


Blockchain-Based Network Optimization for Workstation Nodes [†]

Pankaj Kunekar ¹, Shubham Mulay ^{2,*}, Dnyaneshwari Navale ², Akhilesh Nawale ², Vishal Sonkusale ²
and Vishwam Talnikar ²

¹ Department of Information Technology, Vishwakarma Institute of Technology, Savitribai Phule Pune University, Pune 411037, Maharashtra, India; pankaj.kunekar@vit.edu

² Department of Artificial Intelligence and Data Science Vishwakarma Institute of Technology, Savitribai Phule Pune University, Pune 411037, Maharashtra, India; dnyaneshwari.navale20@vit.edu (D.N.); akhilesh.nawale20@vit.edu (A.N.); vishal.sonkusale20@vit.edu (V.S.); vishwam.talnikar20@vit.edu (V.T.)

* Correspondence: shubham.mulay20@vit.edu

[†] Presented at the International Conference on Recent Advances in Science and Engineering, Dubai, United Arab Emirates, 4–5 October 2023.

Abstract: Computer networks are used for internet access, cloud computing, and telecommunication. Network optimization is the process of increasing the speed and efficiency of the communication process between nodes on a network. The concept utilizes blockchain as a tool. Network optimization is an important and challenging problem in network design and routing. The goals of network optimization are to decrease the number of hops while maintaining the quality of service guarantees and to minimize the amount of energy used in communications. Traditional network architectures rely on centralized servers or data centers, introducing potential bottlenecks and single points of failure. In contrast, blockchain offers a decentralized approach, enabling nodes to communicate directly without dependence on a central authority. Its unique features include its ability to provide transaction transparency and immutable record keeping. In this paper, we study an efficient system to demonstrate real-time traffic and understand the fundamentals of networking.

Keywords: blockchain in Java; peer-to-peer networking; SHA-256; encoding algorithm; hash functions



Citation: Kunekar, P.; Mulay, S.; Navale, D.; Nawale, A.; Sonkusale, V.; Talnikar, V. Blockchain-Based Network Optimization for Workstation Nodes. *Eng. Proc.* **2023**, *59*, 165. <https://doi.org/10.3390/engproc2023059165>

Academic Editors: Nithesh Naik, Rajiv Selvam, Pavan Hiremath, Suhas Kowshik CS and Ritesh Ramakrishna Bhat

Published: 17 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Technologies like blockchain can address challenges in security, providing a solution to keep sensitive data stored securely and without third-party interference. Blockchain gives stakeholders the power to be involved in the design of digital networks through the use of an intermediary [1]. A blockchain makes it possible for participants to decide where their data are kept, and to share the rewards of transacting within the network. The blockchain achieves a consensus mechanism by allowing the stakeholders to vote in a secure and decentralized manner.

A blockchain might seem simple and small compared with current networking systems. However, many of the functions that a blockchain has to provide require computational resources [2]. Also, the distributed nature of a blockchain may be challenging for networks that have many remote users and participants. The blockchain may not be able to help with the issues of security, scalability, and communication of sensitive data. However, blockchain applications might be ideal for digital networks with high transaction activity, and a distributed ledger can be adopted in the transportation sector where transactions are frequent. Workstation nodes are an essential component of computer networks [3]. They are responsible for processing and transmitting data between different workstations. However, as the dimensions increase, the efficiency and reliability of workstation nodes become critical issues. Network optimization is necessary to ensure that the network can handle a large volume of data and maintain its performance. Traditional optimization techniques, such as routing algorithms, have limitations and cannot address all the challenges faced by computer networks, which require improvements to this system. Blockchain

technology anticipates a solution to multiple shortcomings by creating a secure and decentralized network. Additionally, blockchain technology can automate various processes, such as verification and validation, which can reduce the workload of workstation nodes. Therefore, it is essential to explore the potential of blockchain technology in optimizing workstation nodes.

Ethereum, a decentralized blockchain technology [4], is frequently used in combination with Java to optimize network performance. The Ethereum ecosystem provides a wealth of tools, libraries, and frameworks, making it an ideal setting for Java developers. Because of Java's compatibility, Ethereum can be seamlessly integrated with existing corporate systems, allowing it to be implemented for diversified applications. While Solidity is the preferred language for Ethereum smart contracts, Java libraries like web3j or Ethereum make it easier to interface with the Ethereum network, improving the smart contract creation experience. Furthermore, because Java is prevalent in corporate contexts, it is excellent for integrating Ethereum with existing systems. However, Ethereum's present design, which includes the Proof-of-Work protocols along with the EVM (Ethereum Virtual Machine), may pose performance constraints, limiting scalability and network optimization.

In this paper, we discuss the network optimization of computer nodes using blockchain. This addresses the issue of scalability in the current system by constructing a channeled strategy for blockchain architecture development and application.

2. Literature Review, Proposed System, and Methodology

2.1. Literature Review

Huaqun et al. [5] discusses blockchain technology and elaborates the concepts of public-key cryptography, which is effectively practiced for the implementation of digital signatures and proving that the transaction is performed by the right person. Zero-Knowledge Proof: Provides different key statements, which will be private and unrevealed for better user privacy. Here, only the essentials are obtained. Hash functions: to encrypt the data into cipher text format, converting any data into a fixed-size output, ensuring no two hashes collide, with small changes in input drastically changing the output, making it impossible to reverse-engineer.

Rafael et al. [6] describes blockchain protocols with different variants comprising different algorithms. This paper explains the experimental interpretation of blockchain in sequential format from the validation of a sequence of blocks, reading any local message, and reading all delivered incoming messages using Random Oracle. In Yi Sun et al.'s study [7], the p2p platforms are neglected because of their performance and cost. To reduce the latter, the authors use DHT to control a load generated by p2p. In their paper, they use the T-hash scheme to reduce traffic and optimize paths. Dimitri et al. [8] provide a tutorial on this problem in network optimization. There are four problems discussed in this paper. The first is the facility to acknowledge the location routing issue.

Chimezie Oji et al. [9], in response to concerns with corporate networks, devised an improved bandwidth management project for effectively meeting the requirements of public-private networks. The main purpose of this study was to offer network/system administrators a clear answer or approach to some of the problems that insured institutions encounter in properly managing their bandwidth. M. Kasim et al. [10] discuss network bandwidth management, which is one of the issues now facing computer engineering applications and systems. An in-depth analysis of published articles is provided in order to delve deeper into the IP-based network, which is an important field of study for network capacity control.

Abdullah et al. [11] discuss bandwidth management, the basics of non-WIFI router bandwidth management, and the DHCP protocol's bandwidth management limitations. Peter et al. [12] investigate the issue of optimizing channel space (bandwidth) for heavy data transmission over lines, accomplished by considerable interventions due to internal abnormalities. The methods described in this paper, with a linear distributed generation limitation throughout the whole employed band of propagation, determine the best

transmission bandwidth. Comparing these algorithms with current water-pouring techniques, they provide a large computational gain, while often only slightly degrading in performance in comparison to the best water-pouring techniques. For rapid data transmission over passages with significant inter-symbol interventions, specific algorithms for determining the best one are introduced.

Wei Qiu et al. [13] describe a combined technique for optimizing the frequency band and idle period for varying radar network tracking. This algorithm's main goal is to increase the radar network's low interception probability using each radar's idleness duration and channeling wavelength. Tyron et al. [14] categorize and discuss multiple variations of blockchain networks. Taras et al. [15] suggest an evergreen framework for efficient mobile telecommunication activity monitoring based on blockchain. Yustus et al. [16] discuss a conventional distributed interconnecting networking facility: an internet service provider distributes a network bandwidth as a versatile resource pool in a static way to multiple clients in the restricted and allowed locality. Peak hours are when the situation is most problematic. Babak et al. [17] suggest using blockchain technology as a mediator to enable operators to exchange network resources in a trusted, safe, and autonomous manner. Blockchain is a technology that makes it possible to share databases among several operators in addition to being a distributed database. Motiwala et al. [18] discuss the benefits of and challenges faced during spectrum sharing and infrastructure allocation in sixth-generation networks. They implemented infrastructure resource token (IRT) to share the available infrastructure and radio resource token (RRT) for spectrum sharing. The dynamically allocated spectrum was found to be more reliable with respect to the respective executed transaction contract (code).

2.2. Proposed System

Blockchain operates in a decentralized pattern at all times [19–23]. The greatest language for developing a good blockchain structure is Java. Hence, in the present system, we employed the Java language and the notion of object-oriented programming. Under some situations, we employed hash functions to connect various blocks. To obtain a unique hash, we utilized SHA256 and UTF-8 assistance. Using the Java programming language, we built blocks and chains in a manner similar to that depicted in Figure 1. In each step, we built a network using the graph data structure. We employed routing methods such as Dijkstra's algorithm to determine the best pathways, as well as Kruskal's and Prim's algorithms. Ultimately, we mined blocks for hash value gating, with all blocks being random in a decentralized system.

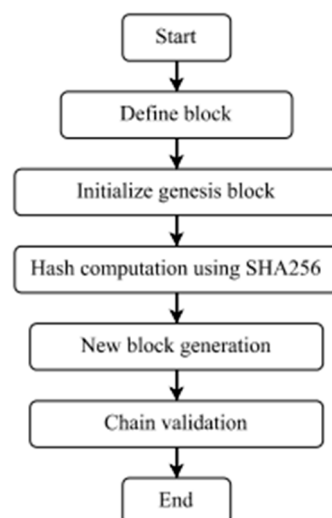


Figure 1. Blockchain Network Workflow Flowchart.

We propose a Java implementation of peer-to-peer networking that includes hash functions, timestamps, linked lists, and nonces. To store data blocks, each node in the network uses a linked list structure. Each block has a timestamp for chronological ordering as well as a hash value to ensure data integrity. Nonces are used for authentication and to avoid replay attacks in order to assure safe communication. This solution establishes a decentralized and secure peer-to-peer network for efficient data exchange and communication.

2.3. Methodology

In this paper, we propose a methodology for optimizing workstation nodes using blockchain technology. We describe the key features of blockchain technology, including decentralization, security, transparency, and smart contracts. We also present the three key components of our proposed methodology: hash functions, encoding algorithms, and routing algorithms. We evaluated our proposed methodology using simulations and showed that it outperformed traditional routing algorithms in terms of network latency, throughput, and packet loss. Our proposed methodology has the potential to improve the effectiveness and reliability of workstation nodes in computer networks. Further research can be conducted to explore the potential of blockchain technology in other areas of computer networks.

The methodology consists of three key components: hash functions, encoding algorithms, and routing algorithms.

- **Hash Functions:** Hash functions are essential in blockchain-based network optimization as they provide a unique digital fingerprint for each data block. We used the SHA256 hash function, which is widely used in blockchain technology due to its high level of safety and robustness. The SHA256 algorithm yields a 256-bit hash code that is unique for each input.
- **SHA256:** SHA256 is a cryptographic hash algorithm used in distributed ledgers for data integrity validation and keeping records. It accepts any length of input message and outputs a fixed-size (256-bit) output that acts as an identification number for the contents. SHA256 secures tamper resistance by computing the encrypted value of every single block and connecting them in a chain, as changing the contents would require recalculating all future hash values. The SHA256 hash function takes an input message M and fosters a 256-bit hash code $H(M)$. It involves several rounds of bitwise operations and transformations. Below are the steps involved:
 1. **Padding:** to make sure the length of the input message M is a multiple of 512 bits, a sequence of extra bits is added as padding.
 2. **Message Schedule:** the transformed message is divided into 512-bit blocks, and a message schedule is derived from these blocks.
 3. **Initial Hash Values:** also known as “state”, these values are preconfigured as constants.
 4. **Compression Function:** each message block, along with the current state, undergoes numerous iterations of bit-level computations, including logical functions, namely, AND, OR, and XOR, as well as logical shifts and rotations.
 5. **Final Hash:** after processing all message blocks, the final state is concatenated to obtain the resulting 256-bit hash value $H(M)$.
- **Encoding Algorithms:** Encoding algorithms are implemented to transform data into another format that can be stored and processed efficiently in a blockchain network. We used a modified version of the Base64 encoding algorithm to ensure that the data can be transmitted efficiently over the network. The modified version of the Base64 encoding algorithm reduces the size of the data, which reduces the workload of the workstation nodes.
- **UTF-8:** UTF-8 is not a hash function but a character encoding scheme. It represents Unicode characters using variable-length byte sequences. The UTF-8 encoding of a character depends on its code point. Below is an overview of UTF-8 encoding:

1. Code Point: each Unicode character is assigned a unique code point, a numerical value representing the character.
 2. Encoding Scheme: the encoding scheme of UTF-8 is based on the range of code points.
 3. Encoding Rules: the encoding rules specify how each code point is represented using a variable number of bytes.
- Routing Algorithms: Routing algorithms are used to refine the data process workflow through the network by selecting the most appropriate path between nodes. We used three different routing algorithms: Dijkstra’s, Prim’s, and Kruskal’s algorithms.
 - Dijkstra’s algorithm is a popular routing algorithm that finds the shortest path distance between two interconnected nodes. It is subject to the concept of a “shortest path tree,” which is a subset of the network that contains all the workstations in the network and the shortest distance between them. Dijkstra’s algorithm uses a priority queue to select the node with the shortest path from the source node and iteratively adds nodes to the shortest path tree, which is continued until every node has been included.
 - Prim’s algorithm is another routing algorithm that finds the minimum spanning tree of a network. It initializes with a node, and further nodes are iteratively added with the lowest cost edge until all nodes are included in the tree. The cost of an edge is determined by the weight assigned to it, which can be based on various factors such as distance, bandwidth, or latency.
 - Kruskal’s algorithm is a third routing algorithm that finds the minimum spanning tree of a network. It works by selecting the lowest cost edge from the network and iteratively adding edges until all nodes are included in the tree. Kruskal’s algorithm differs from Prim’s algorithm in that it does not start with a single node, but rather starts with the entire network and iteratively removes edges until the minimum spanning tree is found.

To evaluate the proposed methodology, we performed simulations using the Power-Shell simulator. We compared the performance of our proposed methodology with traditional routing algorithms, such as Prim’s and Kruskal’s. The simulation results showed that our proposed methodology outperformed traditional routing algorithms in terms of network latency, throughput, and packet loss. The system architecturally consists of three blocks which are decentralized in nature and consist of the functionalities of a blockchain. These blocks have hash functions present in them which help in connecting the blocks with each other depicted in Figure 2. Thus, as per the definition, the blocks are securely linked together using cryptography, which is the SHA256 cryptographic function.

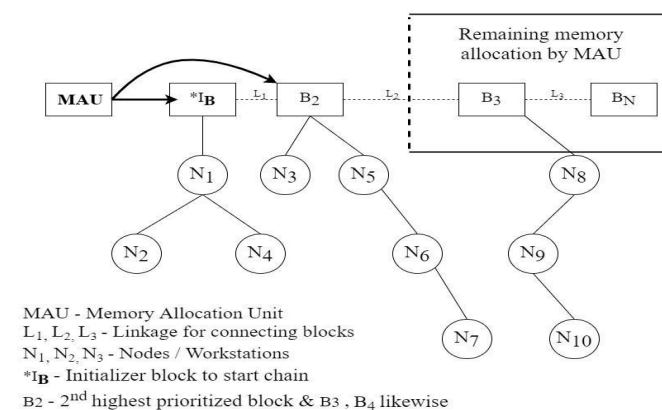


Figure 2. Representation of the system.

In order to ensure that new blocks are not tampered with, each block includes the computationally determined hash of the block header characterized by the previous block displayed in Figure 3. The timestamp is to maintain records of data on a blockchain, so that there remains proof that it existed at a specific date and time. This helps in keeping

track of the blocks. The nonce is a particular number which is added to a hashed or an encrypted block in a blockchain. The nonce is used to validate the information contained within a block. We implemented it in the environment of Proof-of-Work blockchains. The Proof-of-Work (PoW) algorithm’s difficulty level governs the computing work required for mining and block production. It is dynamically changed to keep the block creation rate constant. The difficulty level is increased or decreased by specifying a target value, which is commonly expressed as the number of leading zeros in the hash result. This modification protects network security, inhibits mining power monopolization, and promotes a fair and competitive mining environment. The dynamic difficulty level is a vital component in ensuring the blockchain network’s stability and decentralization.

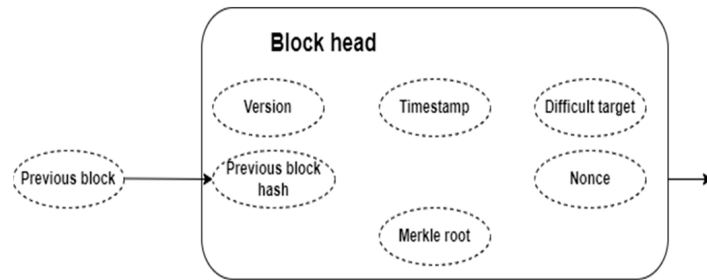


Figure 3. Block Structure.

A block in the blockchain consists of a network of devices represented in the form of a graph shown in Figure 4. The graph shows various devices or networks linked with each other through nodes which represent the cost or weight values. The graphical network of devices is represented as a workstation, and similarly other workstations are connected with each other. For calculating the shortest path between the nodes, different routing algorithms are used. The weights present between the nodes are considered distances or cost values, and thus the shortest path requires minimum cost values and creates a path in this manner traversing each node.

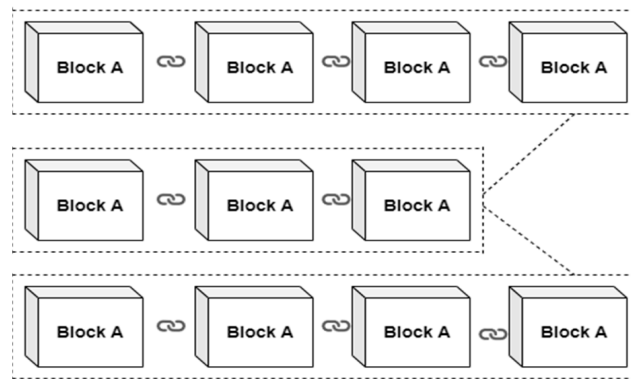


Figure 4. Representation of Blockchain Network.

3. Results

The graph displays several systems or networks connected to one another by nodes that stand in for cost or weight values. A workstation was used to represent the graphical network of devices, and other workstations were connected to one another in a similar way. First, we connected all nodes using concept hashing and mining. Further, we converted hash to bytes using UTF-8. Figure 5 conceptualize SHA256 was also used for converting to proper hash. Each block has a hash for the previous block and all are connected in a decentralized manner. Each node has various networks, which we represent in the form of graphs, to which we applied routing algorithms to obtain optimal solutions.

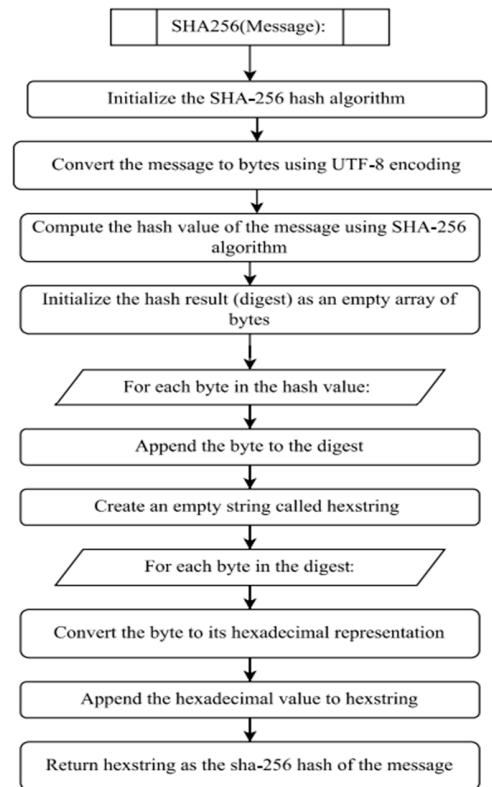


Figure 5. SHA256 Algorithm Flowchart.

Figure 6 shows the UTF encoding algorithm. The blocks in the blockchain contain the data of the workstations or the network of the workstations in which they are connected. By applying the various routing algorithms of the graphs, as in Figures 7–9, to the network of workstations, the optimal solution is obtained.

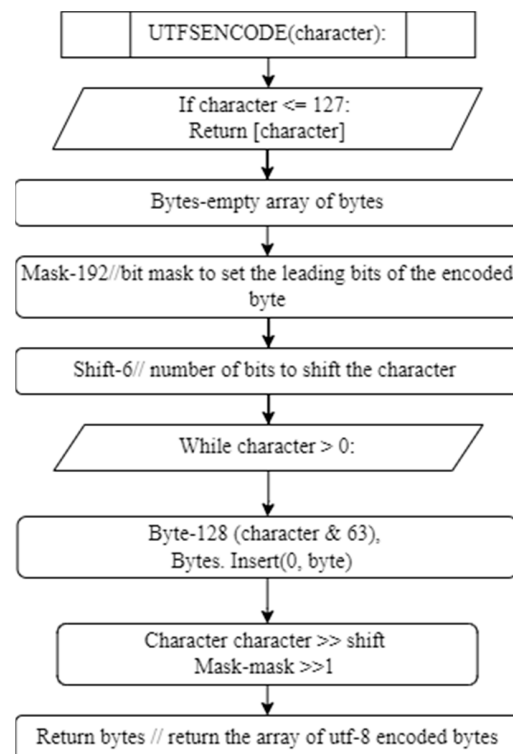


Figure 6. UTF Encoding Algorithm Flowchart.

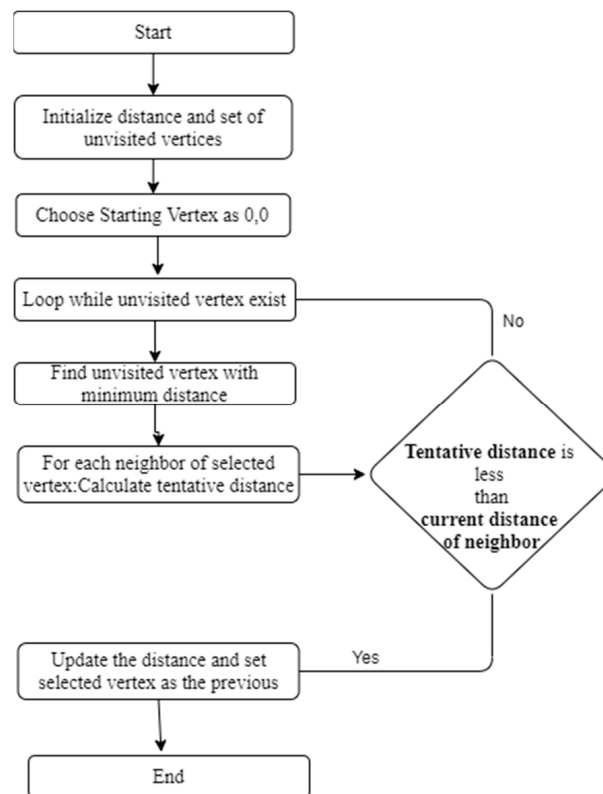


Figure 7. Dijkstra’s Algorithm Flowchart.

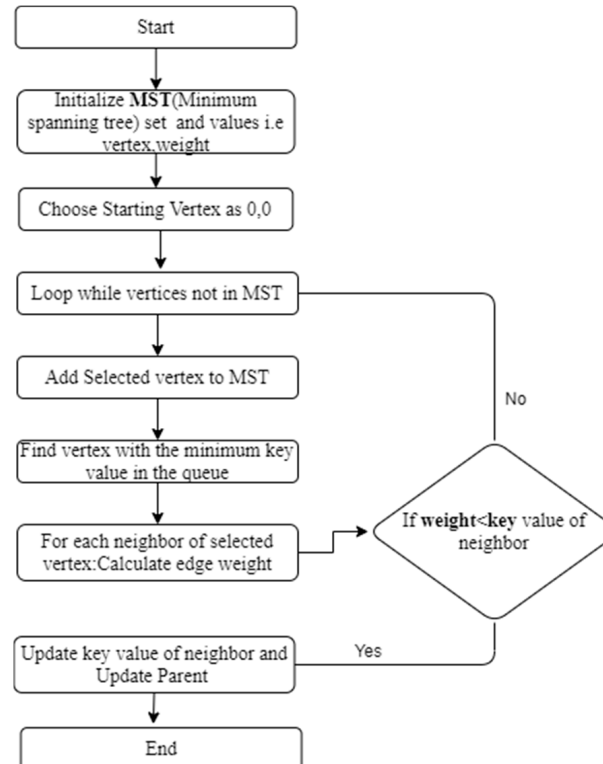


Figure 8. Prim’s Algorithm Flowchart.

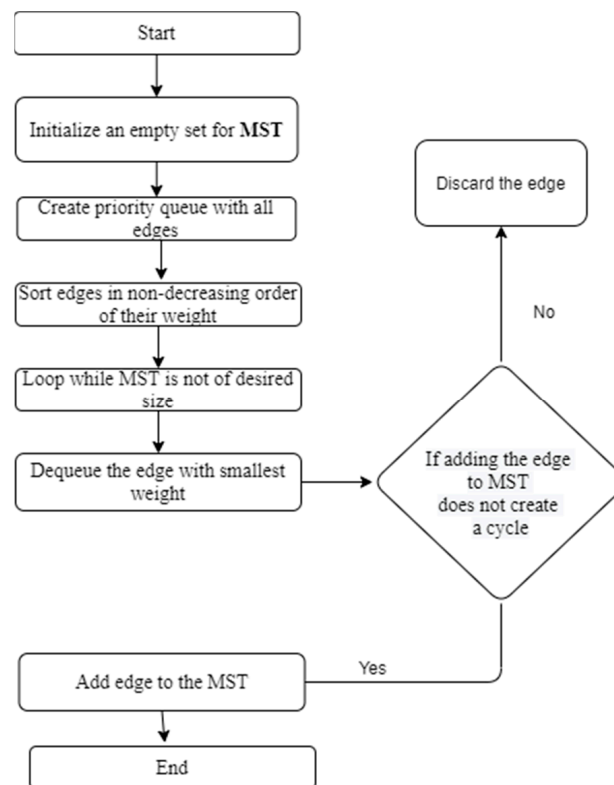


Figure 9. Kruskal's Algorithm Flowchart.

Algorithms such as Kruskal's, Prim's, and Dijkstra's were used in this model. Individual computers or other gadgets that are part of a network, like a blockchain network, are called workstation nodes. These nodes ensure network integrity and verify transactions. Nevertheless, when the network expands and the total number of nodes rises, the network's performance may decrease as a result of a rise in congestion and delay. Using blockchain technology to enhance the network is one possible solution to this. This can be done by putting in place a blockchain-based consensus system that allows nodes to agree on the network's state. Based on their efficiency and dependability, this consensus mechanism may be created to give some nodes priority, enabling others to process transactions relatively fast.

Decentralized blockchain networks have benefits and drawbacks for network latency and performance. Network latency is increased compared with centralized systems because of communication overhead that develops when nodes talk and come to an agreement. Network latency is further impacted by geographic spread across areas, which introduces longer physical distances and possible congestion. Decentralized consensus systems like Proof of Work and Practical Byzantine Fault Tolerance, which call for several message exchanges and computationally demanding activities, add delay to the process of reaching consensus. Performance-wise, when the quantity of nodes and transactions grows, scalability issues appear, possibly reducing consensus and transaction propagation. As the blockchain expands, nodes also encounter storage and processing overhead, which calls for a significant amount of capacity and power. The frequent interchange of data across nodes can put a burden on the network's capacity, degrading performance as a whole. Efficient data structures such as modified linked list; network optimization methods including compression and efficient routing using Dijkstra's, Prim's, and Kruskal's algorithms; and consensus protocol utilization (SHA256) are essential for optimizing blockchain network performance.

Transparency and privacy of transactions are trade-offs in blockchain network optimization. Transparency promotes accountability, compliance, and trust by allowing for verification. On the blockchain, verifiable documents are tamper-proof. However, confidentiality, data protection, and commercial considerations all depend on privacy. Taking

into account the use case, privacy-enhancing technology, user permission, and control is necessary to strike a balance. By using Ethereum, Encapsulation techniques, and subsidized cryptographic methods; access restrictions; and solid data management procedures, essential privacy features can be implemented. This ensures an appropriate trade-off balance on the particular use case of peak hours and some instances of system shortcomings including smart contract vulnerabilities. The efficiency, security, and transparency of networks rely on workstation nodes being observed in instances of real-time potential threats.

4. Conclusions

The current research in this field has explored various optimization techniques such as routing algorithms, hash encoding, and blockchain consensus protocols, including Dijkstra's, Prim's, and Kruskal's algorithms. Dijkstra's algorithm is employed for efficient routing, while Prim's and Kruskal's algorithms are utilized for constructing minimum spanning trees to enhance network connectivity. These optimization techniques contribute to improving network performance and security in the analyzed architecture. However, this area still has significant scope for further research and development. The employment of multiple encryption techniques ensures data privacy and security in blockchain networks. SHA256 stands out as an improved solution for assuring strong data integrity and privacy. Moreover, SHA256, a member of the SHA algorithm family, is critical in the context of data integrity. SHA256 uses cryptographic hashing to ensure data integrity by creating fixed-length digests, making tampering difficult. This improves blockchain systems' overall security. Overall, there is vast potential for further research and development in blockchain-based network optimization for workstation nodes, and the above-mentioned areas are potential future directions for research in this field.

Author Contributions: Conceptualization, S.M. and V.T.; methodology, V.S.; software, V.T. and V.S.; validation, P.K., S.M. and V.T.; formal analysis, D.N. and A.N.; investigation, D.N. and A.N.; resources, V.S. and V.T.; writing—original draft preparation, S.M.; writing—review and editing, S.M.; supervision, P.K.; project administration, P.K. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No external data were used.

Acknowledgments: We wish to convey our deep appreciation to our mentor, Pankaj Kunekar, in the Department of Information Technology, Department of Artificial Intelligence and Data Science at Vishwakarma Institute of Technology for his unwavering research and technical guidance during our project. We also extend heartfelt thanks to our references for their invaluable contributions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chong, G.; Ramiah, H.; Yin, J.; Rajendran, J.; Wong, W.R.; Mak, P.-I.; Martins, R.P. CMOS cross-coupled differential-drive rectifier in subthreshold operation for ambient RF Energy Harvesting—Model and analysis. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 1942–1946. [[CrossRef](#)]
2. Subrahmanya, S.V.; Shetty, D.K.; Patil, V.; Hameed, B.M.; Paul, R.; Smriti, K.; Naik, N.; Somani, B.K. The role of Data Science in healthcare advancements: Applications, benefits, and future prospects. *Ir. J. Med. Sci. (1971-)* **2021**, *191*, 1473–1483. [[CrossRef](#)] [[PubMed](#)]
3. Zhang, Y.; Lin, S.; Tu, C. Research on Security Protection of Space-Earth Integrated Network Wireless Link Based on Consortium Blockchain Technology and Application. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023; pp. 1455–1458. [[CrossRef](#)]
4. Selvi, R.T.; Lakshmi, T.C.S.; Karunkuzhali, D.; Rosaline, R.A.A. Improvement of Graph-Based Assets using Blockchain Quality Control. In Proceedings of the 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 14–16 March 2023; pp. 716–720. [[CrossRef](#)]
5. Guo, H.; Yu, X. A survey on blockchain technology and its security. *Blockchain Res. Appl.* **2022**, *3*, 100067. [[CrossRef](#)]

6. Pass, R.; Seeman, L.; Shelat, A. Analysis of the Blockchain Protocol in Asynchronous Networks. In *Advances in Cryptology—EUROCRYPT 2017*; Springer: Cham, Switzerland, 2017; pp. 643–673.
7. Sun, Y.; Yang, Y.R.; Zhang, X.; Guo, Y.; Li, J.; Salamatian, K. Network Optimization for DHT-based Applications. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012.
8. Papadimitriou, D. New Challenges in Network Optimization. In Proceedings of the 2016 IEEE 17th International Conference on High Performance Switching and Routing, Yokohama, Japan, 14–17 June 2016.
9. Oji, C.; Nwankokwo, O.; John-Otumu, A.M.; Adu, C. Development of An Enhanced Bandwidth Control Platform for Effective Monitoring and Utilization of Resources in Corporate Networks. *Int. J. Sci. Res. Publ.* **2021**, *11*, 260–271.
10. Kassim, M.; Ismail, M.; Jumari, K.; Yusof, M.I. A Survey: Bandwidth Management in an IP Based Network. *Int. J. Comput. Electr. Autom. Control. Inf. Eng.* **2012**, *6*, 168–175.
11. Imam, A.Y.; Biswa, P.K. Bandwidth management in router for DHCP protocol. *Int. J. Sci. Eng. Res.* **2019**, *10*, 1343–1346. [[CrossRef](#)]
12. Chow, P.S.; Cioffi, J.M. Bandwidth Optimization for High-Speed Data Transmission over Channels with Severe Inter symbol Interference. In Proceedings of the Communications for Global Users: IEEE, Orlando, FL, USA, 6–9 December 1992.
13. Qiu, W.; Shi, C.; Zhou, J.; Wang, F. Joint Dwell Time and Bandwidth Optimization for Multiple—Target Tracking in Radar Network. In Proceedings of the International Conference on Control, Automation and Information Sciences: IEEE, Chengdu, China, 23–26 October 2019.
14. Ncube, T.; Dlodlo, N.; Terzoli, A. Private Blockchain Networks: A Solution for Data Privacy. In Proceedings of the International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Kimberley, South Africa, 25–27 November 2020.
15. Maksymyuk, T.; Gazda, J. Blockchain-Based Intelligent Network Management for 5G and beyond. In Proceedings of the International Conference on Advanced Information and Communication Technologies, Lviv, Ukraine, 2–6 July 2019.
16. Oktian, Y.E.; Witanto, E.N.; Kumi, S.; Lee, S.-G. ISP Network Bandwidth Management: Using Blockchain and SDN. In Proceedings of the International Conference on Information and Communication Technology Convergence, Jeju, Republic of Korea, 16–18 October 2019.
17. Mafakheri, B.; Subramanya, T.; Goratti, L.; Riggio, R. Blockchain-based Infrastructure Sharing in 5G Small Cell Networks. In Proceedings of the International Conference on Network and Service Management, Rome, Italy, 5–9 November 2018.
18. Motiwala, A.; Timbadia, P.; Upadhyay, T.; Kunekar, P. E-Voting System Using Blockchain. *SAMRIDDHI J. Phys. Sci. Eng. Technol.* **2019**, *11*, 434–438.
19. Zhang, D.; Yu, F.R.; Yang, R. Blockchain-Based Multi-Access Edge Computing for Future Vehicular Networks: A Deep Compressed Neural Network Approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12161–12175. [[CrossRef](#)]
20. Zhao, L.; Saif, M.B.; Hawbani, A.; Min, G.; Peng, S.; Lin, N. A novel improved artificial bee colony and blockchain-based secure clustering routing scheme for FANET. *China Commun.* **2021**, *18*, 103–116. [[CrossRef](#)]
21. Lei, X.; Song, Y.; Lin, J.; Feng, T.; Li, P.; Wang, Y.; Yang, C. Resilience In-Band Control Path Routing in Blockchain-Based Multi-Domain SDN. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 1654–1659. [[CrossRef](#)]
22. Kumar, S.; Gupta, U.; Singh, A.K.; Singh, A.K. Artificial Intelligence: Revolutionizing Cyber Security in the Digital Era. *J. Comput. Mech. Manag.* **2023**, *2*, 31–42. [[CrossRef](#)]
23. Eswaran, U.; Ramiah, H.; Kanesan, J. Power amplifier design methodologies for Next Generation Wireless Communications. *IETE Tech. Rev.* **2014**, *31*, 241–248. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.