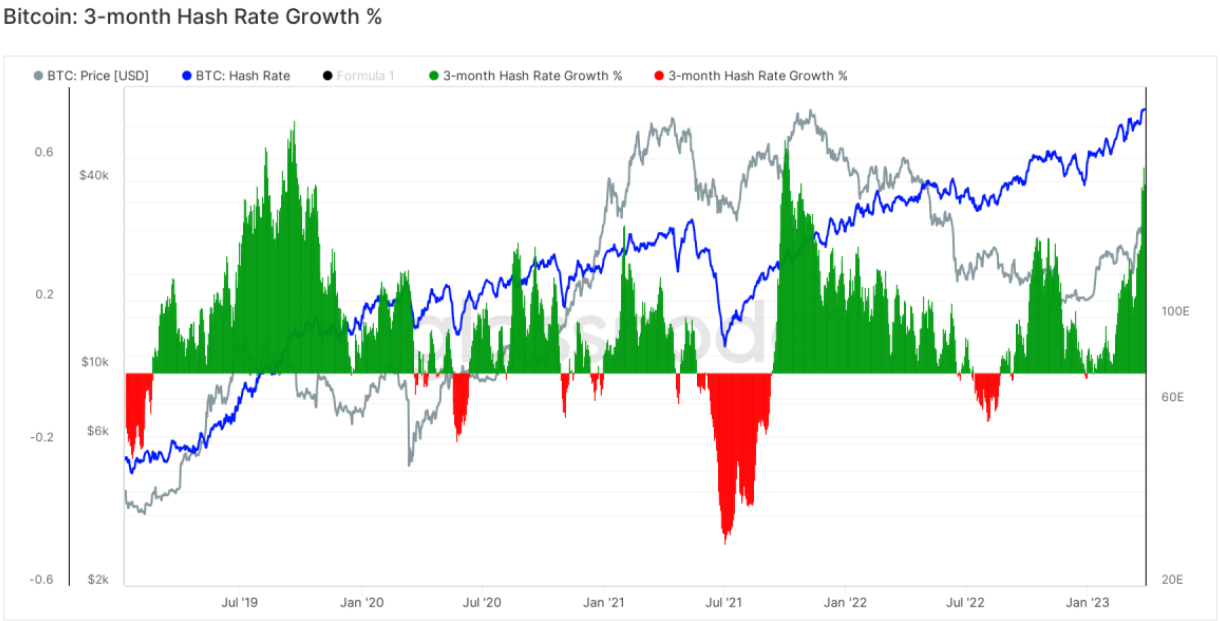

Figure 3 Estimated hash rate of the actual Bitcoin network from July 2019 to January 2023. [5]

5. Implementation and evaluation results

The environment for the recommended method has been built using the programming language. With the aid of the cryptographic hash technique SHA-256, the Proof of Work has been finished. Core coding has been applied to the Genesis block, which is devoid of both a transaction record and a previous hash value. The manager of the following block will always be the miner who establishes the first connection to the system.

This approach has been implemented online using a Docker container. Docker provides a Linux-based container with a network interface. A unique network has been created in Docker, to which all peers will be linked. Ubuntu was utilized with a Core i5-5200U CPU working at 2.2 GHz to complete the implementation. The installed RAM is 4.00 GB. 10% of the total resource has been distributed to each miner to guarantee that they all have the same amount of processing power.

Using its components and resources, a comparable environment has been constructed to compare the test results with the current system. The miners operate independently under this setup. They compete with one another, much like in the current setup, with the winner taking home the whole prize.[21]

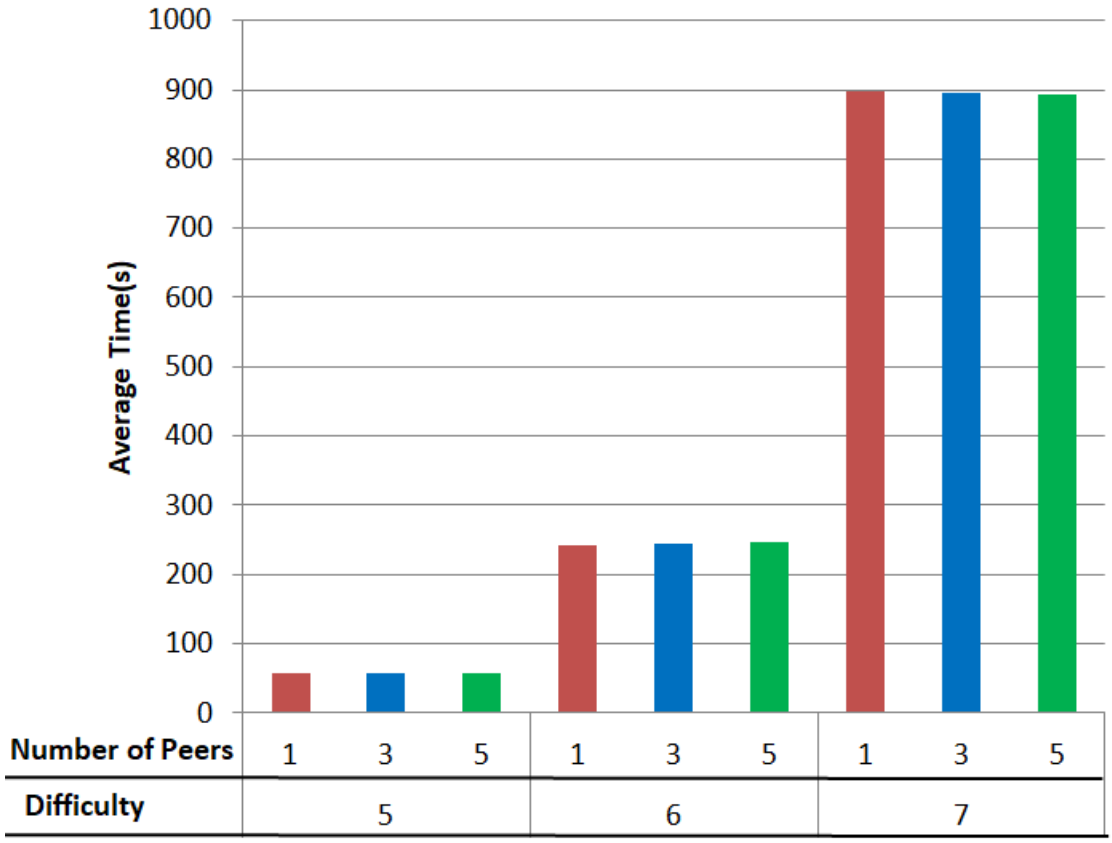

Figure 4: Test result for solo mining [21]

Both solo and parallel mining scenarios have seen the administration of tests with varying numbers of peers and varying degrees of difficulty. Here, the difficulty level indicates the minimum number of consecutive zeros that must begin a valid hash. Based on both solo and parallel mining, the test results are shown in Figures 4 and 5. The Average Time(s) in this case indicates the average number of seconds needed to solve a block. This is calculated by averaging the results of multiple tests conducted under the same circumstances. To find the answer, you need the transaction hash, nonce, timestamp, index, and previous hash as inputs. In this instance, the timestamp and previous hash for a given block are identical for every miner in the solo mining index. For a particular block, all miners have the same transaction hash while mining in parallel utilizing these data.

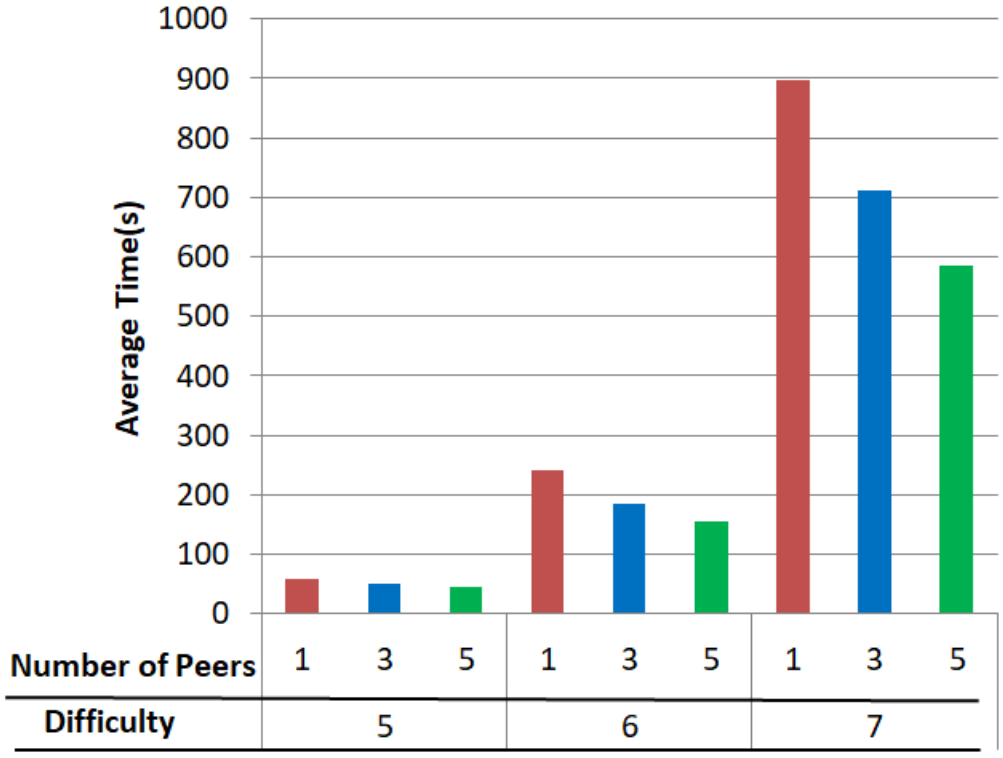

Figure 5: Test result for parallel mining [21]

The difference between mining solo and simultaneously didn't seem to be that great for difficulty levels 1, 2, 3, and 4. Nonetheless, parallel mining has significantly improved for levels 5, 6, and 7. Both the difficulty level and the quantity of peers have an impact on the average time when mining in parallel. The difficulty level alone determines the average time when mining. As the difficulty increases, so does the average time needed. As the number of peers increases, the average time decreases since the miners work in parallel and no two of them complete the same assignment. Another significant discovery is that, regardless of the number of peers, the average amount of time required for one peer in parallel mining is nearly identical to that of solo mining. This is accurate because, despite just one miner working in parallel, no work is finished. [21]

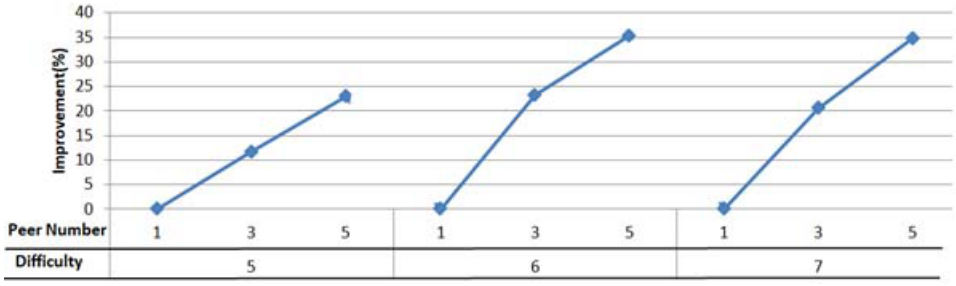

Figure 6: Improvement compared to solo mining [21]

The improvement between solo and simultaneous mining based on test results is shown in Figure 6. When there are five miners instead of just one, the result is 34%. Remember that the findings could change depending on how much processing power the miners are given.

6. Conclusion and future work

We discussed blockchain technology and SimBlock, a simulation tool, in this paper. We found that the distributed ledger could be accurately simulated by the SimBlock simulator. There was also a discussion of a few block, node, and network parameters. We have included the ability to modify Sim Block's hash rate and difficulty adjustment technique. More accurate simulation of blockchain networks is made feasible by the technology. The simulator will be improved in the future. We want to develop a system that supports more contemporary transmission protocols, such as the microblock, even if the existing simulator only imitates a basic block transmission protocol. We also want to offer a simulation of transactions in the future. Sending and receiving transactions, in our opinion, can largely replace some of the techniques used for sending and receiving blocks. Future research will evaluate the simulators based on information that is currently accessible, such as anticipated block sizes, node counts, and hash rates. In the future, we will try to overcome the deficiency of Parallel proof of work and combine it with aggregate signature for better performance of blockchain.

Statement and Declaration:

- **Compliance with Ethical Standards:** This study was conducted by ethical standards for research involving human subjects. Animal subjects were not involved in this study. All participants provided informed consent, and confidentiality and privacy were maintained throughout the study. The authors declare no conflicts of interest that may have influenced the study.

- **Research Data Policy and Data Availability Statement:** All data generated or analyzed during this study are included in this manuscript. The data in this manuscript is based on research done with the help of various research articles, as referenced in the references section. This data can be visible on the journal's home page after publication and to researchers for research purposes only.

- **Competing Interests:** We certify that we have no affiliation with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or material discussed in the manuscript.

7. References

[1] Yusuke Aoki, Kai Otsuki et al. SimBlock: A Blockchain Network Simulator in IEEE Conference,2019.

[3] Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Discovering bitcoins public on topology and influential nodes. 2015.

[4] Arthur Gervais, Ghassan O Karame, Karl W¨ust, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 3–16. ACM, 2016.

[5] https://bitcoinmagazine.com/markets/bitcoin-network-hash-rate-is-rapidly-growing.

[6] Yusuke Aoki, Kai Otsuki, Takeshi Kaneko, Ryohei Banno, and Kazuyuki Shudo. 2019. SimBlock: A Blockchain Network Simulator. In Proceedings of the workshop on Cryptocurrencies and Blockchains for Distributed Systems. ACM, New York, 24–31.

[7] Bellaj Badr, Richard Horrocks, and Xun Brian Wu. 2018. Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger. Packt Publishing Ltd.

[8] Brenn Hill, Samanyu Chopra, and Paul Valencourt. 2018. Blockchain Quick Reference: A guide to exploring decentralized blockchain application development. Packt Publishing Ltd.

[9] Blockchain 2019. Hash Rate: The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing. Retrieved Nov 12, 2019 from https://www.blockchain.com/en/charts/hash-rate?timespan=all.

[10] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16). ACM, New York, NY,

USA, 3–16.

https://doi.org/10.1145/2976749.2978341

[11] B. R. Haverkort, Performance of computer communication systems: a model-based approach. John Wiley & Sons, Inc., 1998.

[12] S. Ferretti and G. D'Angelo, "On the Ethereum blockchain structure: A complex networks theory perspective," Concurrency and Computation: Practice and Experience, vol. 32, no. 12, p. e5493, 2020.

[13] M. Belotti, N. Bo˘zi´c, G. Pujolle, and S. Secci, "A vademecum on blockchain technologies: When, which, and how," IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3796–3838, 2019.

[14] Q. Zhu, S. W. Loke, R. Trujillo-Rasua, F. Jiang, and Y. Xiang, "Applications of distributed ledger technologies to the Internet of things: A survey,"

[15] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah, and M. Omar, "Implementing decentralized auctions using blockchain smart contracts," Technological Forecasting and Social Change, vol. 168, p. 120786, 2021.

[16] N. C. Yiu, "Toward blockchain-enabled supply chain anti-counterfeiting and traceability," Future Internet, vol. 13, no. 4, p. 86, 2021.

[17] N. Angola, V. K. Yadav, S. Venkatesan, S. Verma, et al., "Anonymity on blockchain-based e-cash protocols—a survey," Computer Science Review, vol. 40, p. 100394, 2021.

[18] S. Xu, B. Liao, C. Yang, S. Guo, B. Hu, J. Zhao, and L. Jin, "Deep reinforcement learning assisted edge- terminal collaborative offloading algorithm of blockchain computing tasks for energy internet," International Journal of Electrical Power & Energy Systems, vol. 131, p.107022, 2021.

[19] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, 2008, pp. 1–10.

[20] "The Pioneer's Guide to GX — Decentralized Dependency Management On IPFS". Hacker Noon, 2018, https://hackernoon.com/thepioneers-guide-to-gx-decentralized-dependency-management-on-ipfs-

90064858f4c2 Accessed 4 Nov 2018.

[21] Shihab Shahriar Hazari, Qusay H. Mahmoud, "A Parallel Proof of Work to Improve Transaction Speed and Scalability in Blockchain Systems", 978-1-7281-0554-3/19/$31.00©2019 IEEE.

8. Authors Bibliography:

Ms. Priyanka received his degree in BCA and M.Sc. Computer Science from Maharshi Dayanand University, Rohtak, India, and pursuing a Ph.D. at the Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India. Currently, she is working as an Assistant Professor of Computer Science at Ch. Bansilal University, Bhiwani, Haryana, India. She has published more than 8 research papers in reputed journals and conferences. She also published patents and various book chapters. Her research interests are in Cyber Security, Cryptocurrency, AIoT, etc.

Ritu Makani received his B.E. (Electronics), M. Tech (CSE), and Ph.D. (Network Security and Open-Source Intrusion Detection System). Currently, she is working as an Associate Professor in the Department of Computer Science and Engineering at Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India. She has published more than 25 international journals and conferences. Her area of research interest is network security.