An incentive compatible reward sharing approach for shard-based blockchains

1st Mozhdeh Hemati
department of computer engineering
Amirkabir University of Technology
Tehran, Iran
m.hemati0094@aut.ac.ir

2nd Mehdi Shajari
department of computer engineering
Amirkabir University of Technology
Tehran, Iran
mshajari@aut.ac.ir

2021 29th Iranian Conference on Electrical Engineering (ICEE) | 978-1-6654-3365-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICEE52715.2021.9544332

***Abstract*— Sharding is a way to solving the scalability problem in the blockchain. The sharding method uses traditional consensus algorithms, such as PBFT. In traditional consensus algorithms, network nodes can register blocks by working together. In these algorithms, an important issue is how to motivate group nodes to participate in the consensus algorithm so as not to cause collusion or free-riding. In this paper, we first examine the previous solutions that try to solve the incentive mechanism in sharding blockchain and show that these solutions do not consider the leader's role in involving other nodes in the consensus algorithm. By the game theory, we show that the leader for increasing his benefit prevents some nodes from participating in the consensus algorithm, which creates collusion. We propose a solution to motivate group nodes in the sharding method and by the game theory prove that this solution is incentive compatible and also does not causes any collusion.**

Keywords—blockchain, consensus algorithms, traditional consensus algorithms, sharding, incentive compatible.

I. INTRODUCTION

Nowadays, blockchain is an essential word in the industry and academia. Blockchain technology was first developed solely for digital currency exchange, but its features, such as decentralization, led to widespread applications.

Blockchain began with the introduction of Bitcoin [1]. Bitcoin was the first system built on blockchain and faces many problems, one of which is scalability. In general, the scalability issue in blockchain can be considered in two dimensions: 1. Low transaction number per time unit, 2. High memory usage to store all transactions.

In Bitcoin blockchains, transaction log time is long and reducing this time is not possible due to a number of security issues. For example, Bitcoin, which uses the proof-of-work consensus algorithm to establish security, requires that only one block be logged per 10 minutes because the system security is compromised by decreasing this time. Meanwhile, other payment systems such as Visa and MasterCard have transaction rates of up to 5000 transactions per second. Also, it is not possible to increase the block size. By increasing the block's size, processing and propagating the block on the network take more time; this delay in sending the block causes security issues. Increasing the delay time increases the probability of forking and, as a result, increases the probability of double-spending attacks.

Another problem is the storage that needs for recording the transactions and blocks in the blockchain. In bitcoin, all the transactions should be registered in the blockchain so a large amount of memory is required to store bitcoin data; This is an important scalability issue.

Various solutions have been proposed to solve the scalability problem. In general, these solutions can be divided into off-chain, on-chain and side-chain solutions [2].

The off-chain solution, also called the second layer solution, adds an extra protocol layer to the main blockchain. One idea of the off-chain solution is to create off-chain network channels between the two parties, through which the parties can establish communication. The only final transaction is published in the blockchain network, which reduces the number of transactions registered in the main chain. Lightning network [3] and Raiden network [4] are two examples of off-chain solutions.

Another solution for the scalability problem is the side-chain solution. In this solution, there is one main chain and one or more side chains. The side chains are connected to the main chain and have a two-way connection with it. Each side chain performs its task, thereby preventing the overloading of the main chain by reducing tasks of it. An example of this solution is Plasma [5].

The on-chain approach is another solution for scalability. Solutions in this category, that also called first layer solutions, require changes in the core features of the blockchain, such as resizing the block and reducing block creation time. For example, Segwit [6] is a member of this category. In this solution, without changing the block size and only by removing the signature and witness data from the block, more transactions can be placed in one block. Another solution in the on-chain category is sharding. The main idea in sharding is to split the network into small groups and process the transactions in parallel. Elastico [7], Omniledger [8], and RapidChain [9] are some examples of sharding.

One of the problems with sharding solutions is ignoring the importance of incentive mechanism for motivating the network nodes for participating in consensus algorithm and creating

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

block. The initial solutions to this method, such as Elastico [7], do not consider an incentive mechanism. Without the incentive, we cannot be sure whether network nodes follow each step of our solution or not.

Many solutions were proposed to motivate network nodes, such as Repchain [10]. In many of these solutions, the group leader's role in involving nodes in the consensus algorithm has been ignored. In traditional consensus algorithms, a certain number of signatures is enough to build a block and reach a consensus. To build a block, the leader in this system can collect as many signatures as is enough to create the block and not involve the rest of the network nodes in building the block. For example, in PBFT [11] that is a kind of consensus algorithm, the leader should collect at least $2f+1$ signatures to create a block that f is the number of byzantine nodes in the group. So it is clear that the leader can only collect $2f+1$ signatures and ignores other nodes' signatures. By this work, the leader can deprive some nodes of a getting reward, and also it causes the collusion between the network nodes.

We discuss sharding in more detail in Section II. In this section, the sharding methods that solve the problem of incentive are explained. In Section III, the proposed solutions are discussed and analyzed by game theory. In Section VI, a new solution is proposed to incentivize nodes in sharding methods, and it is proved by the game theory that it is an incentive-compatible solution. In Section V, the proposed solution is evaluated in a simulation environment. In Section VI, future works are discussed, and, finally, in Section VII, the conclusions are presented.

II. SHARDING

The main idea of sharding in the blockchain is to split the transactions into disjoint sets and each transaction in each set is processed simultaneously in parallel with other transactions in other sets. In order to process transactions in parallel, all the nodes are divided into different groups, each of which is responsible for validating a set of transactions. In sharding, each group only needs to validate and maintain its own transactions, while in the Bitcoin network, each node must store all data locally and validate all the transactions.

This approach would be very simple in the system, in which nodes enter the system with identity. However, it will be complicated in the system, in which nodes enter anonymously and the probability of having malicious nodes is high.

In general, the steps that can be taken in the sharding method can be considered as follows:

1. Assigning identity to nodes: In the first step, the identity of the node that will log into the network is determined. In order to prevent attacks like the Sybil attack, nodes need to prove their identity by solving a difficult computational problem, such as the proof-of-work problem, for accessing the network.

2. Forming a group or committee: At this point, the nodes are assigned into different groups, depending on their identities in the previous step. Once the group of each node is determined, all the nodes must find and connect to the nodes that are in the same group.

3. Using consensus algorithm within the committee: In each group, nodes process specific transactions and use traditional consensus algorithms such as PBFT [11] to create a block.

4. Using consensus algorithm between committees: Finally, after each group creates its block, all the groups must send their blocks to the final committee to form the final block.

One of these methods' problems is that none of them has used an incentive mechanism to motivate nodes to participate in consensus algorithm. For example, Elastico [7], Omniledger [8], Rapidchain [9], that are an important solution in sharding, don't consider any incentive mechanism in their system. In the next section, we discuss about incentive and its important role in sharding methods.

A. Incentive mechanism in sharding systems

The sharding solutions use one of the traditional algorithms to solve the consensus within each group. In traditional consensus algorithms, nodes were considered in two types: the honest and the byzantine. Honest nodes act under the terms of the algorithm under any circumstances and byzantine nodes can do any arbitrary actions to harm the system. While this assumption cannot be correct for a system such as a blockchain. In the blockchain, the nodes that are not byzantine can be honest or rational. Rational nodes may violate the protocol to increase their profits. So, when we do not consider any incentives for participating in the consensus algorithm, the rational participants do not see any profits for participating in the consensus algorithm. So it is necessary to consider an incentive mechanism in the sharding methods. In the following, we present some sharding methods that consider incentive.

A.1 Game theory analysis of Elastico

Manshaei et al. [12] presented a work that tried to solve the incentive problem in sharding-based blockchain. This paper discussed considering the behavior of rational nodes, which has not been considered in important papers. Rational nodes always choose the strategy that has the best benefit for them. If following the protocol is not in their best interest, they will violate it. However in other methods, non-malicious nodes are considered completely honest that they follow the protocol under any circumstances. In this solution, the authors modeled the node's behavior with a single shot game. The solution proposed in this paper was to reward only those who participated in the consensus algorithm, rather than dividing the reward equally among all the nodes of the group. This paper proved that equal distribution of rewards between the nodes creates a free riding, So, the Nash equilibrium of the game was not the cooperation of all the nodes in the consensus algorithm. While if the reward is equally distributed only among the nodes participating in the algorithm, the Nash equilibrium of the game will be the participation of all nodes in the algorithm.

A.2 Repchain

Repchain was presented by Huang et al. [10] and is another sharding solution that incentivizes nodes. This paper incentivized nodes by the reputation of nodes that participate in the consensus algorithm. The goal is to make a system that besides security, provides enough incentive for nodes.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

In this system, the group leader is chosen according to top reputation and, finally, the reward of creating each block is divided between the nodes that participated in the consensus algorithm, according to their reputation.

Repchain has two blockchains in this system, one for maintaining the reputation and the other for main data such as transactions. At the end of each epoch, the reputation of each node is calculated and recorded with the agreement of the group nodes in the blockchain. The reputation of each node is calculated according to the action of this node regarding participation in the consensus algorithm.

Repchain randomly puts them into different groups according to their reputation. Each group uses the Raft algorithm to validate transactions and consensus on valid transactions.

A.3 Harmony

Harmony [13] is a new system based on sharding, the main idea of which is that the vote of each node is different from each other and each vote has its weight. The weight of nodes is according to their stake in the network. The weight of each node vote is calculated according to their stake and, then divided into different groups based on their weight. Afterwards, a consensus algorithm based on PBFT is used to reach consensus in each group. Eventually, the reward gained from block registration is distributed among different nodes that participated in consensus according to their weight.

In all the solutions outlined in this section, the role of the leader in allocating rewards to the other group nodes is not considered. A leader can deprive some nodes from participating in consensus algorithm and don't put their signatures in the block and collusion with other nodes in order to get more rewards. In the next section, we discuss this problem and show through the game theory that it is possible to break these solutions.

III. Analysis of previous approaches

Most of the traditional consensus algorithms that use in the sharding methods require a leader to create a primary block and send this block to other nodes of the group for agreement. In these algorithms, the leader must collect a certain number of signatures from other nodes. For example, in the PBFT consensus algorithm, if the number of byzantine node in the group is equal to *f*, the block should contain *2f+1* signatures to be a valid block. The leader can arbitrarily collude with some nodes and only involve them in the consensus algorithm. ignoring some nodes and creating collusion with other nodes causes some security problems in the sharding method.

In the following, we show that in the sharding methods, the leader tends to violating the protocol and does not involve some node from participating in the consensus algorithm or ignores their signatures and does not put their signatures in the block, so Tthese solutions are not incentive-compatible. To this purpose, we first provide a definition of incentive-compatible, then model the previous solutions and show that they are not incentive compatible according to this definition.

Definition 1. A mechanism is incentive-compatible, when all of the player act truly, it is best strategy for one node to act truly too.

A. Game Analysis

The model of game is Bayesian. Bayesian game is a strategic game with incomplete information which has 6 components:

1. Players, 2. States, 3. strategy set, 4. Signal , 5. Belief, 6. Utility function;

In this game, we consider the leader as one player and the node in the group which wants to participate in the consensus algorithm as another player. We also assume that the type of leader is honest and rational. The state set in this game shown by Ω is the leader type and defined as relation 1.

$$\Omega = \{Rational, Honest\} \quad (1)$$

The strategy set of every rational leader is defined as set 2 and the strategy set of every honest leader defines as set 3. In this relations, *f* means "following the protocol and involving other nodes in the consensus algorithm" and *d* means "protocol deviation and not involving all the nodes in the consensus algorithm". An honest node has only one strategy and this is following the protocol.

$$S_R = \{f, d\} \quad (2)$$

$$S_H = \{f\} \quad (3)$$

The strategy set of non-leader nodes defines as set 4. *p* means "participating in the consensus algorithm", and *n* means "not participating in the consensus algorithm".

$$S_n = \{p, n\} \quad (4)$$

In each state, it is clear to the leader what his type is. Thus, in each state, the leader receives a different signal, so the signal function shown by τ for the leader is defined as relation 5.

$$\tau_i(k) \neq \tau_i(j) \quad \text{for all } k,j \in state \quad i=\text{leader} \quad (5)$$

In each state, each non-leader node that has no information about the type of leader, receives the same signal that the leader is honest or not. So, the signal function for every non-leader node is defined as relation 6.

$$\tau_i(k) = \tau_i(j) \quad \text{for all } k,j \in state \quad i \neq \text{leader} \quad (6)$$

According to the signal function, for the non-leader node, belief in each state is that the leader with *p* probability is rational and with *1-p* probability will be honest. Because a non-leader node does not have any information about leader, *p* is equal to 1/2. For the leader, his belief in any state is equal to 1; it means that the leader always knows his type. Figure 1 shows the tree of our game.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

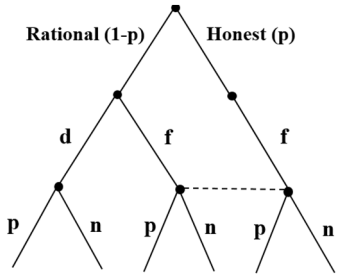


Fig. 1. Game tree

In Bayesian games, the utility function for each node is defined as relation 7. It means, according to the received signal by a node, what is the probability that the game is in a W-state and what action or strategy will other nodes do in this state.

$$\sum_{w\in\Omega} \Pr(w \mid \tau_i)\; u_i((a_i, \hat{a}_{-i}(\mathrm{w}), \mathrm{w}) \quad (7)$$

In the previous solutions the reward of one block is distributed between all of nodes that participated in the consensus algorithm. So, if the block reward shown by R, and the number of nodes that participated in the consensus algorithm shown with k, then B that shown the reward that each participating node and leader receive for registering a block is equal to relation 8.

$$B = \frac{R}{k} \quad (8)$$

İn these solutions, the cost of participating in the consensus algorithm that shown by C is equal to relation 9. In this relation, Ce is the cost for entering the group; for example, in some solution, this cost is equal to solving a proof of work problem. Cp is the cost of participation in the traditional consensus algorithm.

$$C = Ce + Cp \quad (9)$$

According to relation 7, the definition of B in relations 8 and the definition of C in relation 9, the utility function of nodes is defined as table I and II.

Table I. Utility table when type of leader is rational

	f	d
p	$\frac{R}{k}-\mathrm{C}, \frac{R}{k}-C$	$-C, \frac{R}{k-1}$
n	$-\mathrm{C}_e, \frac{R}{k-1}-C$	$-\mathrm{C}_e, \frac{R}{k-1}-C$

Table II. Utility table when type of leader is honest

	f
p	$\frac{R}{k}-\mathrm{C}, \frac{R}{k}-C$
n	$-\mathrm{C}_e, \frac{R}{k-1}-C$

If k nodes participate in the consensus algorithm and the leader follows the protocol and involve all the k nodes in the consensus algorithm, then, the leader's utility function is equal to relation 10.

$$u_{leader}(f_{leader}, p_{\text{-}leader}) = \frac{R}{k} - C \quad (10)$$

If k nodes tend to participate in the consensus algorithm and the leader does not follow the protocol and do not involve x nodes of these k nodes in the consensus algorithm, then, the utility function of the leader is defined as relation 11.

$$u_{leader}(d_{leader}, p_{-leader}) = \frac{R}{k-x} - C \quad (11)$$

It can be concluded from relation 10 and 11 that breaking the protocol and deviating from it, when other nodes follow the protocol, would be more beneficial to the leader. So the previous solutions are not incentive compatible. As a result, the leader tends to prevent some nodes from participating in the consensus algorithm and also creating collusion to increase its profit. In each solution that reward is distributed between participants, there is a tendency to create collusion and depriving some nodes of participating in the consensus algorithm. For example, in the Repchain that leader get the half of the reward and the other half is distributed between non-leader participants, by the same way, the network nodes tend to create collusion and deprive some node to increase their benefit. In the next section, we will present a solution that solves this problem.

IV. THE SUGGESTED PROTOCOL

We consider the number of all the nodes in a group is equal to n, and the number of byzantine nodes is equal to f. We assume that a PBFT consensus algorithm is used for inter-shard consensus of each group, so it is necessary to collect more than $2f+1$ signature for creating a valid block. We use the public key structure, so that everyone has a pair of keys containing both a public key and a private key. Thus, everyone uses their private key to sign and others with their public key can verify and validate his signature. We show that our proposed protocol is incentive-compatible.

A. Game Analysis

As mentioned before, the game that we consider for modeling is a Bayesian game. All of the components of this game are defined as before, except utility function.

If we consider the total amount of rewards earned for creating and registering a block equal to R, the most important part of the protocol here is how to divide this R-value between nodes in the network.

We consider a constant and a maximum as the reward for the leader and other nodes in the network for building a block. However, achieving this maximum requires the right performance of the nodes on the network. Right performance for the leader is defined in such a way that the leader does not prevent other participants from participating in the consensus algorithm. For other nodes, the right performance is defined in

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

such a way that the node participates truly in the consensus algorithms. So, the reward of the leader, if a block is registered in the blockchain, is defined as:

$$B_l = \begin{cases} \frac{2R}{(n+1)} \times \frac{k}{n} & k \geq 2f+1 \\ 0 & else \end{cases} \quad (12)$$

\

R is the reward received for creating a valid block and k is the number of signatures the leader collected from other group nodes. According to this relation, if the leader prevents some node from participating in the consensus algorithm, the reward that can earn by building a block become decreasing.

The reward of other nodes that participate truly in the consensus algorithm, if a block is registered in the blockchain, is defined as relation 13.

$$B_i = \frac{R}{n+1} \quad (13)$$

The maximum reward for the leader is twice the maximum reward for each node. The cost function of each node is also defined as relation 14. *Ce* is the cost for entering the group; for example, in some solution, this cost is equal to solving the proof of work problem. *Cp* is the cost of participation in the traditional consensus algorithm.

$$C=Ce+Cp \quad (14)$$

According to relation 7, the definition of B in relations 12 and 13 and the definition of C in relation 14, we define the utility function of nodes in our game. Table III and IV show the utility function in this game.

According to Tables III and IV, If the leader does not involve a non-leader node in the consensus algorithm, a signature will be deducted from his signature collection and reduce his reward, and also the non-leader node will not receive a reward and will only have the cost of C_e. In the same way, if the non-leader node does not participate in the consensus algorithm, he will only pay the cost C_e and reduces the leader reward. Based on the utility function we have defined, we will first explain more about the utility function of each node and, then investigate the best strategy that each node will choose.

Table III. Utility table when type of leader is rational

	f	*d*
p	$\frac{R}{(n+1)} - C, \frac{2R}{(n+1)} \times \frac{k}{n} - C$	$-C, \frac{2R}{(n+1)} \times \frac{k-1}{n} - C$
n	$-C_e, \frac{2R}{(n+1)} \times \frac{k-1}{n} - C$	$-C_e, \frac{2R}{(n+1)} \times \frac{k-1}{n} - C$

Table IV. Utility table when type of leader is honest

	f
p	$\frac{R}{(n+1)} - C, \frac{2R}{(n+1)} \times \frac{k}{n} - C$
n	$-C_e, \frac{2R}{(n+1)} \times \frac{k-1}{n} - C$

B. The Best Strategy for Each Node

We will show that it is always the best strategy for rational nodes to follow the protocol when other nodes follow the protocol. First, we demonstrate to a non-leader that validating the received block and sending it to the leader will be the best strategy while others follow the protocol. Then, we show the same thing to the leader. If the leader participates all nodes in the consensus algorithm while other nodes also follow the protocol, it is the best strategy for the leader.

If other nodes in the group follow the protocol, the utility function of following the protocol for a non-leader node i, defines as relation 15.

$$u_i(p_i, d_{-i}) = (\frac{R}{n+1}) - C \quad (15)$$

While if a non-leader node i violates the protocol in this condition, the utility function is defined as relation 16.

$$u_i(n_i, f_{-i}) = -Ce \quad (16)$$

According to relations 15 and 16, it is obvious that violating the protocol would not be profitable for a non-leader when we assume that everyone in the group is following the protocol. So, "following the protocol" is the best strategy for a non-leader node. We will show that following the protocol, while other nodes also follow the protocol, will be the best strategy for the leader. we assume that the leader puts k signature of other nodes in the new block. If other nodes follow the protocol, but the leader violates the protocol and does not put some signatures (for example, x signature) of k signature on the block or does not involve x nodes from k participant's node, then, the leader utility function is defined as relation 17.

$$u_l(\mathrm{d}_l, \mathrm{p}_{-l}) = \begin{cases} (\frac{2R}{n+1} \times \frac{k-x}{n}) - C & \text{if} \quad k-x \geq 2f+1 \\ 0 & \text{else} \end{cases} \quad (17)$$

If the leader also follows the protocol and puts all signatures in the block and involve all the node in the consensus algorithm, his utility function is defined as relation 18.

$$u_l(f_l, \mathrm{p}_{-l}) = B_l - C = (\frac{2R}{n+1} \times \frac{k}{n}) - C \quad (18)$$

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

It is obvious that $u_l(f_l, p_{-l}) \geq u_l(d_l, p_{-l})$, Thus, following the protocol for the leader, while other nodes follow the protocol, is the best strategy. So, we show that following the protocol will be the best strategy for everyone. As a result, our protocol is incentive compatible.

C. Probability of create collusion

According to this reward sharing mechanism, it is clear that a group of nodes with collusion cannot increase their benefit because each node's reward is separated from other nodes. By creating collusion, the reward that one node can obtain in this system cannot be changed. So a group of nodes, by creating collusion, cannot get more reward, so they have no incentive to create collusion.

V. Evaluation

We simulated a blockchain system based on sharding, in which a simple traditional consensus algorithm similar to PBFT was implemented. In the game theory proof that was proposed, it was assumed that the utility function for each node is positive. In fact, the reward received for each node is more than the cost incurred. In this simulation, we examine for what values of the defined parameters, the utility function will be positive, and by changing the value of the variables in each step, how the number of participants changes. We evaluated the number of participants in the consensus algorithm in three phases.

Firstly, by varying the number of rewards, Secondly, by changing the number of group nodes, Finally, by combining the two previous phases via changing the reward and number of group nodes.

As shown in the figures, the blue chart shows the number of nodes that participate and the red chart shows the number of nodes that do not participate in the consensus algorithm.

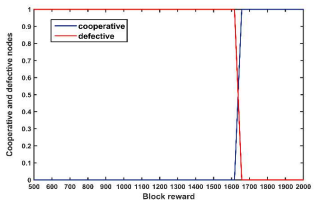
According to Figure 2, if the block reward is higher than a specific value, all the nodes of the group tend to participate in the consensus algorithm and follow the protocol because we say that the nodes participate in the algorithm if their utility function is positive.

Similarly, in Figure 3, since the reward that each node earns is related to the number of group nodes, this number should not exceed a specific limit; as the number of group nodes increases, the received reward of each node is reduced. Thus, the number of group nodes should be less than the limit in order for the utility function of each node to becomes positive and these nodes have the motivation to participate in the consensus algorithm. So, for a specific number of nodes in the group, all the nodes tend to participate in the consensus algorithm and follow the protocol.

In Figure 4, in the same way, we have a combination of the two previous phases to show that for how many group nodes and block reward, nodes are willing to participate in the consensus algorithm.

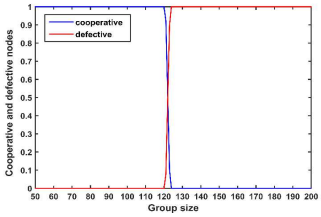
Table V. simulation parameters

C_e	Entering the group cost
C_p	Consensus algorithm cost
C_v	Transaction validation cost
N	Network size
N	Group size
Fee	Transaction fee
BR	Block reward



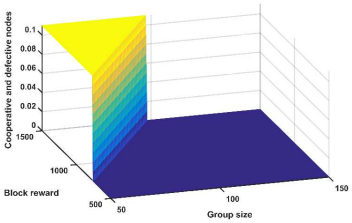
C_e	10
C_p	6
C_v	0.5
N	1000
N	100
Fee	20-40
BR	500-2000

Fig. 2. cooperative and defective nodes for the different block reward.



C_e	10
C_p	6
C_v	0.5
N	1000
N	50-200
Fee	20-40
BR	2000

Fig. 3. cooperative and defective nodes for the different group size.



C_e	10
C_p	6
C_v	0.5
N	1000
N	50-200
Fee	20-40
BR	500-1500

Fig. 4. cooperative and defective nodes for different group sizes and block reward

VI. Future Work

In our proposed solution, the problem is to ignore the byzantine nodes that can arbitrarily represent any behavior. If one of these nodes is chosen as a leader, he can prevent some of the nodes from participating in the consensus algorithms. In this case, some nodes of the network will not be allowed to participate in the group by the leader and will not receive any reward. One subject that needs to be investigated in the future is whether we can provide a consistent incentive algorithm with no such problem and we are ensured that malicious nodes will not be involved in the participating of other nodes or in rewarding to other nodes.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.

VII. Conclusion

In this paper, we presented the challenge of scalability in the blockchain. In this regard, we reviewed sharding methods that had attempted to solve the scalability problem of the blockchain. Finally, the incentive challenge, with which all these solutions were faced, was presented. In all the previous approaches, a brief explanation was provided on why recent solutions had not been fully successful in motivating network nodes. By the game theory, the drawbacks encountered with the previous solutions were introduced. Finally, our protocol was presented and it was demonstrated by the game theory that this protocol was an incentive-compatible one.

References

[1] Nakamoto, S.:Bitcoin: A peer-to-peer electronic cash system(2008)

[2] Kim, S., Kwon, Y., Cho, S.: A survey of scalability solutions on blockchain. In 2018 International Conference on Information and Communication Technology Convergence (ICTC) 2018 Oct 17, pp. 1204-1207. IEEE(2018)

[3] Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments.(2016) https://lightning. network/lightning-network-paper. pdf

[4] Raiden Network: Fast, Cheap, Scalable Token Transfers for Ethereum.(2018) https://raiden.network

[5] Poon, J., Buterin, V.: Plasma: Scalable autonomous smart contracts.(2017) https://plasma. io. 2018;72.

[6] Bitcoin Core. Segregated witness benefits. https://Bitcoincore. org/en/2016/01/26/segwit-benefits/.[Online, 2016.

[7] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P.: A secure sharding protocol for open blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 17-30.(2017)

[8] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E. and Ford, B.: A secure, scale-out, decentralized ledger via sharding. In 2018 IEEE Symposium on Security and Privacy (SP), pp. 583-598. IEEE(2018)

[9] Zamani, M., Movahedi, M. and Raykova, M.: Rapidchain: Scaling blockchain via full sharding. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 931-948.(2018)

[10] Huang, C., Wang, Z., Chen, H., Hu, Q., Zhang, Q., Wang, W., Guan, X.: Repchain: A reputation based secure, fast and high incentive blockchain system via sharding. arXiv preprint arXiv:1901.05741.

[11] Castro M, Liskov B.: Practical Byzantine fault tolerance. InOSDI 1999 Feb 22, Vol. 99, No. 1999, pp. 173-186.(1999)

[12] Manshaei, M.H., Jadliwala, M., Maiti, A., Fooladgar, M.: A game-theoretic analysis of shard-based permissionless blockchains. IEEE Access(2018)

[13] The Harmony Team.: Open Consensus for 10 Billion People. (2018)

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 08:13:59 UTC from IEEE Xplore. Restrictions apply.