mathematics

MDPI

Article

Queuing Theory of Improved Practical Byzantine Fault Tolerant Consensus

Fan-Qi Ma [1] and Rui-Na Fan [2,*]

[1] School of Economics and Management Sciences, Yanshan University, Qinhuangdao 066004, China; mafanqi@stumail.ysu.edu.cn
[2] School of Management, Fudan University, Shanghai 200433, China
* Correspondence: fanruina@fudan.edu.cn; Tel.: +86-135-0444-4345

Abstract: In recent years, the use of consensus mechanism to maintain the security of blockchain system has become a considerable concern of the community. Delegated proof of stake (DPoS) and practical Byzantine fault tolerant (PBFT) consensus mechanisms are key technologies in maintaining the security of blockchain system. First, this study proposes a consensus mechanism combining DPoS and PBFT, which can rapidly deal with malicious witness nodes and shorten the time of block verification. Second, the M/PH/1 queuing model is used to analyze the performance of the proposed consensus mechanism, and the performance of the improved practical Byzantine fault tolerant consensus mechanism is evaluated from steady-state conditions and key performance measure of the system. Third, the current study uses the theoretical method of open (Jackson) queuing network, combined with the blockchain consensus process, and provides theoretical analysis with special cases. Lastly, this research utilizes numerical examples to verify the computability of the theoretical results. The analytic method is expected to open a series of potentially promising research in queueing theory of blockchain systems.

Keywords: blockchain; practical Byzantine fault tolerant; delegated proof of stake; phase-type distribution; queueing theory

Citation: Ma, F.-Q.; Fan, R.-N. Queuing Theory of Improved Practical Byzantine Fault Tolerant Consensus. *Mathematics* **2022**, *10*, 182. https://doi.org/10.3390/math10020182

Academic Editor: Daniel-Ioan Curiac

Received: 13 December 2021
Accepted: 5 January 2022
Published: 7 January 2022

1. Introduction

Blockchain is a new decentralized distributed system that uses cryptography, consensus mechanism, peer-to-peer communication, and other technologies to ensure the consistency and effectiveness of data across a network [1]. The use of consensus mechanism to make all nodes in the system reach consensus has consistently been the focus of blockchain technology research [2]. Given that proof of work (PoW), delegated proof of stake (DPoS), and practical Byzantine fault tolerant (PBFT) consensus mechanisms have been proposed, the problems of blockchain system in security, availability, and system performance have been relatively solved [3,4].

In recent years, researchers have made numerous contributions to the PBFT consensus mechanism. Initially, the Byzantine system needed exponential algorithms to solve it [5]. Subsequently, Castro and Liskov [6] proposed a polynomial Byzantine protocol, which substantially reduces the overhead of Byzantine protocol. Wood [7] proposed the Srcooge protocol, which reduces the response delay of the system and solves the problem that the performance of the system dropped significantly when there are Byzantine servers in the system. In order to meet performance requirements of commercial applications, Zhang [8] proposed a concurrent Byzantine fault tolerant algorithm model based on actor, which improves the transaction processing speed of the blockchain system. Abraham [9] proposed the Solida protocol, which is based on the reconfigurable Byzantine consensus decentralized blockchain, and improved Bitcoin within the confirmation time. Sukhwani [10] used random reward network to model the consensus process of PBFT and used data model to verify and conduct sensitivity analysis on various system parameters. Schwartz [11] proposed the ripple protocol consensus algorithm (RPCA) consensus mechanism, which

was combined with the Byzantine general problem. This consensus mechanism eliminated the limitation of reaching consensus through mining. The Hyperledger Project launched by the Linux Foundation [12] adopts the PBFT [13] algorithm to reach the consensus of the entire network. The Hyperledger Project is still in the development stage, and system reliability has not been tested on a large scale. Buchman [14] proposed a Tendermint consensus mechanism, which simplifies the design of the BFT consensus mechanism by relying on peer-to-peer gossip protocol between nodes.

However, when the number of nodes participating in PBFT consensus mechanism increases, the requirements for network bandwidth will increase at the polynomial level. If all nodes in the blockchain system join the PBFT consensus process, then meeting the requirements of network bandwidth and dynamics will be difficult. The DPoS consensus mechanism [15–18] can be used to elect nodes in the blockchain system initially, and the PBFT consensus mechanism can be carried out thereafter on alternative witness nodes. Hence, the number of nodes can be optimized, and the nodes election process can be transparent and tamper-resistance.

Malicious witness nodes may exist in the witness nodes elected by the DPoS consensus mechanism, and malicious witness nodes cannot produce blocks normally [19,20]. Moreover, the PBFT consensus mechanism can handle malicious witness nodes and complete the validation of blocks in a considerably short amount time. In summary, PBFT consensus mechanism can make up the shortcomings of DPoS consensus mechanism.

We should evaluate the performance of the consensus mechanism combining DPoS and PBFT. Markov process theory can be used as an effective mathematical tool to evaluate the performance of a blockchain system. Carlsten [21] used Markov process theory to analyze the influence of selfish mining on transaction costs in Bitcoin networks. Gobel [22] analyzed the mining competition between selfish mining pool and honest community by using a two-dimensional Markov process and extended the Markov model of selfish mining. Kiffer and Rajaraman [23] provided a simple Markov process framework, which is used to analyze the consistency attributes of blockchain protocols. Huang [24] established a Markov process with an absorption state and used it to analyze the performance measurement of the Raft consistency algorithm. Saulo [25] proposed a simple queueing model to capture the relationship between different quantities that jointly impact delays in a blockchain system. Moreover, the proposed queueing theory model accounts for factors such as the activity time of blocks and mean time between transactions.

The contributions of this study are threefold. The first contribution is to propose a consensus mechanism combining DPoS and PBFT, which can rapidly deal with malicious witness nodes and shorten the time of block verification. The second contribution is to develop a markedly general framework of Markov processes in the study of the consensus mechanism and to establish the consensus protocol of the consensus regulation. The third contribution is to analyze the performance of the consensus mechanism combining DPoS and PBFT by using the M/PH/1 queuing model and to evaluate the performance of the improved practical Byzantine fault-tolerant consensus mechanism from the steady-state conditions and key performance measure of the system. We use the theoretical method of open (Jackson) queuing network, combined with the blockchain consensus process, and provide some theoretical analyses with special cases.

The remainder of this paper is organized as follows. Section 2 establishes the consensus protocol of the consensus mechanism and describes the model. Section 3 uses the M/PH/1 queuing model to analyze the steady-state conditions of the system, and the steady-state distribution of the system is calculated thereafter. In addition, the important performance measures of the system are solved according to the steady-state distribution of the system. Section 4 verifies the feasibility of the theoretical results by numerical examples. Lastly, Section 5 provides the concluding remarks.

2. Consensus Protocol and Model Description

This section uses the DPoS consensus mechanism to elect witness nodes that can produce blocks. However, not all the witness nodes elected by the DPoS consensus mechanism are honest nodes, and malicious witness nodes cannot produce blocks normally. Therefore, the PBFT consensus mechanism is used to immediately verify the blocks generated by witness nodes, thereby ensuring rapid processing of malicious witness nodes and shortened time of block verification.

2.1. Consensus Protocol

Based on the background of blockchain, a consensus protocol combining DPoS consensus mechanism and PBFT consensus mechanism is defined. The consensus process is mainly divided into three parts. The first part is witness node election, and the second part is witness node conversion. What is more is that the third part is witness node block generation and block verification.

(1) Witness node election process

The nodes in the blockchain system are elected through the DPoS consensus mechanism, and nodes with superior performance in production blocks and high votes are selected as candidate nodes. The candidate nodes are divided into the witness node set and alternative witness node set. Witness nodes have the power to package blocks, whereas alternative witness nodes are responsible for verifying blocks produced by witness nodes and dealing with malicious witness nodes.

(2) Witness node transformation process

For the blockchain system, not all witness nodes elected by the DPoS consensus mechanism are honest nodes, and there will be node downtime and node malicious inevitably. When a malicious witness node appears, the node will not be able to produce new blocks normally. When the production block of the witness node fails, it will be transformed to the alternative witness node and its work will be changed from block production to block verification. Moreover, a node is selected from the alternative witness node set to transform into a witness node, and its work is transformed from block verification to block production.

(3) Witness node block generation and block verification

When the witness node produces a block, it is immediately sent to the alternative witness node for validation. In order to immediately complete the block verification work, the PBFT consensus mechanism is run in alternative witness nodes. Therefore, the major node is elected from the alternative witness node set. Each node can be the major node in turn, and then a round of election is conducted. After a major node is elected, other nodes in the alternative witness node set are slave nodes. The PBFT consensus mechanism is mainly divided into three stages: prepare, commit, and reply stages.

(a) Prepare stage: The witness node sends the produced block to the alternative witness node set. After the major node in the alternative witness node receives the block information, it will broadcast to the slave nodes. After the message is verified to be correct, the slave node will broadcast the prepare message to all the slave nodes, and the prepare stage will pass.

(b) Commit stage: The node sends a confirmation message to all other nodes, as well as votes on the received block message and broadcasts the final voting result. The major node and slave nodes are summarized according to the received confirmation results. The commit stage passes when over two-thirds of the nodes in the blockchain system confirm the transaction content of the block. In general, a node will leave the system immediately after voting for a block. However, the properties of some Byzantine nodes have changed (e.g., Byzantine nodes are changed to non-Byzantine nodes, which means that the nodes that fail and forge information become nodes that fail but do not forge information). This kind of situation exists in PBFT consensus mechanism. Thus, it is necessary to vote for a second time.

(c) Reply stage: When the commit stage is completed and the received verification information is entered into the reply stage, the node can be determined to have reached a round of consensus and the block can be added to the blockchain.

2.2. Model Description

We provide the model description and related system parameters according to the consensus protocol.

(1) Block arrival process: The actual situation of the blockchain system indicates that when the witness node in the DPoS consensus mechanism produces a block, it will be immediately sent to the alternative witness nodes for verification. We regard this process as the block arrival process in the queuing system and assume that it follows the Poisson process with parameter $\lambda(\lambda > 0)$.

(2) Block consensus process: In the PBFT consensus at the alternative witness node, when some nodes in the blockchain system are Byzantine nodes, the slave nodes in the system vote on the received block messages and broadcast the final voting results. Slave node confirms according to the received voting result and broadcasts the confirmation result again. The assumption is that the voting time of this process follows the exponential distribution with parameter $\mu_1(\mu_1 > 0)$. Meanwhile, the verified block leaves the system with probability p and joins the blockchain. When the Byzantine node in the blockchain system becomes a non-Byzantine node, slave nodes in the system will vote for the received block message for the second time. The block enters the second voting process with a probability of $q = 1 - p$, assuming that the voting time of this process follows the exponential distribution with parameter $\mu_2(\mu_2 > 0)$. After the second voting, the verified blocks will leave the system and join the blockchain immediately. In the block consensus process, the voting time of block obeys the exponential distribution of two different parameters, and its state space belongs to a two-dimensional Markov process. Block voting time can be expressed as a two-phase Phase Type (PH) distribution and its irreducible matrix is expressed as (α, T), where α is a substochastic vector of order m, and T is a transfer rate matrix of order m.

(3) Block consensus discipline: Blocks generated by the witness nodes follow the First Come First Service (FCFS) queuing rule, and the system only makes consensus for one block at a time. If there are blocks waiting for consensus, then the newly arrived block will enter the trading pool for queuing. If there is no block waiting for consensus, then the consensus will be presented by the major node as soon as the block arrives.

(4) Independence: We assume that all previously defined random variables are independent of one another.

On the basis of the preceding model description, the blockchain queuing system of block consensus process is shown in Figure 1.

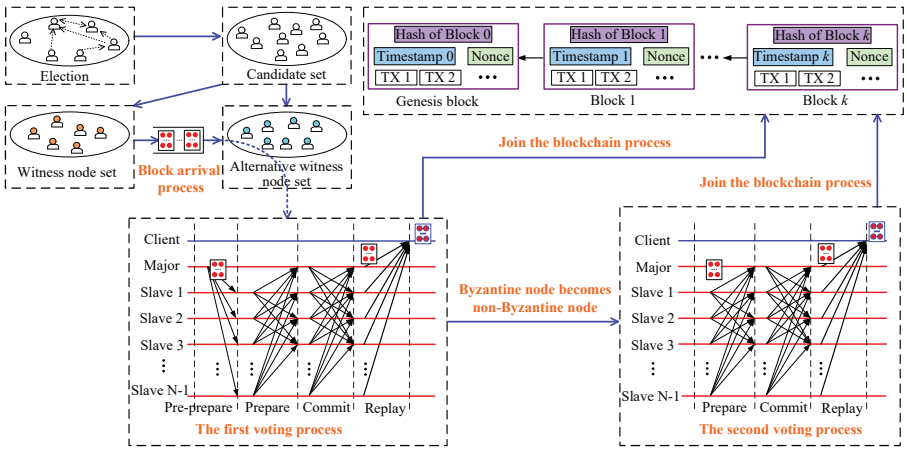

Figure 1. Blockchain queuing system based on the block consensus process.

3. Steady-State Analysis

3.1. The M/PH/1 Type Queueing System

This section establishes a continuous time M/PH/1 queueing system for the blockchain system. First, the steady-state conditions of the system are analyzed. Second, the steady-state distribution of the system is calculated. Lastly, the important performance measure is solved according to the steady-state distribution of the system.

In the blockchain system, let $N(t)$ represent the number of blocks in the system at time t and $J(t)$ is the phase of the environment when a block receives the consensus at time t. Meanwhile, $\{N(t), J(t) : t \geq 0\}$ is a continuous time Markov process, the state transition relations of which are shown in Figure 2.

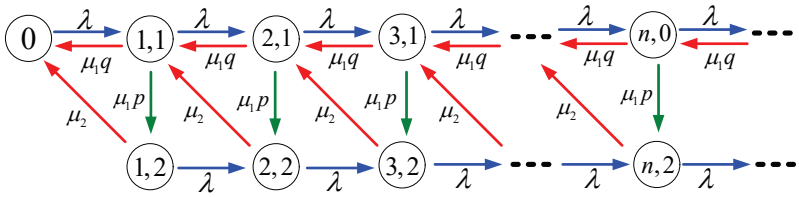

Figure 2. State transition relation of the system.

Figure 2 shows that the state space of the Markov process $\{N(t), J(t) : t \geq 0\}$ is given as follows.

$$\Omega = \{0\} \cup \{(n, j), n \geq 1, 1 \leq j \leq 2\}$$

Level $l(0)$ corresponds to an empty system and level $l(n)$ corresponds to the state $\{(n, j), n \geq 0,\ j = 1, 2\}$, indicating that there are n blocks in this system. Among these blocks, $n - 1$ blocks are waiting in line and one block is accepting consensus and in the jth phase.

According to the preceding analysis, the infinitesimal generator of the Markov process $\{N(t), J(t) : t \geq 0\}$ is as follows:

$$Q = \begin{pmatrix} -\lambda & \lambda\alpha & 0 & 0 & \cdots \\ t & T - \lambda I & \lambda I & 0 & \cdots \\ 0 & t \cdot \alpha & T - \lambda I & \lambda I & \cdots \\ 0 & 0 & t \cdot \alpha & T - \lambda I & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where the following is the case.

$$\alpha = (1, 0), t = \begin{pmatrix} \mu_1 q \\ \mu_2 \end{pmatrix}, T = \begin{pmatrix} -\mu_1 & \mu_1 p \\ 0 & -\mu_2 \end{pmatrix}.$$

The following theorem provides a necessary and sufficient condition under which the Markov process $\{N(t), J(t) : t \geq 0\}$ is stable. Based on this, we can obtain the system stability of the block consensus.

Theorem 1. *The Markov process $\{N(t), J(t) : t \geq 0\}$ is positive recurrent if and only if the following is the case.*

$$\rho = \frac{(\mu_1 p + \mu_2)\lambda}{\mu_1 \mu_2} < 1. \tag{1}$$

Proof of Theorem 1. According to the mean drift method given Neuts [26], we write the following.

$$A = t\alpha + (T - \lambda I) + \lambda I = \begin{pmatrix} \mu_1 q - \mu_1 & \mu_1 p \\ \mu_2 & -\mu_2 \end{pmatrix}$$

It is clear that matrix A is irreducible, aperiodic, and has positive recurrence due to the fact that its state space is finite and $Ae = 0$, where e is a column vector with all components ones.

In this case, let $\theta = (\theta_0, \theta_1)$ be the stationary probability vector of the matrix A. Therefore, the stationary probability vector satisfies the system of linear equations: $\theta A = 0$ and $\theta e = 1$. Based on this, we obtain the following.

$$(\mu_1 q - \mu_1)\theta_0 + \mu_2 \theta_1 = 0,$$

$$\mu_1 p \theta_0 - \mu_2 \theta_1 = 0,$$

$$\theta_0 + \theta_1 = 0.$$

Hence, we obtain the following.

$$\theta = (\frac{\mu_2}{\mu_1 p + \mu_2}, \frac{\mu_1 p}{\mu_1 p + \mu_2})$$

By using the mean drift method, it is easy to see that the Markov process $\{N(t), J(t) : t \geq 0\}$ is positive recurrent if and only if the following is the case.

$$\theta \lambda I e < \theta t \alpha e.$$

It is easy to check the following:

$$\theta \lambda I e = \lambda,$$

where I is an identity matrix. At the same time, we have the following.

$$\theta t \alpha e = \frac{\mu_1 \mu_2}{\mu_1 p + \mu_2}.$$

Thus, we obtain the following.

$$\frac{\mu_1 \mu_2}{\mu_1 p + \mu_2} > \lambda,$$

$$\frac{(\mu_1 p + \mu_2)\lambda}{\mu_1 \mu_2} < 1.$$

Let the following be the case.

$$\rho = \frac{(\mu_1 p + \mu_2)\lambda}{\mu_1 \mu_2}.$$

This completes the proof. □

For the Markov process $\{N(t), J(t) : t \geq 0\}$, to compute its stationary probability vector, we need to first obtain the rate matrix R.

Theorem 2. *For the following matrix equation:*

$$R^2 t\alpha + R(T - \lambda I) + \lambda I = 0, \tag{2}$$

its solution R satisfies $Rt = \lambda e$ and has a minimum nonnegative solution.

$$R = \lambda(\lambda I - T - \lambda e \alpha)^{-1}$$

Proof of Theorem 2. Multiply both ends of Equation (2) by e, where $Te = -t$:

$$R^2t - Rt - \lambda Re + \lambda e = 0,$$

$$(I - R)(\lambda e - Rt) = 0,$$

$I - R$ is reversible; therefore, $Rt = \lambda e$, and when it is substituted into the Equation (2), the following is obtained.

$$R(\lambda I - T - \lambda e\alpha) = \lambda I,$$

We can obtain the following.

$$R = \lambda(\lambda I - T - \lambda e\alpha)^{-1}.$$

This completes the proof. □

Theorem 3. *When $\rho < 1$, the steady-state probability vector of Markov process $\{N(t), J(t) : t \geq 0\}$ satisfies the following.*

$$\begin{cases} \pi_0 = 1 - \rho, \\ \pi_k = (1 - \rho)\alpha R^k, k \geq 1. \end{cases} \tag{3}$$

Proof of Theorem 3. According to the matrix geometric solutions, the following is the case:

$$\pi_k = \pi_1 R^{k-1}, k \geq 2,$$

and (π_0, π_k) satisfies the following.

$$\begin{cases} -\lambda\pi_0 + \pi_1 t = 0, \\ \lambda\pi_0\alpha + \pi_1(Rt\alpha - \lambda I + T) = 0. \end{cases} \tag{4}$$

Considering $Rt = \lambda e$, the solution of Equation (4) is as follows:

$$\pi_1 = \pi_0\alpha R.$$

According to the normalization condition,

$$\pi_0 + \pi_0\alpha R(I - R)^{-1}e = 1. \tag{5}$$

If we substitute R into Equation (5) and obtain the following:

$$\pi_0 + \lambda\pi_0\alpha(T + \lambda e\alpha)^{-1}e = 1, \tag{6}$$

the inverse operation is as follows.

$$\begin{aligned} (T + \lambda e\alpha)^{-1} &= T^{-1}(I + \lambda e\alpha T^{-1})^{-1} \\ &= T^{-1}\sum_{j=0}^{\infty}(-I)^j\lambda^j(e\alpha T^{-1})^j \\ &= T^{-1}\left\{I - \lambda(1 - \rho)^{-1}e\alpha T^{-1}\right\}. \end{aligned} \tag{7}$$

By substituting Equation (7) into Equation (6), we can obtain the following:

$$\pi_0 - \lambda\pi_0\alpha T^{-1}e + \lambda^2\pi_0(1 - \rho)^{-1}\alpha T^{-1}e\alpha T^{-1}e = \pi_0(1 - \rho)^{-1} = 1$$

thus, the following is obtained.

$$\pi_0 = 1 - \rho.$$

This completes the proof. □

When $\rho < 1$, the distribution of the number of blocks in the Markov process $\{N(t), J(t) : t \geq 0\}$ is as follows.

$$P\{Q = k\} = \pi_k e = \begin{cases} 1-\rho, & k = 0, \\ (1-\rho)\alpha R^k e, & k \geq 0. \end{cases} \tag{8}$$

When the system is in steady state, the average number of blocks $E[N]$ in the process of block consensus is as follows.

$$E[N] = \sum_{k=0}^{\infty} k\pi_k e = (1-\rho)\sum_{k=0}^{\infty} kR^k e = (1-\rho)\alpha R(I-R)^{-2}e. \tag{9}$$

3.2. Special Case: Open (Jackson) Queuing Network

This section uses an open (Jackson) queuing network to analyze the blockchain consensus process. Thus, it can show the universal applicability of using queuing theory to solve the problem of blockchain consensus process.

When the blocks produced in the blockchain system arrive at the Byzantine nodes (node 1) with intensity $\lambda_1(\lambda_1 > 0)$, Byzantine nodes in the system vote on the received block messages and broadcast the final voting results, thereby confirming the voting results and broadcasting the confirmation results. The assumption is that the voting time of this process follows the exponential distribution with parameter $\mu_1(\mu_1 > 0)$. Furthermore, the verified blocks leave the system with probability p and join the blockchain. When the Byzantine node in the blockchain system becomes a non-Byzantine node, the slave nodes in the system will vote for the received block message for the second time. At this time, the produced block arrives the non-Byzantine node (node 2) with intensity λ_2, and the block enters the second voting process with a probability of $q(q = 1-p)$. The assumption is that the voting time of this process follows the exponential distribution with parameter $\mu_2(\mu_2 > 0)$. After the second voting, the verified blocks will leave the system and join the blockchain immediately.

The block consensue process can be viewed as an open network. We take the outside (input/output) as node 0 and assume that block arrive node 0 with intensity λ. The routing matrix is given as follows.

$$\Theta = \begin{pmatrix} 0 & 1 & 0 \\ p & 0 & q \\ 0 & 1 & 0 \end{pmatrix}.$$

Then, we obtain the following.

$$\begin{cases} \lambda = p\lambda_1, \\ \lambda_1 = \lambda + \lambda_2, \\ \lambda_2 = q\lambda_1. \end{cases}$$

Thus, the following is the case.

$$\lambda_1 = \lambda/p,\ \lambda_2 = \lambda q/p.$$

We set ρ_1, ρ_2 as the load coefficients of node 1 and node 2, respectively. Then, $\rho_1 = \lambda_1/\mu_1$, $\rho_2 = \lambda_2/\mu_2$. We denote the number of block in node 1 and node 2 by k_1 and k_2, respectively. By Jackson's Theorem, the steady-state probability of the network can be expressed as follows:

$$\pi(k_1, k_2) = \pi_1(k_1)\pi_2(k_2)$$

in which the following is the case.

$$\pi_i(k_i) = (1-\rho_i)\rho_i^{k_i},\ i = 1, 2, k_i = 0, 1, 2, \ldots$$

The average number of block in node i is given by the following.

$$\overline{k_i} = \frac{\rho_i}{1-\rho_i},\ i = 1, 2. \tag{10}$$

Thus, the average number of block in the network is as follows.

$$E[M] = \overline{k_1} + \overline{k_2} = \frac{\rho_1}{1-\rho_1} + \frac{\rho_2}{1-\rho_2}. \tag{11}$$

4. Numerical Analysis

This section uses some numerical examples to verify computability of our theoretical results and shows how the performance measure depends on the main parameters of this blockchain system.

(a) Analyze the influence of λ on $E[N]$

In this blockchain system, we take the following basic parameters.

$$\alpha = (1, 0), \mu_1 = 4, \mu_2 = 4.$$

Figure 3 shows how the average number of blocks $E[N]$ depends on $\lambda \in (0.5, 2)$ when $p = 0.1, 0.2$ and 0.3. Note that when the p is constant, $E[N]$ increases and λ increases. When λ is constant, $E[N]$ increases and p increases. This numerical result can be intuitively understood as follows. When λ increases, the number of blocks increases; hence, $E[N]$ increases. When p increases, the number of blocks that can reach consensus increases; thus, $E[N]$ increases.

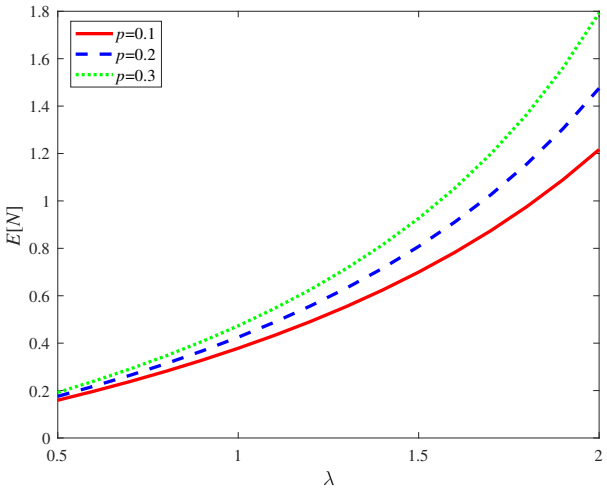

Figure 3. Relation between $E[N]$ and λ.

(b) Analyze the influence of μ_1 on $E[N]$

In this blockchain system, we take the following basic parameters.

$$\alpha = (1, 0), \lambda = 2, \mu_2 = 4.$$

Figure 4 shows how the average number of blocks $E[N]$ depends on $\mu_1 \in (3, 5)$ when $p = 0.1, 0.2$ and 0.3. Note that when the p is constant, $E[N]$ decreases and μ_1 increases. When μ_1 is constant, $E[N]$ increases and p increases. Intuitively, when μ_1 increases, the number of blocks that verify legitimacy increases; hence, $E[N]$ decreases. When p increases, the number of blocks that can reach consensus increases; thus, $E[N]$ increases.

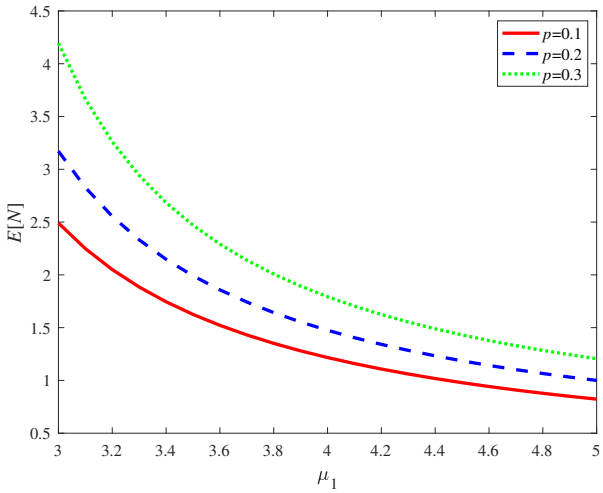

Figure 4. Relation between $E[N]$ and μ_1.

(c) Analyze the influence of μ_2 on $E[N]$
In this blockchain system, we take the following basic parameters.

$$\alpha = (1,0), \lambda = 2, \mu_1 = 4.$$

Figure 5 shows how the average number of blocks $E[N]$ depends on $\mu_2 \in (2,4)$ when $p = 0.1, 0.2$ and 0.3. Note that when p is constant, $E[N]$ decreases, and μ_2 increases. When μ_2 is constant, $E[N]$ increases and p increases. Intuitively, when μ_2 increases, the number of blocks verified to be legal in the second round of voting increases; thus, $E[N]$ decreases. When p increases, the number of blocks that can reach consensus increases; hence, $E[N]$ increases.

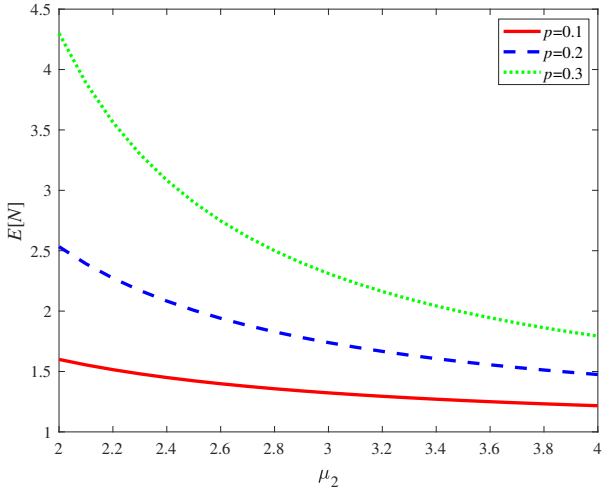

Figure 5. Relation between $E[N]$ and μ_2.

A comparison between Figures 4 and 5 shows that μ_2 has a greater impact on $E[N]$ than μ_1. This result indicates that when the property of Byzantine nodes in the system changes, the consensus process of the blockchain system will be substantially affected.

5. Conclusions

This research aims to develop a queuing theory of blockchain systems, simplify the consensus steps, and improve the efficiency of consensus by establishing the queuing model, thereby optimizing the performance of a blockchain system. Accordingly, we propose a consensus mechanism combining DPoS and PBFT, which can rapidly deal with malicious

witness nodes and shorten the time of block verification. By using the M/PH/1 Markov process model, we obtain a system stable condition and also express a key performance measure, which is the average number of blocks in the block consensus process. Moreover, we use the theoretical method of open (Jackson) queuing network, combined with the blockchain consensus process and conduct theoretical analyses with special cases. Lastly, we use numerical examples to verify computability of our theoretical results. The numerical examples indicate that adjusting system parameters within a certain range can substantially improve the system's consensus efficiency.

Author Contributions: Conceptualization, F.-Q.M.; formal analysis, F.-Q.M.; investigation, R.-N.F.; methodology, R.-N.F. and F.-Q.M.; resources, F.-Q.M.; supervision, R.-N.F.; writing—original draft, F.-Q.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: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

PBFT	Practical Byzantine Fault Tolerant;
DPoS	Delegated Proof of Stake;
FCFS	First-Come-First-Service;
PoW	Proof of Work.

References

1. Bonneau, J.; Miller, A.; Clark, J.; Narayanan, A.; Kroll, J.; Felten, E. SoK: Research perspectives and challenges for bitcoin and cryptocurrencies. In Proceedings of the 2015 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 17–21 May 2015; pp. 104–121.
2. Cho, H. ASIC-Resistance of Multi-Hash Proof-of-Work Mechanisms for Blockchain Consensus Protocols. *IEEE Access* **2018**, *6*, 66210–66222. [CrossRef]
3. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data, Honolulu, HI, USA, 25–30 June 2017; pp. 557–564.
4. 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 Trans. Ind. Inform.* **2019**, *15*, 3680–3689. [CrossRef]
5. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [CrossRef]
6. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst. (TOCS)* **2002**, *20*, 398–461. [CrossRef]
7. Wood, T.; Singh, R.; Venkataramani, A.; Shenoy, P.; Cecchet, E. ZZ and the Art of Practical BFT Execution. In Proceedings of the EuroSys 2011 Conference, Salzburg, Austria, 10–13 April 2011; pp. 123–137.
8. Zhang, C.; Wang, R.; Tsai, W.T.; He, J.; Li, Q. Actor-based Model for Concurrent Byzantine Fault-tolerant Algorithm. In Proceedings of the 2019 International Conference on Computer, Network, Communication and Information Systems, Qingdao, China, 27–29 September 2019; pp. 552–558.
9. Abraham, I.; Malkhi, D.; Nayak, K.; Ren, L.; Spiegelman, A. Solida: A Blockchain Protocol Based on Reconfigurable Byzantine Consensus. *arXiv* **2016**, arXiv:1612.02916.
10. Sukhwani, H.; Martinez, X.; Chang, X.; Trivedi, K.; Rindos, A. Performance Modeling of Pbft Consensus Process for Permissioned Blockchain Network (hyperledger Fabric). In Proceedings of the 2017 IEEE 36th Symposium on Reliable Distributed Systems, Hong Kong, China, 26–29 September 2017; pp. 253–255.
11. Schwartz, D.; Yungs, N.Y.; Britto, A. The Ripple Protocol Consensus Algorithm. Available online: https://ripple.com/files/ripple~{}consensus~{}whitepaper.pdf (accessed on 15 February 2018).
12. Platania, M.; Obenshain, D.; Tantillo, T.; Amir, Y.; Suri, N. On Choosing Server or Client Side Solutions for BFT. *ACM Comput. Surv.* **2016**, *48*, 1–30. [CrossRef]
13. Cachin, C. Architecture of the Hyperledger Blockchain Fabric 2016. Available online: https://www.zurich.ibm.com/dccl/papers/cachindccl.pdf (accessed on 15 July 2016).
14. Buchman, E.; Kwon, J.; Milosevic, Z. The Latest Gossip on Bft Consensus. *arXiv* **2018**, arXiv:1807.04938.

15. Larimer, D. DPOS Consensus Algorithm—The Missing Whitepaper. Available online: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper (accessed on 28 May 2020).
16. Snider, M.; Samani, K.; Jain, T. Delegated Proof of Stake: Features and Tradeoffs. Available online: https://multicoin.capital/wp-content/uploads/2018/03/DPoS-Features-and-Tradeoffs.pdf (accessed on 15 January 2021).
17. Qian, H.; Yan, B.; Han, Y.; Yu, J. An Improved Delegated Proof of Stake Consensus Algorithm. *Procedia Comput. Sci.* **2021**, *187*, 341–346.
18. Yang, F.; Zhou, W.; Wu, Q.; Long, R.; Xiong, N.; Zhou, M. Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism. *IEEE Access* **2019**, *7*, 118541–118555. [CrossRef]
19. Sun, Y.; Yan, B.; Yao, Y.; Yu, J. DT-DPoS: A Delegated Proof of Stake Consensus Algorithm with Dynamic Trust. *Procedia Comput. Sci.* **2021**, *187*, 371–376. [CrossRef]
20. Mu, Y.; Chen, W.; Liang, X.; Gao, Y. A Weak Centralized Consensus Mechanism with More Incentive Effects. *J. Phys. Conf. Ser.* **2019**, *1302*, 032037. [CrossRef]
21. Carlsten, M. The Impact of Transaction Fees on Bitcoin Mining Strategies. Master's Thesis, Princeton University, Princeton, NJ, USA, 2016. Available online: https://www.cs.princeton.edu/research/techreps/TR-983-16 (accessed on 16 May 2016).
22. Gobel, J.; Keeler, H.P.; Krzesinski, A.E.; Taylor, P.G. Bitcoin Blockchain dynamics: The Selfish-mine Strategy in the Presence of Propagation Delay. *Perform. Eval.* **2016**, *104*, 23–41. [CrossRef]
23. Kiffer, L.; Rajaraman, R.; Shelat, A. A Better Method to Analyze Blockchain Consistency. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 729–744.
24. Huang, D.; Ma, X.; Zhang, S. Performance Analysis of the Raft Consensus Algorithm for Private Blockchains. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 172–181. [CrossRef]
25. Saulo, R.; Eduardo, F.; Daniel, S.M. Learning blockchain delays: A queueing theory approach. *ACM Sigmetrics Perform. Eval. Rev.* **2019**, *46*, 122–125.
26. Neuts, M.F. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*; Johns Hopkins University Press: Baltimore, MD, USA, 1981.