Decentrally-Consented-Server-Based Blockchain System for Universal Types of Data

Miraz Uz Zaman
Computer Science Program
Louisiana Tech University
Ruston, Louisiana 71272
muz001@latech.edu

Manki Min
Computer Science Program
Louisiana Tech University
Ruston, Louisiana 71272
mankimin@latech.edu

***Abstract*—The rapid increase of data makes the third party cloud storage more and more popular. However, decentralization, immutability, and integrity characteristics of Blockchain systems are making Blockchain systems another potential platform for data storage. Using the Blockchain system for data storage will eliminate the dependency on any central authority thus increasing the data security, privacy, data retention probability, and eliminate the single point of failure. Most of the Blockchain consensus mechanism have been developed over the decade are about storing the transactional data only. In this paper, we present a novel idea of the Blockchain consensus system for universal types of data. Our proposed consensus mechanism is designed in such a way that, most of the devices or entities can participate in the mechanism by doing low computational validation work. To add an extra layer of security we introduce a two-step validation process in this consensus mechanism. The probabilistic fair rotation of the block creator in our consensus mechanism will significantly reduce the probability of centralization which is a common issue in most of operating Blockchain consensus mechanism based on one user/group winning strategy. Moreover, we analyze our consensus mechanism using game theory and queuing theory.**

***Index Terms*—Universal Data Storage, Blockchain Consensus, One-time Consented Arbitrator**

I. Introduction

The *Blockchain* technology makes the first public appearance by the hand of Bitcoin [1] back in 2009. Within a decade of its debut, Blockchain became the frontier decentralized technology with a total market cap of $215.64B [2]. Though a great extent of Blockchain applications are finance-related, the evolved feature of Blockchain can impact other numerous industries. Some real-world industrial use cases are healthcare, digital voting, transportation, business, decentralized governance, data sharing, etc. To serve the demand for the increasing applications of Blockchain technology, a wide variety of different consensus mechanisms are being proposed.

The rapid evolution of IoT devices and the resulting outbreak of data produced by the massively interconnected devices make the data to be outsourced in remote servers. These days cloud-based storage is becoming the standardized outsourcing method in storing, processing, and distributing data. Most of the cloud-based storage data is centrally controlled and backed by a handful of technology companies with the incredible ability of data storing. However, these centralized data storages are also susceptible to some downsides such as data breaching, server misbehavior, high unavailability of the storage server, storage costs, and absence of stored data validity.

978-1-7281-5628-6/20/$31.00

Decentralized Blockchain based storage is becoming a potential solution to the centralized data storage's challenges due to its inherent characteristics such as immutability and public verifiability. Since not all the participants in the Blockchain network are uniformly powered in storing and processing data, careful assignment of the responsibilities to the users based on their power will allow more mobile devices to be integrated into the network. On another note, the biggest issue in Blockchain is its capacity for handling the growing data in the block (scalability), network speed, and latency. With an optimized tradeoff between the inherent characteristics of Blockchain and its issues, a Blockchain based data storage platform could be an ideal candidate to replace the centralized cloud-based data storage.

In this paper, we propose a Blockchain consensus mechanism for data storage and data validation which is scalable to mobile devices. Through incorporating a two-step validation process with these devices we make sure that our proposed consensus mechanism could be more secure and significantly more economical than centralized cloud based storage systems. Our main contribution can be summed up as follows:

- We designed a framework of the proposed consensus mechanism for the Blockchain based distributed storage
- We performed a game theoretic analysis of the proposed consensus mechanism by predicting the behavior or strategies of consensus participants for different situations: the stability of these situations has been calculated using the Nash Equilibrium
- We performed a queuing theoretic analysis to study the expected congestion, i.e., expected waiting time of the participants and the expected number of participants: this will provide the participants to detect the unusual activities in the network

The rest of the paper is arranged as follows: literature on the mostly used consensus algorithm is briefly discussed in II. Our proposed scheme is described in III and analyzed in IV,

and the paper is concluded by V.

II. Related Work

Most of the Blockchain based file sharing consensus mechanisms are built based on the idea of BitTorrent [3], one of the most popular peer to peer file sharing protocols. Another open source, decentralized Blockchain based storage platform similar to BitTorrent called STORJ [4] has been widely studied. Alike the BitTorrent, STORJ split the file into multiple pieces through the process called file sharding. Then the sharded files are distributed through the STORJ network. The main difference between BitTorrent and STORJ is using the distributed hash table to locate all the sharded files so that only the owner of the file knows all the locations. The building block of the STORJ network is Kademlia [5], a distributed hash table. The owner of the file can select the different levels of redundancy to increase the file availability in the network.

Another Blockchain based decentralized storage platform is called *Sia* [6] and the idea is proposed in *HackMIT 2013*. This is a contract based platform where the storage provider and client have an agreement on the storage data and the price of storing the data. As a part of the agreement, the storage provider needs to prove that the data is still intact stored through *Proof of Storage* [7] consensus mechanism. Moreover, the agreement also specifies the duration of file storage, the frequency of providing proof, the reward for the valid proof, and the maximum number of proofs that can be missed. The contracts are *successfully terminated* when the duration of data storage is over and *unsuccessfully terminated* when the maximum number of proofs that can be missed is exceeded. The contracts are stored in the Blockchain network system so that they can be made publicly auditable, immutable, and decentralized.

Lately *IPFS* (Interplanetary File System) [8], another open source distributed and decentralized file sharing technology, has been invented by the Protocol Labs. This is being considered as a successor of modern internet architecture. The key idea in *IPFS* is the addressing format. Instead of the location addressing like in HTTP, it is using content addressing. That means the same files in the network can be referred by a unique address, i.e., the hash of the file. There are mainly two fields in an IPFS object: unstructured binary data less than 256 kB and a link of other IPFS objects. Again in the link structure, there are three data fields: the name of the link, the hash of the linked IPFS object, and the cumulative size of the linked IPFS object. If the original IPFS object is less than 256 kB, the link field will be kept empty. As of January 15, 2020, there is no clear pathway of tracking the versioned file in the IPFS system. To keep motivating IPFS users through rewarding a cryptocurrency is built on top of the IPFS technology called *FileCoin* [9].

As Bitcoin and other Blockchain based cryptocurrency become more and more popular, some self contrary issues are also emerging. Such an issue is the centralization, the main reason for this issue is that the computation power and storage capabilities are dominated only by a handful of organizations or systems. For example, the idea of aggregation of hash power or *pool mining* [10] in bitcoin triggered the centralization issue in the network. As of December 15, 2019, the hash rate distribution for bitcoin networks among the pools has been shown in [11]. If a group of pools work together and control the majority of the hash rate in the bitcoin network then the decentralization of the system could be easily destabilized through 51% attack and double spending. As a consequence of growing hash computation among the miners, the probability of solo miners getting profit is decreasing rapidly. Similarly, in a storage based decentralized Blockchain if a batch of server controls the majority of storage capabilities then, in the same way, the decentralization issue can pop up. In addition to the centralized pool mining, another issue of unfairness resulting from the imprecise reward scheme because of the inclusion of several costs, i.e., the pool operating cost and the pool transaction cost set by the pool owner. On the other hand, in our proposed consensus mechanism there is no need to form the pool and any user who participates in our validation process can only the selected validation will be rewarded with a fair and transparent reward scheme.

Moreover, most of the cryptocurrency mechanism is fully dependent on one validator (miner) throughout the generation of the block. So, the validators need to have high computational power and/or storage to compete with other validators. As moving forward due to ever-increasing competition, the consensus mechanisms will be more and more biased to the richer miners which will lead to the centralization. However, having a fixed validator for every block can launch different attacks such as DDoS attack [12] and Eclipse attack [13]. Our proposed consensus mechanism can minimize these attacks because of its probabilistically fair rotation of validators and multiple users' participation for the block generation.

III. Proposed Scheme

A. Entities

The entities in our proposed scheme consist of block-warehouses that store the entire chain of blocks and users who wants access to the chain of blocks. The roles involved in our proposed consensus mechanism are named as *Selectors*, *Validators* and *Invalidators*. Typically selectors have high storage capabilities and validators could be any mobile or personal devices that can connect to the Blockchain network. Invalidators in the network could be anyone from the set of validators of the network. The main task of the invalidators is to point out any errors or wrongdoings in the validation process. To increase the accountability of each selector we also use a counterpart of the selector in our consensus mechanism. We define the former one as *Positive Selector* and the latter one as *Negative Selector* to distinguish them.

B. Choosing Positive Selectors and Negative Selectors

In this section, we explain the new concept of choosing positive and negative selectors for our proposed scheme.

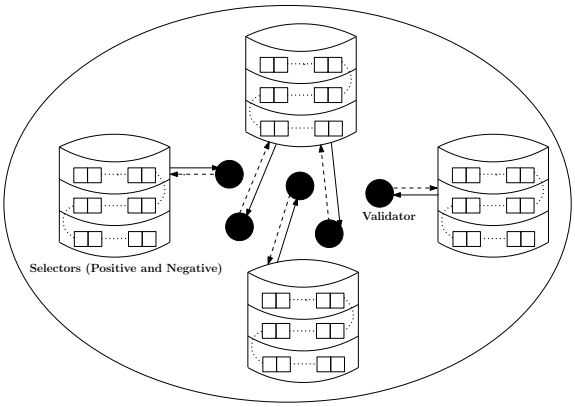


Fig. 1: Role interaction diagram

All the validations for each block in the Blockchain system will be divided into multiple sections. In a generic term, validation proof of a section of a block is the ID of the most recent block that has a connection to the section. The connection between a block and a section can be defined accordingly for applications, for example, for financial transaction-based blocks, a block and a section that contains the same transaction entities can be connected. Low computational validation work is simply checking whether the proof is valid or not. Since there can be multiple mining attempts made and we don't allow any types of forking, we will use a (temporary) central entity to take care of collecting all the sectional validation work and selecting one work among them. The fee or compensation paid to the selector can be accordingly designed based on the applications, for example, we can let the rewarded miner pay the fee to the selector. In the next section, a game theoretic approach has been exploited for the selectors' fee.

The key point of this selector idea is to make each block-warehouse take turns to perform selector service block by block and such a rotation must be as fair as possible. Without the use of a central server to choose the selectors, the completely fair rotation wouldn't be possible. So we are targeting for the fair enough or probabilistically fair rotation of selector service. For this purpose, our scheme requires every block-warehouse to declare and introduce itself to the other block-warehouses so that every block-warehouse has the same list of the block-warehouses. Later every block-warehouse is expected to be able to locate the same selector based on the list and the rule.

The first attempt of designing the selector rotation rule is to use the block-warehouses' ID's (this can be some publicly verifiable information such as the public key of the miner). We can start the rotation of selector service by choosing the block-warehouse with the smallest ID for the first block and then choose the block-warehouse for the second smallest id for the next block and so on. This approach may suffer from an unfair choice of selectors by indefinitely avoiding a block-warehouse as the selector. Assuming new block-warehouses with smaller ID than that of the current selector declare themselves after each block verification, then any block-warehouse with the ID which is higher than that of the current selector will be indefinitely avoided. Depending on the fee/compensation scheme for the selectors, this unfairness can be amplified to become a significant issue.

To avoid such a situation and maintain the probabilistically fair rotation, we are using block-related information, such as the hash code of the block, which is assumed to be pseudorandom and hence each possible hash code is almost equally likely. The two block-warehouses with the ID that are closest to the block-related information will be chosen as the selectors for the block. Since this type of choice depends on the block-related information, if anyone who announces the new block can manipulate this information, the fairness can be broken. However, if the block-related information is properly processed, for example into the hash code, it can be computationally hard to manipulate to break the fairness. Hence this approach shows a probabilistically fair rotation.

For each block, two block-warehouses are chosen as the positive and negative selectors. The positive selector's main role is to divide the block into sections and collect sectional validation proofs and select one of them for reward. The negative selector's main role is to collect invalidation proof (which points out the wrongdoing of positive selectors or validators) and announce the wrongdoing or the finalization of the block otherwise. The two selectors can invalidate other's previous work.

C. Block Generation Process

As soon as a positive selector has been determined through the process mentioned in III-B, the positive selector will start sequentially working in two processes: 1) validating the previous negative selector work and, 2) dividing the block into a set of non-overlapping sections. At the time of validating the previous negative selector's work, the invalidator can submit the invalidation report with the proper proof. If any report is found, the positive selector will penalize the reward of the negative selector and redistribute the rewards to the invalidators and the positive selector itself. The positive selector will finalize the previous block including the information i.e., the penalty of the negative selectors, reward for the invalidators by signing with its public ID. Then the positive selector will announce the set of sections for the current block to the network through signing with its public ID. Validators can choose any section for validating. Now the validators will sign with their public ID and submit their validation proof to the positive selector. In the meantime, the positive selector also does the validation work. After getting the submissions, the positive selector will select the first right validator from the submitted validation proofs which will match with its validation proof. Afterward, positive selector signs with its public ID and announce the winner for each section. Now if no error is reported by the invalidarors, the block will be recorded by the negative selector through signing the block and including all the information, i.e., positive selector, negative selector, winners, and reward for each section. If any error is reported in the validation process the invalidators can bring

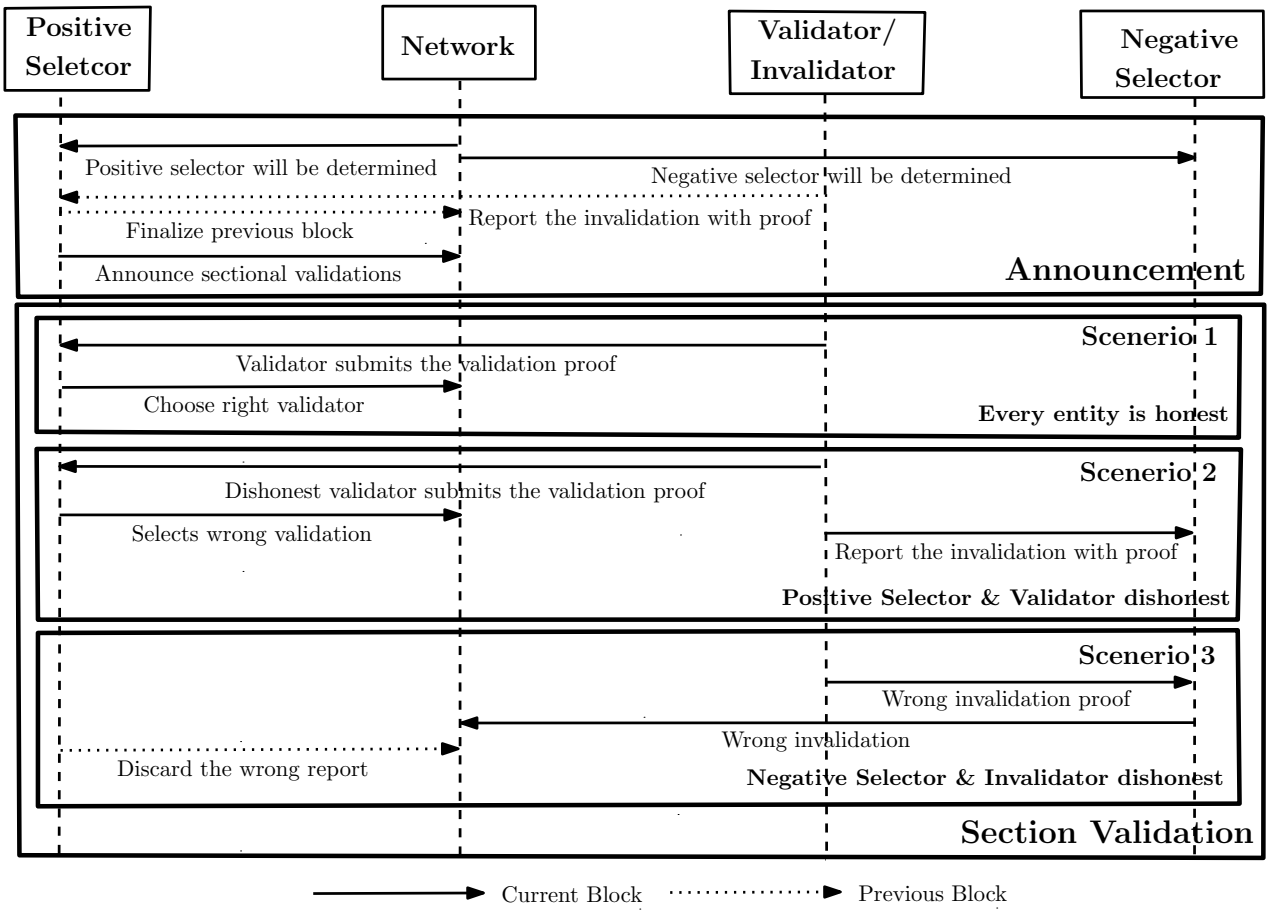


Fig. 2: Workflow of section announcement and section validation

themselves to the action of the block creation process by notifying it to the negative selectors with proper proof. Some examples of wrongdoings in the validations are intentionally or unintentionally choosing wrong validation, choosing a validator whose submission time is significantly later than other legitimate validators, etc. In the case of wrongdoing, the negative selector can deduct the reward of the entities involved and redistribute the reward to the positive selector, winner validators, and the negative selector. The negative selector now forms the block by recording all the information in the block header and will wait for the finalization of the next block positive selector.

The whole block generation process can be divided into two processes: 1) Announcement and, 2) Sectional Validation. The main part of the block generation process and three different scenarios in generating the block have been pictorially described in the workflow Figure 2. Three scenarios are 1) All the entities are honest, 2) Positive selector or Validator is dishonest, and 3) Negative selector or Invalidator is dishonest. In the scenario, 3, the wrong invalidation recorded by the negative selector will be handled by the next block positive selector. Apart from these three scenarios, another scenario could exist where all the entities involved are dishonest. However, the probability of the practical existence of that type of scenario is low. We analyze the scenario using *Nash Equilibrium* in IV-A.

IV. Analysis

A. Game Theoretic Analysis

Our proposed consensus approach assumes that the entities are rational decision-makers, i.e., will expect maximum outcome for their strategies in their interactions of consensus given the other entities' strategies. Such a system can be analyzed in terms of interaction through Game theory based mathematical tools if each entity's behaviors can be predicted. One of the most influential and important tools is Nash Equilibrium [14] which specifies the optimal outcome of an interaction where no participants have the motivation to deviate from their strategies. According to Osborne and Rubinstein [15] this interaction process can be classified as *games* and the entities are *players*.

The optimum condition of our consensus approach has been analyzed by calculating the Nash equilibrium. To calculate the Nash equilibrium, we set up two validation games of the different behavioral situations (Honest & Dishonest) by predicting the payoff of every entity in the system. Dishonest behavior can be intended or unintended. The validation game can be classified either "At least one honest selector is present"

TABLE I: When at least one honest selector is present in the validation process

Entities Behavior		Positive Selector & Negative Selector											
		00			01			10			11		
Validators	0	5	500	500	5	500	-10000	(10000+5)/2	-10000	(10000+5)/2	5	-10000	-10000
	1	-100	500	600	-100	500	-10000	-100	-10000	10000+(500+100)	-100	-10000	-10000

TABLE II: When there is no honest selector is present in the validation process

Entities Behavior		Positive Selector and Negative Selector											
		00			01			10			11		
Validators	0	5	500	500	0	-10000	(500+10000)/x	(5+1000)/2	-10000	(5+1000)/2	0	500	500
	1	-100	500	500+100	5	-10000	(500+1000)/(x+1)	-100	-10000	500+(100+10000)	-100	500	(500+100)/x

or "No honest selector is present". For the sake of simplicity, both games are formed only for one section of a block. With no logical difficulty, these two games are valid for all sections in a block. The payoff matrix for these two games is shown in Table I and Table II where the columns are representing the strategies of the selectors' behavior and the rows are representing the strategies of the validators. The table data cell representing the pay off for the validators, positive selectors, and negative selectors respectively for each combination of strategy among the selectors and validators. In the table, honest behavior is represented as '0' and dishonest behavior is represented as '1'. The Nash equilibrium has been calculated using Gambit [16], an open-source collection of tools, to perform the calculations in game theory. We used the global newton method [17] approach to calculate the Nash equilibrium.

To form and implement the previously discussed two games we assumed that a validator's highest pay off is 100 and the selector's highest payoff is 10000. Moreover, we assumed that honest pay off is $5, 500,$ and 500 and dishonest pay off (penalty) is $-100, -10000, -10000$ respectively for the validators, positive selector and negative selector.

B. One honest selector is present in validation

In this game, one honest selector is present in the validation who can point out the dishonest behavior of the entities. The forfeited payoff from the dishonest entities will be distributed to the other honest entities. Nash equilibrium result is shown with the probability of 1 for all the honest behavior of the entities and 0 for all the dishonest behavior of the entities.

C. No honest selector is present in validation

As there is no honest selector, the dishonest entities will receive the highest payoff. The forfeited payoff from the honest entities will be redistributed to the dishonest entities. In the payoff matrix, 'x' represents the number of the colluding selectors. We use a different number of colluding selectors with different "penalty ratio" (multiplied to the base penalties to scale them up) to calculate the Nash equilibrium. From the result, we see that Nash equilibrium is reestablished into the probability of 1 for honest entities and 0 for dishonest entities when there is a certain number of colluding selectors present in the system. Two 3D plot for the positive selectors Nash profile with penalty ratio of 2 & 5 has been shown in Figure 3a and Figure 3b where X, Y, Z axis represents the honest behavior Nash probability, dishonest behavior Nash probability and number of colluding selectors respectively. Figure 3c is showing a linear relationship between the number of colluders and the number of selectors which also indicates the lower bound of dishonest selectors.

D. Queuing Theoretic Analysis

In this section, we will analyze the expected waiting time of a validator by setting up a first come first serve queue model. Let's consider that an indefinite number of validators can submit their validation work for each section. Validators arrive at the queue according to the Poisson process with a rate of λ. After the validation check validators leave the queue with a rate of μ. Moreover, assume that both the arrival time of the validator and service time for the selector is exponentially distributed. As the selector validates one validation at a time, the queue model can be represented as $M/M/1$ queue model [18]. The queue model and flow diagram for $M/M/1$ queue are shown in Figure 4.

Let p_n be the probability that the system is in state n, i.e., number of validators occupying the queue. From the flow diagram it can be shown that,

$$\lambda p_n = \mu p_{n+1} \Rightarrow p_{n+1} = \frac{\lambda}{\mu} p_n$$

Consider $\rho = \frac{\lambda}{\mu}$ and total sum of state probability will be equal to 1.

$$1 = \sum_{0}^{\infty} p_i = \sum_{0}^{\infty} \rho_i p_0 = \frac{p_0}{1-\rho} \Rightarrow p_0 = 1 - \rho.$$

Through using Little's theorem [19] different useful characteristics can be derivable. The expected number of validators in the system, $E[N]$

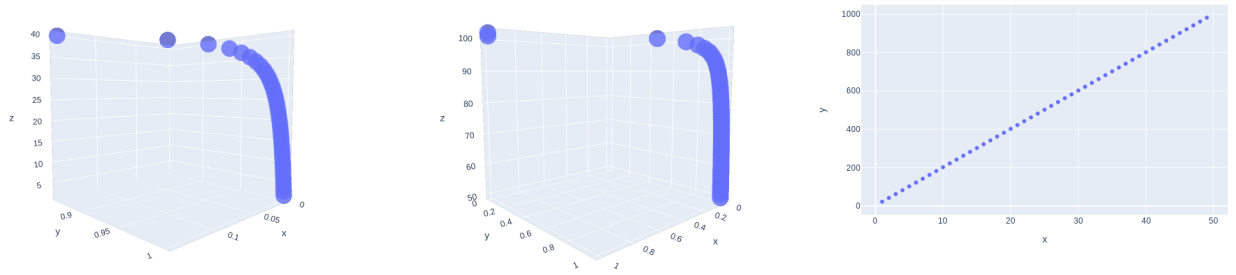
$$E[N] = \sum_{0}^{\infty} n p_n = \sum_{0}^{\infty} n(1-\rho)p^n = (1-\rho)\sum_{0}^{\infty} n p^n = \frac{\rho}{1-\rho}$$

Total time in the queue for a validator can be computed using Little's theorem,

$$E[N] = \lambda E[T] \Rightarrow E[T] = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu - \lambda}$$

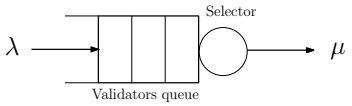
Now total expected time in the whole queue system T of a validator is composed of total expected time in the queue T_q for a validator and expected validation time (service time).

$$E[T] = E[T_q] + \frac{1}{\mu} \Rightarrow E[T_q] = \frac{1}{\mu - \lambda} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}$$

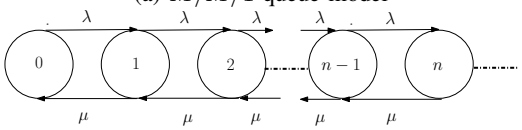


(a) When penalty ratio 2 (b) When penalty ratio 5 (c) Number of colluder vs Penalty ratio

Fig. 3: Nash profile of the positive selector



(a) $M/M/1$ queue model



(b) Flow diagram for $M/M/1$ queue model

Fig. 4: $M/M/1$ queue

Similarly, expected number of validators in the queue N_q can be derived as,

$$E[N_q] = \frac{\rho^2}{1-\rho}.$$

V. Conclusion

The root cause of the most Blockchain attacks, i.e., 51% attack, DDoS attack, DNS attack, consensus delay, selfish mining is the centralization in mining or validation and forking in the chain. In this paper, we proposed a Blockchain consensus mechanism that can effectively tackle these two issues and withhold against the above-mentioned attacks. The features of choosing the selectors in a probabilistically fair rotation can keep the Blockchain system decentralized. On the other hand, dividing a block into multiple sections can enhance user participation irrespective of users' computational power and storage. Moreover, by introducing, multiple validations of each block the security of the Blockchain system can be enhanced.

Using our Blockchain consensus mechanism based on the application separate reward mechanism can be designed. However, using two game theoretic models we show that with a proper incentive or reward mechanism entities in the system will be discouraged from launching any attack or doing any misbehavior. In this paper, we also show that a prospective user of this consensus mechanism can use queuing theory as a supporting tool to get an idea about the congestion and waiting delay in the network. The future work includes the design of a standalone Blockchain system using this consensus mechanism.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf."

[2] "Blockchair: Blockchair is the most private search engine for for bitcoin, ethereum, ripple, bitcoin cash, litecoin, cardano, telegram open network," https://blockchair.com/markets, accessed: 12/28/2019.

[3] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," in *Proceedings of the 4th International Conference on Peer-to-Peer Systems*, ser. IPTPS'05. Berlin, Heidelberg: Springer-Verlag, 2005, p. 205–216.

[4] S. Wilkinson and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.

[5] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 53–65. [Online]. Available: http://dl.acm.org/citation.cfm?id=646334.687801

[6] D. Vorick and L. Champine, "Sia: Simple decentralized storage," 2014.

[7] S. Kamara, "Proofs of storage: Theory, constructions and applications," in *Algebraic Informatics*, T. Muntean, D. Poulakis, and R. Rolland, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 7–8.

[8] J. Benet, "IPFS - content addressed, versioned, P2P file system," *CoRR*, vol. abs/1407.3561, 2014. [Online]. Available: http://arxiv.org/abs/1407.3561

[9] P. Labs, "Filecoin: A decentralized storage network," 2017.

[10] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, p. 95–102, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3212998

[11] Bitcoin network hashrate. [Online]. Available: https://data.bitcoinity.org/bitcoin/hashrate/

[12] M. Vasek, M. Thornton, and T. Moore, "Empirical analysis of denial-of-service attacks in the bitcoin ecosystem," vol. 8438, 03 2014, pp. 57–71.

[13] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman

[14] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950. [Online]. Available: https://www.pnas.org/content/36/1/48

[15] P. J. Reny, M. J. Osborne, and A. Rubinstein, "A course in game theory," 1994.

[16] R. Mckelvey, A. Mclennan, and T. Turocy, "Gambit: Software tools for game theory, version 16.0.1." Tech. Rep., 2016. [Online]. Available: http://www.gambit-project.org

[17] S. Govindan and R. Wilson, "A global newton method to compute nash equilibria," *Journal of Economic Theory*, vol. 110, pp. 65–86, 05 2003.

[18] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer Publishing Company, Incorporated, 2010.

[19] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*, 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.