Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 6, No.1, January 2024, pp. 02401018-01 ~ 02401018-08
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v6i1.17878

The Implementation and Analysis of The Proof of Work Consensus in Blockchain

Alvin Christian Davidson Therry*, Rizka Ardiansyah, Muhammad Yazdi Pusadan, Yuri Yudhaswana Joefrie, Anita Ahmad Kasim

Information Technology Department, Faculty of Engineering, Tadulako University, Jl. Soekarno Hatta No KM 9 Palu 94148, Central Sulawesi, Indonesia

*alvintherry20@gmail.com

Abstract. Communication in peer-to-peer (P2P) networks presents challenges in maintaining security, data integrity, and decentralization. Consensus mechanisms play a crucial role in addressing these challenges by validating data and ensuring that each entity has synchronized data without intermediaries. This research focuses on the implementation and analysis of the Proof of Work (PoW) consensus mechanism, widely used in blockchain, to enhance understanding of its functions, benefits, and workings or flow. This research, conducted using the Go programming language, successfully implements Proof of Work (PoW) as a security measure, ensuring data integrity, and preventing manipulation. Through black-box testing, this research confirms the functionality and reliability of the implemented Proof of Work (PoW) consensus. These findings contribute to a deeper understanding of consensus mechanisms, offering insights to optimize blockchain protocols and foster trust among entities. This research highlights the relevance of sustainable Proof of Work (PoW) in blockchain technology, emphasizing its role in enhancing security and ensuring data integrity in decentralized networks.

Keywords: Blockchain, Consensus, Proof of Work, Security, Decentralized

(Received 2023-12-20, Accepted 2023-12-27, Available Online by 2024-01-04)

1. Introduction

The development of blockchain usage in recent years has experienced a very significant increase across various fields. This significant increase is because blockchain offers several advantages and solutions to some problems in different sectors. Blockchain is a distributed database technology that operates through a peer-to-peer (P2P) network, meaning the database is distributed among all entities in the network, and communication between entities occurs directly without intermediaries [1], [2].

Communication through peer-to-peer (P2P) networks poses significant challenges in maintaining security, data integrity, consistency, and decentralization. A mechanism is needed to validate and prevent data falsification and manipulation, ensuring that each entity has identical and synchronized copies of the database without intermediaries [3]–[6]. This mechanism is called consensus.

Consensus is a key in blockchain that operates on each entity within the network. Consensus functions to validate data before it is added as a new block to the blockchain, ensuring that each entity has synchronized data and without intermediaries [7]–[10]. Consensus is a solution to existing challenges of maintaining security, data integrity, consistency, and decentralization in the blockchain. Proof of Work (PoW) is one of the most widely used consensus today and is also known as the first consensus applied in blockchain. One example of the use of Proof of Work (PoW) is in the Bitcoin network, which aims to verify and secure transactions and create new blocks in the blockchain by harnessing computational power.

This research aims to implement and analyze Proof of Work (PoW) consensus on blockchain. The results of this research are expected to increase knowledge and insight about consensus, especially about proof of work and its implementation in blockchain.

To guide readers through the content of this paper, the following is the structure that will be followed. The next section will delve into the Methods, discussing the research type employed and the system development undertaken. Subsequently, the paper will move on to the Results and Discussion section, covering planning, analysis, system design, prototype, implementation, and system testing. The paper will conclude with a summary and suggestive recommendations for future developments in this field.

2. Methods

2.1. Research Type

This research employs exploratory research. Exploratory research is a type of scientific investigation conducted to explore a topic without specific hypotheses. This research aims to broaden insights and gain a deeper understanding of a topic that is not well explored, thus providing a foundation for future research [11].

2.2. System Development

The system development is carried out using the prototype method. This method is chosen because it allows for easy changes and adjustments during the development process, making it flexible and providing a visual or interactive representation of the system's flow before development begins. This representation enables developers to see and understand the system's requirements better throughout the design process and allows for development adjustments [12].

The main technology used in developing the system is the Go programming language. The Go programming language creates the blockchain and Proof of Work (PoW) consensus due to several advantages that support blockchain, such as fast performance, parallelism, clean and easily understandable syntax, efficiency in resource usage, and more [16]. Supporting technologies include BadgerDB, gRPC, and Docker. BadgerDB is a key-value database for each blockchain entity because it is designed with high performance, storage efficiency, and other features. gRPC for communication between blockchain entities due to its advantages that align with communication needs in blockchain, including efficiency, lightweight, strong security and authentication, using Protocol Buffers (Probuf), asynchronous communication, and more. Docker to run multiple services in the blockchain network, simplifying the configuration process.

3. Results and Discussion

3.1. Planning

At this stage, the process begins with identifying issues and conducting a literature review on security and decentralization systems within the blockchain. This leads to the consensus mechanisms as a solution to these problems. Based on these issues, the researcher aims to implement and analyze the Proof of Work (PoW) consensus within the blockchain to gain a deeper understanding and further insights into consensus, especially Proof of Work (PoW), within the blockchain.

3.2. Analysis

At this stage, the process involves analyzing aspects related to consensus. The main focus is on the functions and benefits of consensus in blockchain, the workings or flow of Proof of Work (PoW), and the strengths and weaknesses of Proof of Work (PoW). Based on the analysis conducted, consensus in blockchain functions as a fundamental mechanism to achieve agreement among distributed entities, ensuring the validity and consistency of transactions. It prevents data tampering and secures the integrity of decentralized data. Proof of Work (PoW) is a consensus algorithm in which entities solve complex mathematical puzzles, validate transactions, and add blocks to the blockchain. Its strength lies in providing robust security, decentralization, and a proven track record in networks. However, the energy-intensive computations of Proof of Work (PoW) raise environmental concerns. Despite these weaknesses, Proof of Work (PoW) remains a reliable and secure consensus mechanism.

3.3. System Design

During this stage, the implementation of the Proof of Work (PoW) consensus in blockchain is designed. The system design starts when an entity receives data to be added as a new block. This data, along with the prev hash, nonce, and difficulty, undergoes hashing using the SHA256 algorithm. Prev hash is the hash of the previous block. Nonce is a value that can be changed by the entity to achieve the desired hash value. Difficulty is the level of difficulty that an entity must overcome to add a new block to the blockchain. Subsequently, validation occurs by checking if the resulting hash of the block is less than the target value. If true, a new block is added and disseminated to other entities in the blockchain network. If false, the process iterates by incrementing the nonce and repeating the hashing and validation steps until a valid node is found [13]–[15]. The system design can be seen in Figure 1.

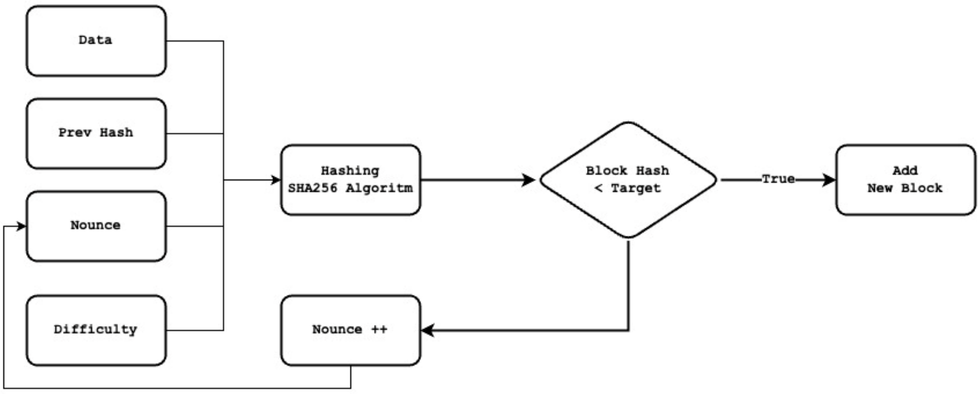


Figure 1. Design System

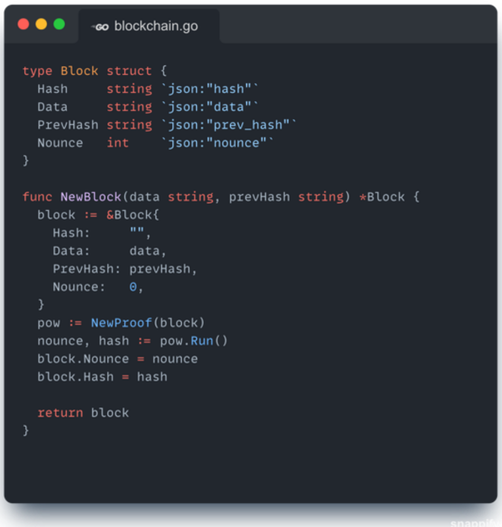
3.4. Prototype

This stage aims to implement the proof of work consensus flow, which involves hashing and validation to add a new block to the blockchain through prototype design, enhancing the understanding of system requirements.

3.5. Implementation

In this stage, the system design is implemented in the form of code using the Go programming language. Go is a programming language developed by Google, commonly used on the server side, and possesses several advantages such as simple syntax, fast compilation, automatic memory management, and more [16].

NewBlock function is responsible for adding a new block to the blockchain. Within this function, an instance of the Block struct is created, the NewProof function is called to determine the target, the Run method of the instance is used to perform hashing and validate the new block, and the instance of the block is returned. The NewBlock function can be seen in Figure 2.



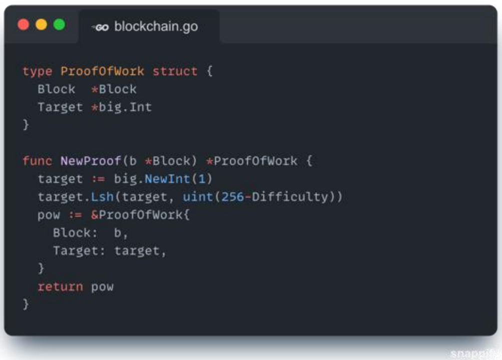
```go
type Block struct {
  Hash     string `json:"hash"`
  Data     string `json:"data"`
  PrevHash string `json:"prev_hash"`
  Nounce   int    `json:"nounce"`
}

func NewBlock(data string, prevHash string) *Block {
  block := &Block{
    Hash:     "",
    Data:     data,
    PrevHash: prevHash,
    Nounce:   0,
  }
  pow := NewProof(block)
  nounce, hash := pow.Run()
  block.Nounce = nounce
  block.Hash = hash

  return block
}
```

Figure 2. NewBlock Function

NewProof is a constructor method to create a consensus instance, in this case is the main logic of the ProofOfWork consensus algorithm. The purpose is to determine the target, which is the value that entities must achieve to validate a block, typically in binary units [17], [18]. In this function, an instance of the target is created using the big.Int data type, the Lsh method is employed to shift bits from the specified value, an instance of pow is created from the ProofOfWork struct, and the pow instance is returned. The NewProof Function can be seen in Figure 3.

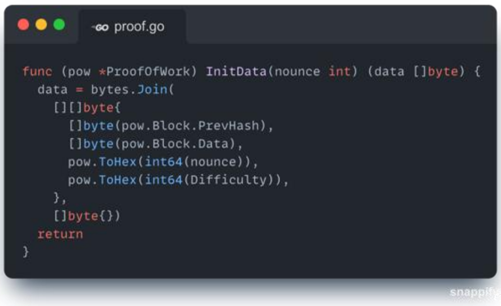


```go
type ProofOfWork struct {
  Block  *Block
  Target *big.Int
}

func NewProof(b *Block) *ProofOfWork {
  target := big.NewInt(1)
  target.Lsh(target, uint(256-Difficulty))
  pow := &ProofOfWork{
    Block:  b,
    Target: target,
  }
  return pow
}
```

Figure 3. NewProof Function

initData method of the pow instance is aims to concatenate several byte slices, namely PrevHash and Data from the pow instance, along with nonce and Difficulty, into a single byte slice without any separator characters. Nonce is a value that entities can modify to achieve the desired hash value; generally, the nonce is less than the target. Meanwhile, Difficulty represents the level of difficulty that entities must overcome to add a new block to the blockchain. The greater the difficulty value, the higher the difficulty level, and vice versa [19]–[21]. The InitData method can be seen in Figure 4.

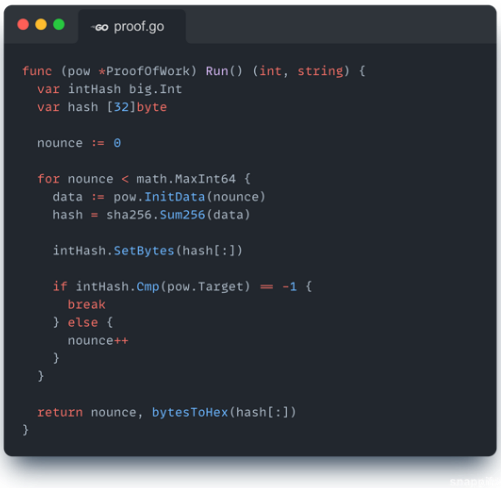


proof.go

```go
func (pow *ProofOfWork) InitData(nounce int) (data []byte) {
  data = bytes.Join(
    [][]byte{
      []byte(pow.Block.PrevHash),
      []byte(pow.Block.Data),
      pow.ToHex(int64(nounce)),
      pow.ToHex(int64(Difficulty)),
    },
    []byte{})
  return
}
```

Figure 4. InitData Method

Run method of the pow instance serves to hash the byte slice data that has been combined through the initData method of the pow instance and validate whether the hash of the new block is less than the target. If true, it is considered valid; otherwise, if false, it means it is not valid. In such cases, the nonce is incremented, and the process repeats with hashing and validation until a valid node is found. The Run method can be seen in Figure 5.



proof.go

```go
func (pow *ProofOfWork) Run() (int, string) {
  var intHash big.Int
  var hash [32]byte

  nounce := 0

  for nounce < math.MaxInt64 {
    data := pow.InitData(nounce)
    hash = sha256.Sum256(data)

    intHash.SetBytes(hash[:])

    if intHash.Cmp(pow.Target) == -1 {
      break
    } else {
      nounce++
    }
  }

  return nounce, bytesToHex(hash[:])
}
```

Figure 5. Run Method

3.6. System Testing

This stage is crucial for testing the created system to determine if it operates effectively and to identify and reduce potential errors. System testing employs the black box method, a software testing approach that focuses on the functionality of the system [22]. The challenge faced during system testing using black box testing is the limitation of computational resources for testing Proof of Work (PoW) consensus with high difficulty because if the difficulty increases, the computational energy required for the computation also increases. The System Testing Results can be seen in Table 1.

Table 1. System Testing Results

Test Case	Description	Result
Hash Algoritm	Hash a slice of bytes with the SHA256 Algorithm.	Success

Validate New Block	Validation of the block hash is less than the target. If the validation is successful, it will be added to the blockchain. However, if the validation fails, the nonce will be incremented, followed by hashing and validation again.	Success
Add New Block	If the block validation is successful, the block will be added to the blockchain and disseminated to other entities in the blockchain network.	Success
Edit Block	If there is a change in data within the blockchain, the validation in the consensus will become invalid, and the blockchain of the entity will self-correct by copying the blockchain from other entities.	Success
Delete Block	If there is a deletion in data within the blockchain, the validation in the consensus will become invalid, and the blockchain of the entity will self-correct by copying the blockchain from other entities.	Success

4. Conclusion

Based on the results of the system testing, this research successfully implemented the proof of work consensus in the blockchain using the go programming language. In this system, the consensus functions as a security measure and ensures data integrity by operating on every entity in the blockchain network to validate each transaction. This prevents manipulation and forgery of data and assists each entity in having accurate data through distribution without intermediaries or third parties, thus maintaining a decentralized system. System testing conducted through black box testing confirms the functionality and reliability of the implemented proof of work consensus.

This research contributes to a deeper understanding of the functions and benefits of the proof of work consensus in the blockchain. By delving into the nuanced dynamics of this consensus mechanism, the study offers perspectives for optimizing blockchain protocols and fostering trust among network participants These findings underscore the continued relevance of proof of work in the evolving landscape of blockchain technology and its potential implications for future advancements in distributed ledger systems and emphasize the role of this consensus mechanism in enhancing security and ensuring data integrity in the blockchain network.

For future research, it is recommended to explore alternative consensus mechanisms and their implications for blockchain systems. Additionally, investigating scalability issues and energy efficiency related to proof of work could provide valuable insights for the development and improvement of blockchain technology. A highly suggested area for potential exploration is to investigate the development and implications of alternative consensus mechanisms, such as Proof of Stake (PoS), Delegated Proof of Stake (DPoS), or other new protocols. This exploration can offer valuable insights into enhancing the sustainability and efficiency of blockchain systems. Moreover, researching scalability challenges and proposing innovative solutions to address them will be crucial to accommodate the growing demands of blockchain networks. Overall, this research lays the foundation for further exploration and refinement of consensus mechanisms in the dynamic field of blockchain.

References

[1] S. S. Sarmah, “Understanding Blockchain Technology,” Computer Science and Engineering, vol. 8, no. 2, pp. 23–29, 2018, doi: 10.5923/j.computer.20180802.02.

[2] H. Sheth, J. D.-A. J. F. C. I. Technology, and undefined 2019, “Overview of blockchain technology,” asianssr.org, vol. V Issue I, Accessed: Nov. 21, 2023. [Online]. Available: http://asianssr.org/index.php/ajct/article/view/728

[3] A. Haleem, M. Javaid, R. P. Singh, R. Suman, and S. Rab, “Blockchain technology applications in healthcare: An overview,” International Journal of Intelligent Networks, vol. 2, pp. 130–139, Jan. 2021, doi: 10.1016/J.IJIN.2021.09.005.

[4] H. Guo and X. Yu, “A survey on blockchain technology and its security,” Blockchain: Research and Applications, vol. 3, no. 2, p. 100067, Jun. 2022, doi: 10.1016/J.BCRA.2022.100067.

[5] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, “Blockchain Technology in Healthcare: A Systematic Review,” Healthcare 2019, Vol. 7, Page 56, vol. 7, no. 2, p. 56, Apr. 2019, doi: 10.3390/HEALTHCARE7020056.

[6] H. Xiong, T. Dalhaus, P. Wang, and J. Huang, “Blockchain Technology for Agriculture: Applications and Rationale,” Frontiers in Blockchain, vol. 3, p. 462928, Feb. 2020, doi: 10.3389/FBLOC.2020.00007/BIBTEX.

[7] J. Michael, A. Cohn, J. B.-T. Journal, and undefined 2018, “Blockchain technology,” steptoe.com, 2018, Accessed: Nov. 16, 2023. [Online]. Available: https://www.steptoe.com/a/web/171269/3ZEKzc/lit-febmar18-feature-blockchain.pdf

[8] D. Yaga, P. Mell, N. Roby, K. S. preprint arXiv:1906.11078, and undefined 2019, “Blockchain technology overview,” arxiv.org, Accessed: Nov. 23, 2023. [Online]. Available: https://arxiv.org/abs/1906.11078

[9] E. Tijan, S. Aksentijević, K. Ivanić, and M. Jardas, “Blockchain Technology Implementation in Logistics,” Sustainability 2019, Vol. 11, Page 1185, vol. 11, no. 4, p. 1185, Feb. 2019, doi: 10.3390/SU11041185.

[10] D. P. Oyinloye, J. Sen Teh, N. Jamil, and M. Alawida, “Blockchain Consensus: An Overview of Alternative Protocols,” Symmetry 2021, Vol. 13, Page 1363, vol. 13, no. 8, p. 1363, Jul. 2021, doi: 10.3390/SYM13081363.

[11] M. Casula, N. Rangarajan, and P. Shields, “The potential of working hypotheses for deductive exploratory research,” Qual Quant, vol. 55, no. 5, pp. 1703–1725, Oct. 2021, doi: 10.1007/S11135-020-01072-9/TABLES/4.

[12] E. Fridayanthie, … H. H.-J. K., and undefined 2021, “Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan (Persis Gawan) Berbasis Web,” simlitabmas.umkendari.ac.id, vol. 23, no. 2, 2021, doi: 10.31294/p.v23i2.10998.

[13] B. Sriman, S. Ganesh Kumar, and P. Shamili, “Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake,” Advances in Intelligent Systems and Computing, vol. 1172, pp. 395–406, 2021, doi: 10.1007/978-981-15-5566-4_34.

[14] N. Sapra, I. Shaikh, and A. Dash, “Impact of Proof of Work (PoW)-Based Blockchain Applications on the Environment: A Systematic Review and Research Agenda,” Journal of Risk and Financial Management 2023, Vol. 16, Page 218, vol. 16, no. 4, p. 218, Mar. 2023, doi: 10.3390/JRFM16040218.

[15] S. K. Zakarneh, Z. Qaroush, and A. Dawabsheh, “Cryptocurrencies Advantages and Disadvantages: A Review,” International Journal of Applied Sciences and Smart Technologies, vol. 4, no. 1, pp. 1–20, Jun. 2022, doi: 10.24071/IJASST.V4I1.4610.

[16] N. Kadek et al., “Implementation of Golang and ReactJS in the COVID-19 Vaccination Reservation System,” ADI Journal on Recent Innovation, vol. 5, no. 1, pp. 1–12, Feb. 2023, doi: 10.34306/AJRI.V5I1.877.

[17] S. Joshi et al., “Adoption of Blockchain Technology for Privacy and Security in the Context of Industry 4.0,” Wirel Commun Mob Comput, vol. 2022, 2022, doi: 10.1155/2022/4079781.

[18] S. Noda, K. Okumura, and Y. Hashimoto, “An Economic Analysis of Difficulty Adjustment

Algorithms in Proof-of-Work Blockchain Systems," SSRN Electronic Journal, Jun. 2019, doi: 10.2139/SSRN.3410460.

[19] H. Xiong, M. Chen, C. Wu, Y. Zhao, and W. Yi, "Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms," Future Internet 2022, Vol. 14, Page 47, vol. 14, no. 2, p. 47, Jan. 2022, doi: 10.3390/FI14020047.

[20] H. Zhu et al., "Blockchain Technology, Its Applications and Open Research Challenges," J Phys Conf Ser, vol. 1950, no. 1, p. 012030, Aug. 2021, doi: 10.1088/1742-6596/1950/1/012030.

[21] Q. Liu, Y. Xu, B. Cao, L. Zhang, and M. Peng, "Unintentional forking analysis in wireless blockchain networks," Digital Communications and Networks, vol. 7, no. 3, pp. 335–341, Aug. 2021, doi: 10.1016/J.DCAN.2020.12.005.

[22] N. Rahadi, C. V.-J. Infotekmesin, and undefined 2020, "Pengujian Software Aplikasi Perawatan Barang Miliki Negara Menggunakan Metode Black Box Testing Equivalence Partitions," pdfs.semanticscholar.org, vol. 11, no. 01, 2020, doi: 10.35970/infotekmesin.v11i1.124.