


Article

Reputation-Based Sharding Consensus Model in Information-Centric Networking

Jia Shi ^{1,2} , Xuewen Zeng ^{1,2} and Yang Li ^{1,2,*}

¹ National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; shij@dsp.ac.cn (J.S.); zengxw@dsp.ac.cn (X.Z.)

² School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: liyang@dsp.ac.cn

Abstract: The various integration systems of blockchain and information-centric network (ICN) have been applied to provide a trusted and neutral approach to cope with large-scale content distribution in IoT, AR/VR, or 5G/6G scenarios. As a result, the scalability problem of blockchain has been an increasing concern for researchers. The sharding mechanism is recognized as a promising approach to address this challenge. However, there are still many problems in the existing schemes. Firstly, real-time processing speed trades off security of validation. Secondly, simply randomly assigning nodes to the shards may make nodes located very far from each other, which increases the block propagation time and reduces the efficiency advantage brought by the sharding mechanism. Therefore, we optimize a reputation-based sharding consensus model by multi-dimension trust and leverage the affinity propagation (AP) algorithm for gathering consensus nodes into shards. Given the minimal possibility to be at fault in the security of validation, clients can achieve real-time processing speed with assurance. The evaluation results show that the normalized mean square error (NMSE) between the estimated reputation value and the real reputation value of our reputation scheme is less than 0.02. Meanwhile, compared with the classical sharding scheme Omniledger, TPS performance can achieve 1.4 times promotion in the case of a large-scale blockchain network of 1000 nodes.

Keywords: blockchain; information-centric network (ICN); sharding; reputation; consensus; affinity propagation



Citation: Shi, J.; Zeng, X.; Li, Y. Reputation-Based Sharding Consensus Model in Information-Centric Networking. *Electronics* **2022**, *11*, 830. <https://doi.org/10.3390/electronics11050830>

Academic Editors: Nurul I. Sarkar and Juan-Carlos Cano

Received: 8 February 2022

Accepted: 5 March 2022

Published: 7 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Blockchain technology, characterized by its decentralized tamper resistance, shows great potential in dealing with security and trust challenges in various application scenarios, such as medical systems, internet of things and edge computing [1–3]. Various integrated systems of blockchain and information-centric network (ICN) [4] have also been a concern of researchers in order to provide a credible and neutral method to deal with the trust problem of large-scale content distribution [5–9]. However, due to the low throughput and weak scalability of traditional blockchain technology, the wider application of blockchain technology is seriously restricted. For example, Bitcoin can only process about 10 transactions per second, with a maximum block size of 1 MB and an average block cycle of 10 min [10], which are incapable of providing support for higher throughput scenarios.

In order to handle a large number of verification transactions, side chain technology and off-chain technology are proposed, which are called vertical scaling [11]. Transaction verification is processed outside the main chain. For example, Plasma builds various applications on Ethereum [12] through side chain technology. From the perspective of users, they can minimize interaction with the blockchain to reduce latency, but this scheme cannot improve the throughput of the blockchain [13].

To address this problem, researchers began to propose solutions from the perspective of horizontal scaling, aiming to improve scalability while maintaining the decentralization

and security of blockchain. Sharding is considered to be the most promising method to solve this challenge [14]. The sharding mechanism divides the whole blockchain into multiple consensus groups and allows participating nodes to process and store a few shards (i.e., only parts of the blockchain). The transactions in these pre-selected shards are processed and validated in parallel, which significantly enhances the network scalability and transaction throughput [15,16]. Various sharding protocols have been introduced such as Elastico [17], OmniLedger [18], RapidChain [19] and Zilliga [20] in existing blockchain cryptocurrencies. Researchers employed a random algorithm to maintain the fairness of the shard distribution [21–25]. However, those schemes assign shards based on simple randomness without any consideration for the reliability and efficiency in consensus procedures. Determined by a well-known blockchain trilemma [26] shown as Figure 1, when the number of nodes decreases in the consensus process, the malicious nodes can easily launch attacks. Meanwhile, simply randomly assigning nodes to the shards without considering the network distance between consensus nodes may make nodes located very far from each other, which increases the block propagation time and, hence, reduces blockchain throughput.

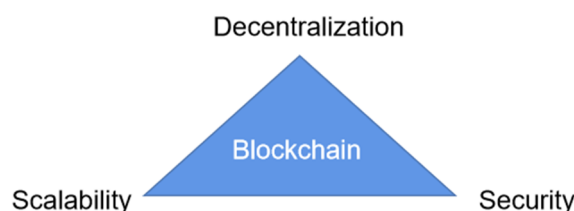


Figure 1. Blockchain trilemma. No current blockchain system can simultaneously achieve decentralization, security and scalability.

Therefore, the motivation of our paper is to improve security issues by combining the reputation of blockchain nodes. We provide a multi-dimension reputation model in which the reputation value is computed by the trust parameters including quality of service (QoS), quality of security performance (QoS_P), evaluation reputation, past reputation and recommendation reputation. A topology-based sharding scheme is also proposed to optimize the transmission of the blockchain network to reduce latency. The validator distribution among shards is achieved by the aggregated trustworthiness scores calculated by the reputation value and the network distance. Meanwhile, we introduce the name resolution system in the ICN prototype to provide advantages for the registration and query of shards and help the rapid and effective distribution of transactions.

The main contributions of this paper are as following:

1. We propose a novel multi-factor trust model for the reputation evaluation of consensus nodes. These evaluation factors measure reputation value from the perspective of objective trust, subjective trust and historical trust, so as to minimize the probability of malicious nodes controlling shards and improving the reputation by collusion and deception.
2. The shard distribution scheme is optimized by the affinity propagation algorithm (AP), which is used to find the optimal sharding group by combining security and timeliness. We introduce the calculation method of algorithm input by cosine similarity. The clustering results well consider the reduction in distance within intra-shards. All the shard clusters are evenly distributed in the network instead of gathering in a domain.
3. We introduce the definition of unequal consensus verification group and give the security rules in each epoch. It is proved that the epoch time can be extended to a few days without security problems, which greatly reduces the overhead of the sharding scheme.
4. The experimental analysis is given by OMNET++. Simulation results show that the normalized mean square error (NMSE) between the estimated reputation value and the real reputation value of our reputation scheme is less than 0.02, which proves the

reliability of our reputation evaluation mechanism. Meanwhile, compared with the classical sharding scheme Omniledger, the TPS performance can achieve 1.4 times promotion in the case of a large-scale blockchain network of 1000 nodes.

The structure of this paper is organized as follows. In Section 2, we analyze the limitations and shortcomings of the existing work. Section 3 briefly introduce the preliminaries and system model of our scheme. Section 4 presents the multi-factor model of reputation algorithm and the design procedure of our sharding protocol, including its working rules and the related parameters. Section 5 gives the security analysis of the whole scheme. The simulation results are presented and analyzed in Section 6 to characterize our sharding protocol. Finally, Section 7 offers our conclusion.

2. Related Literature

For the several famous and typical sharding mechanisms in the blockchain such as Monoxide [27], Elastico [17], Omniledger [18], Rapidchain [19], Chainspace [28] and Ethereum 2.0 [29], most of them are based on randomness, that is, shuffling the consensus nodes regularly and grouping randomly to ensure the security of the system. Moreover, in order to build good randomness, the random scheme needs to meet several important attributes: unbiased, unpredictable and public verifiability. However, these are not easy to achieve. Generating public randomness is difficult because active opponents may dishonestly bias public random choices to their advantage. Existing solutions cannot extend the blockchain to hundreds or thousands of participants as needed. Without good randomness, the security of the blockchain system is destroyed. For example, through a complex attack, such as a bribery attack [30] or 1% attack [31], an opponent may have the ability to control a temporary majority of the overall computing power (e.g., more than 50%), which in turn may damage the entire system.

To address the challenge, trust-based sharding distribution algorithms were introduced to enable the distribution of nodes among the shards based on the trust score of a node. The trust score used in sharding and leader election processes can minimize the adversary influence of malicious attacks. Ref. [32] proposed a trust model that evaluates the quality of shards by the average difference of the aggregated trust results in each shards. The trust value is obtained by peer review of each node in the process of consensus. Ref. [33] introduced a novel reputation scheme with two factors: the accuracy of valid information and aggregated contributions a consensus node has made. To ensure the security of the local shards' sub-chains, refs. [34,35] increased the reputation evaluation dimension based on its trustworthiness by the peer's customers, i.e., the customers associated with the peer in the same shard, not only the evaluation of the consensus process. The new sharding clustering method is also concerned by researchers. Ref. [36] used the adaptive hedge algorithm [37] for committee selection. It is a decision-theoretic online learning method to minimize cumulative loss of the consensus node according to the best strategy. Ref. [38] designed the trust-based shard distribution (TBSD) model based on genetic algorithm (GA) to provide a sufficiently good solution in optimization problems of sharding distribution. The optimal shards are computed based on a modified GA to quickly find a solution by genetic variation. Ref. [39] proposed a Geographical Proximity Sensing Clustering (GPSC) method based on the K-Means algorithm to reduce the network latency on consensus and ensure parallel broadcast between clusters. The method distributes the consensus nodes into several shards based on the famous K-Means algorithm. These sharding distribution schemes optimize the security and efficiency of simple randomly assigning methods to a certain extent, but they still ignore heterogeneity among the nodes. The evaluation reputation of peers is aggregated without considering the quality of service among the nodes. Those less-competent consensus nodes may become a bottleneck and hamper the system throughput. The limitations of the existing literature are briefly summarized in Table 1.

Table 1. Limitation overview of existing literature.

Scheme Name	Allocation	Attack Tolerant	Epoch Length
Elastico [17]	Randomness (PoW puzzles)	Weak, attacker behaves arbitrarily, 1% attack	10 min
OmniLedger [18]	Randomness (RandHound [24])	Medium, DDOS resistance Poor to resist 1% attack	<one day
RapidChain [19]	Randomness (PoW puzzles)	Weak, attacker behaves arbitrarily, 1% attack	>one day
Ethereum 2.0 [29]	Randomness (RANDAO [40] and VDF [41])	Medium, 1% attack resistance, Uncoordinated majority	One week
Halgamuge et al. [32]	Aggregated trust value balance	Medium, 1% attack resistance, Poor to resist collusion attack	N/A
Gang Wang [33]	Aggregated trust value balance	Medium, 1% attack resistance, Poor to resist bribery attack	N/A
YUN et al. [34]	Network coordinate balance	Medium, DDOS resistance Poor to resist 1% attack	N/A
Hao et al. [39]	Aggregated trust value balance	Medium, 1% attack resistance, Poor to resist bribery attack	N/A

In conclusion, the existing schemes leave the problems of security and efficiency unsolved. Even if the sharding scheme based on trust model is proposed, the trust scheme itself has security problems. Meanwhile, most of the clustering schemes need to determine the central nodes and the parameters are single, which is not conducive to finding the optimal sharding results. In addition, the optimization scheme proposed in the paper [32–39] does not give the selection of epoch, which is closely related to the overhead and security of the system. Based on those analyses, our scheme analyzes these directions in the next section.

3. Preliminaries and System Model

In order to better illustrate our scheme, we briefly introduce the system model of our proposal and related preliminaries in this section. The system model is shown in the Figure 2. The key components include three parts: reputation statistics, deriving and sharing the shard sets to consensus nodes and finding available shards for parallelizing transaction commitments.

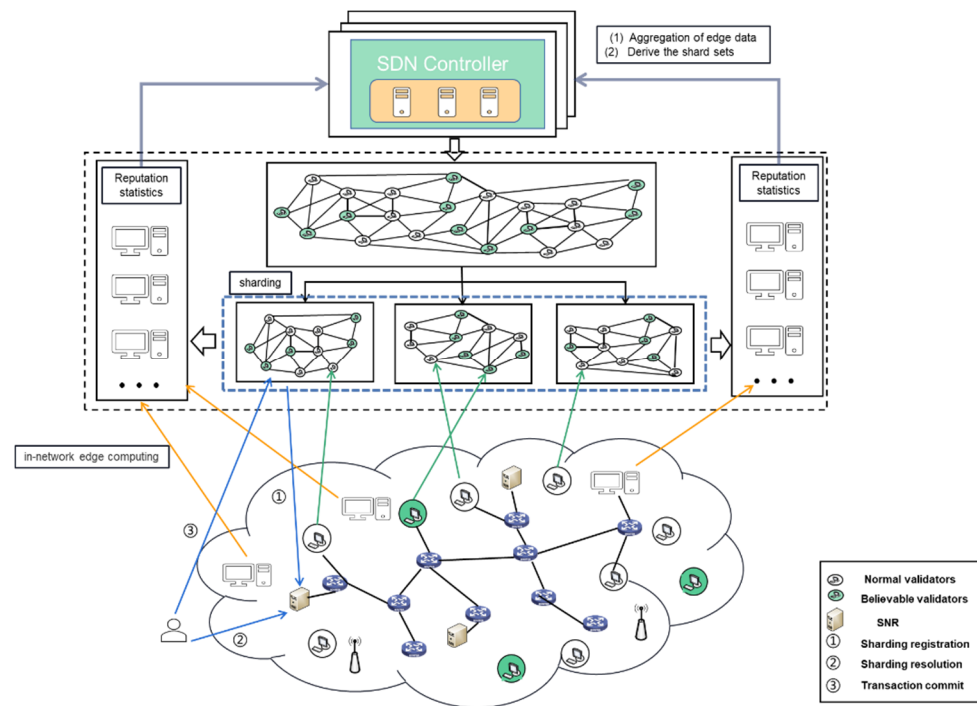


Figure 2. The system model.

Furthermore, with the gradual maturity of architecture design, ICN is envisioned as a promising candidate to support complex interoperability scenarios such as IoT, 5G, or MEC. Considering the practicability and scalability of the ICN scenarios, we implement system management by support with in-network edge computing and a Software Defined Network (SDN), which have been proposed to optimize resource allocation in ICN [42–48]. Meanwhile, the Standalone Name Resolution (SNR) [49–52] can be responsible for finding available shards in parallelizing transaction commitments by its registration and resolution functions.

As shown in Figure 2, the reputation data can be gathered, processed and validated by in-network edge computing, which monitors the behavior of consensus nodes to manage and calculate the reputation value. The aggregation of edge data can be realized by SDN technology, so as to derive the optimal shard sets that meet fair and secure shard distribution rules. The sharding registration, resolution and transaction commitments are shown as steps 1–3. Shard primary nodes in consensus stages register with their ID-NA in local SNR. When transaction commitments are submitted to client nodes, they request the network access (NA) of the primary node from the SNR to find the nearest serviceable shard sets. In the subsequent section, we discuss the reputation aggregation algorithm and sharding algorithm in detail. In order to better illustrate the security and rationality our scheme, some related technologies and definitions are introduced as follow.

- **Certificate:** our model defaults that there is a complete identity authentication scheme in the ICN system. That is, users can obtain identity certificates from public key infrastructure. Every consensus node obtains a legal identity certificate, which cannot be forged.
- **Data security:** Considering the security of data transmission and storage, cryptography (e.g., RSA and ECC) is utilized to encrypt and sign the security of the important information and messages. The security scheme data is not discussed in this paper. The transmission and storage of data cannot be tampered with by default.
- **Consensus Protocols:** consensus protocol used in intra-shard is PBFT (Practical Byzantine Fault Tolerance). Meanwhile, we reference the Believable-First approach proposed in paper [53]. The protocol divides all validators in an intra-shard into two groups, a believable league and a normal league. Believable validators complete the consen-

sus process of transactions quickly in the first phase. Afterwards, normal validators sample and verify the results in the second phase to provide supervision. The rules of a node being elected into the believable league are determined by the believability score, which is the reputation score value in our model.

- Cross-shard transactions: our research in this paper does not focus on the cross-shard transactions. The proposed sharding scheme can be applied to multiple cross-shard transactions approach in paper [54–57].

4. Proposed Scheme

4.1. Reputation Model

The sharding mechanism enhances the scalability of blockchain networks but also increases the influence of malicious attacks. In order to address this problem, researchers introduce the trust model of consensus nodes to improve the election process security. However, existing researches calculate the reputation value based on the node’s behavior in the block transaction, contributions to the community, reviews, etc. These dimensions are based on the subjective evaluation of other nodes in the blockchain without considering the network features of consensus nodes in the network layer. The existence of security risks such as collusion or monopoly makes these schemes unable to provide sufficient trust.

Therefore, we derive a multi-factor trust model for the believable validators selection process which extends trust parameters to quality of service (QoS), quality of performance (QoSP), evaluation reputation, past reputation and recommendation reputation. These evaluation factors measure reputation value from the perspective of objective trust, subjective trust and historical trust, which is shown in Figure 3. Our model ensures two important problems, that is, the accumulation of reputation value requires a certain period, and the high reputation value cannot be obtained by cheating. Any new node cannot significantly improve reputation value by performing well in a short time. Meanwhile, if a believable validator is detected as misbehaving, it loses all reputation value in the system and its identity is included in the blacklist. The cost of malicious behavior is to be powerfully expensive so that the validators have no incentive to misbehave under any circumstances.

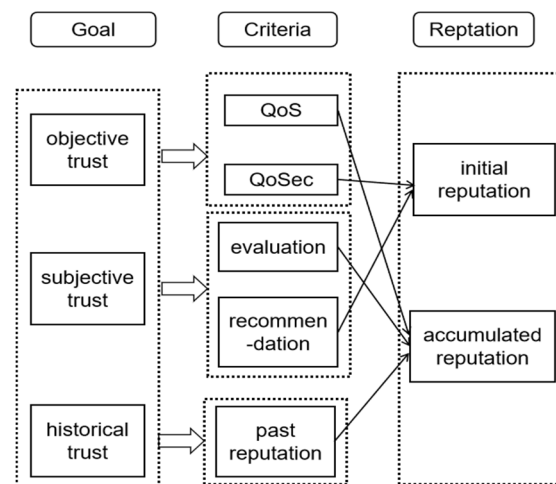


Figure 3. Hierarchy of reputation evaluation model.

We describe the trust parameters of our multi-factor trust model as initial reputation and accumulated reputation. The initial reputation of a newly added node is determined by QoSP reputation and recommendation reputation. The accumulated reputation is the aggregation of QoS reputation, evaluation reputation and past reputation. The quantitative scheme of reputation value is defined as follows. In order to calculate the aggregate value of multi-factors, we set the reputation value range of all factors to a dimensionless score from 1 to 10.

- **QoSP (Quality of security performance):** this parameter is a measure of the efficiency of consensus nodes in cryptographic calculations [58–60], such as signature, hash, or random operation. A node with strong computing power can provide better services for privacy and secure consensus process. An evaluation methodology for computing a quantitative QoSP metric is described as follows. Firstly, define the security parameters vector $E_i\{e_{i1}, e_{i2}, e_{i3}, \dots, e_{in}\}$ which can be expressed as security level. For instance, in our research, we are concerned about the asymmetric algorithm, encryption algorithm and hash algorithm. The vector $E_i\{e_{i1}, e_{i2}, e_{i3}\}$ represents the basic efficiency of security policy. e_{i1}, e_{i2}, e_{i3} can be described as the executions of algorithms per second. When new nodes access the network, they provide a set of performance parameters $E_x\{e_{x1}, e_{x2}, e_{x3}\}$ as vector E_i . The score $S_{xQoS P}$ of new nodes in our example is defined as the Equation (1)

$$S_{xQoSec} = S_{iQoSec} + S_{iQoSec} * \sum_{k=1}^n \frac{(e_{xk} - e_{ik})}{e_{ik}} \tag{1}$$

We convert this result into a score of 1 (low score) to 10 (high score) for the calculation of overall reputation value. The score $S_{iQoS P}$ of basic vector is 6. If the final calculated score exceeds the highest score 10 or is lower than the lowest score 1, the score is the highest score or the lowest score. The security policy’s audit process is proved by providing reports of some certified authority, such as Commercial Cryptography Testing Center of State Cryptography Administration (SCCTC). Nodes that do not provide this partial proof are assigned a lower initial value 1.

- **Recommendation reputation:** nodes can also seek recommendation from other peers with high reputation value, and the effectiveness of recommendation is directly proportional to the recommender reputation value. Similarly, the range of recommended reputation score value is set to 1–10. The recommended reputation can be either a good reputation recommendation or a report of a malicious node. If the system records that the node has found malicious behavior in the past, the node may be disqualified from joining the consensus group or get a very low initial score. The system can also provide high reputation recommendation value for nodes with good historical performance to enter the consensus group. The scheme to detect malicious nodes can refer to the papers [61–63].

Therefore, when nodes access the blockchain for the first time, the trust management system of access domain calculates an initial reputation value given in Equation (2).

$$R_0 = \begin{cases} \alpha S_{QoS P} + (1 - \alpha) \frac{1}{n} \sum_{j=1}^n w_j S_{recom} \\ a_0, \text{ when } S_{QoS P}, S_{recom} = 0 \end{cases} \tag{2}$$

$$w_j = \begin{cases} \frac{S_{recomer}}{S_{recom}}, \text{ when } S_{recom} > S_{recomer} \\ 1, \text{ when } S_{recom} < S_{recomer} \end{cases} \tag{3}$$

where R_0 is the initial reputation and $S_{QoS P}$ is the QoSP score. S_{recom} is the recommend reputation score value of the new node, $S_{recomer}$ is the recommender reputation score value and α is a tuning parameter of the weight. w_j is the weighting factors of the recommender. When the recommender gives a recommendation score higher than $S_{recomer}$, its credibility is the ratio of $S_{recomer}$ to S_{recom} . When the recommender gives a recommendation score lower than $S_{recomer}$, its reliability is 1. The recommended reliability of system records is 1. The Equation (3) ensures the fairness of recommendation reputation.

Besides, the accumulated reputation parameters are defined as follow:

- **Evaluation reputation:** it refers to the aggregation of reputation value obtained by nodes after participating in the consensus contribution and getting the evaluation of other peers. Our evaluation reputation does not only come from the consensus process of nodes. We also collect the evaluation information of non-consensus interaction

between nodes in the network. For the consensus reputation score, it can only be obtained and updated directly from the consensus process. The calculation information of the score gathers from the node validation logs on the prepare and commit. The total number of valid and invalid records in the consensus process can be obtained by gathering the log information of these nodes. Assuming there are n consensus nodes, each node saves the other n validation results into a $1 \times n$ vector. Based on the node validation result obtained from the peer-to-peer nodes, the $n \times n$ matrix can form an evaluation table quantitatively, which is shown in Figure 4.

$SC_{1,1}$	$SC_{1,2}$	\dots	$SC_{1,n}$
$SC_{2,1}$	$SC_{2,2}$		$SC_{2,n}$
\vdots	\vdots	\ddots	\vdots
$SC_{n,1}$	$SC_{n,2}$	\dots	$SC_{n,n}$

Figure 4. Node consensus trust evaluation.

$SC_{i,j}$ is the j th node consensus reputation score defined by Equation (4), $V_{i,j}$ and $NV_{i,j}$ are the total number of valid and invalid responses of the j th node (sending) with respect to the i th node (receiving) in the prepare and commit phase extracted from the log information. Then the consensus reputation score SC_{icon} is given by Equation (5), S_{total} is the upper limit of the node reputation value, $S_{total} = 10$, $SC_{i,i} = 0$.

$$SC_{i,j} = \frac{V_{i,j}}{V_{i,j} + NV_{i,j}} \times S_{total} \tag{4}$$

$$\frac{1}{n-1} \times \begin{bmatrix} SC_{1,1} & SC_{1,2} & \dots & SC_{1,n} \\ SC_{2,1} & SC_{2,2} & & SC_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ SC_{n,1} & SC_{n,2} & \dots & SC_{n,n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} SC_1 \\ SC_2 \\ \vdots \\ SC_n \end{bmatrix} \tag{5}$$

where $w_i = \frac{S_i}{S_{total}}$.

For the non-consensus evaluation reputation, the score value is obtained by the evaluation information of non-consensus interaction between nodes in the network. For instance, nodes can be publishers to participate in the resource contribution and get the evaluation of other subscribers. The minimum and maximum values of the SNC reputation value are between 1 and 10, respectively. The non-consensus score value is described as follow:

$$SNC_{incon} = \frac{n}{n+1} * T(i) * SNC_{inconh} + \frac{1}{n+1} * SNC_{inew} \tag{6}$$

where SNC_{incon} is the latest non-consensus evaluation reputation value. SNC_{inconh} is the historical non-consensus reputation value. n is the total number of historical interactions. SNC_{inew} is the latest non-consensus evaluation reputation score from peers. $T(i)$ is a time factor defined to measure the freshness of the reputation value. It is calculated according to Equation (7).

$$T(i) = e^{-\delta(t_i - t_h)} \tag{7}$$

where δ is a parameter to adjust the influence of time attenuation factor $T(i)$. The formation time of the previous reputation value SNC_{inconh} is t_h and the current time slot is t_i . The delay between them is used to measure the effectiveness of historical reputation, which

gradually weakens with the passage of time. The closer it is to the current time slot, the higher the weight of reputation value.

Therefore, evaluation reputation value is given by Equation (8); β is the weight coefficient:

$$Rup_{eval} = \beta * SC_{icon} + (1 - \beta) * SNC_{incon} \tag{8}$$

- Past reputation: this parameter measures the accumulated value of rewards or penalties received by nodes due to their contributions or malicious behaviors to the blockchain network in the past time, which is marked as S_{pr} .

For the contribution accumulation of a new node, several factors need to be considered. First, at the beginning of network access, the acquisition of positive reputation value contributes slowly to the growth of reputation value. This is to prevent some malicious nodes from improving reputation value through a small number of successful interactions. After a certain period, the change speed is accelerated. In the stable stage, the change speed tends to be steady. The design principle of the correction parameter $f(x)$ is given by Equation (9):

$$f(x) = \frac{K}{1 + \eta e^{-\mu x}}, \mu > 0, \eta > 0 \tag{9}$$

where K is the max value of past reputation score. μ and η measure the speed of curve change. Then, the S_{pr} is given by Equation (10):

$$S_{pr} = \frac{10 * \sum_{i=1}^n (f(n) * b)}{n * S_{prt}} \tag{10}$$

where b is the reward value obtained each time, S_{prt} is the upper limit value of reputation rewards obtained in the past and n is the number of rewards obtained.

- QoS: it is a parameter used to measure the service level of a consensus node based on the network features of nodes in the network layer. Researchers in paper [64–66] have laid a foundation for the QoS measuring methods in ICN. The performance characteristics are represented by a set of general parameters which can be latency, jitter, packet loss rate, effective caching, bandwidth, etc. QoS parameters are quantified by the aggregate value of each specific metric. Considering the compatibility of our model with multiple measuring methods, we uniformly convert the evaluation results of different methods into a “QoS score” ranging from 1 (Low Priority) to 10 (High Priority). It should be noted that we do not discuss the specific aggregation methods of QoS value in this paper. In the subsequent analysis, we use the score value S_{QoS} as a parameter to measure the QoS reputation of all nodes. S_{QoS} is set by Equation (11) as follow:

$$S_{QoS} = \frac{10}{QoS_{ul}} \left(\frac{1}{n} \sum_{i=1}^n QoS_{tk} \right) \tag{11}$$

where t_k is the time interval of QoS measurement, QoS is the measured value at time slot is t_k and n is the total number of QoS measurement. QoS_{ul} is the upper limit of the QoS value. We normalized the value of QoS into the S_{QoS} of our scoring criteria. The QoS reputation can also be applied to the node inactivation detection as the basic failure detection strategy.

The node trust S_i is computed by multiplying the multi-factor trust parameters and the trust weight, which is represented as follow:

$$S_i = \rho * R_0 + (1 - \rho) * (w_{rep} * Rup_{eval} + w_{pr} * S_{pr} + w_{QoS} * S_{QoS}) \tag{12}$$

where $\rho = T(j) = e^{-\tau(t_j - t_0)}$. w_{rep} , w_{pr} , and w_{QoS} is the weight coefficient of Rup_{eval} , S_{pr} and S_{QoS} . $T(j)$ is a time factor, defined to measure the access duration of consensus nodes. τ is a time attenuation factor $T(j)$. The access time of the consensus node is t_0 and the current time

slot is t_j . The delay between them is used to measure the effectiveness of initial reputation, which gradually weakens with the passage of time.

In addition, due to the high reliability of our reputation evaluation mechanism, in the view change in the consensus stage, we can directly select the node with high reputation value in the intra-shards to replace the primary node, so as to avoid the broadcast overhead caused by the view change in primary node inactivation. In the next section, we describe the sharding scheme of our model.

4.2. Sharding Scheme

The proposed reputation-based sharding scheme’s objective in assigning nodes to shards is to find an optimal shard distribution set according to the reputation value of nodes and network distance. Let N be the total number of consensus nodes satisfying $N = \sum_{i=1}^K N_k$, where N_k represents the number of nodes in the k th shard. S_{ki} is the reputation score of the node i in the k th shard. AT_k represents the aggregated trust of the k th shard set, $AT_k = \sum_{i=1}^{N_k} S_{ki}$. r_k is the total number of believable validators in the k th shard whose reputation score exceeds threshold S_{th} . D_{ij} is the network distance of node i and node j . The problem statement of our model is described as follow:

- (1) The aggregate trust difference for each shard should be less than threshold θ :

$$\sum_{i=1}^K \sum_{j=1, j \neq i}^K \frac{|AT_i - AT_j|}{C_K^2} = \sum_{i=1}^K \sum_{j=1, j \neq i}^K \frac{2 * |AT_i - AT_j|}{K(K - 1)} \leq \theta \tag{13}$$

- (2) The aggregated reputation value of believable validators and normal validators for each shard should be more than threshold Φ_1 and Φ_2 :

$$\sum_{i=1}^{r_k} S_{ki} \geq \Phi_1, \sum_{j=1}^{N_k - r_k} S_{kj} \geq \Phi_2 \tag{14}$$

- (3) All the shard clusters are evenly distributed in the network instead of gathering in a domain;
- (4) Nodes with the same consensus process in each shard should be separated into different shards in the next epoch, which is to ensure the randomness of shards and avoid collusion attack.
- (5) To optimize a shard algorithm, aim to minimize network diameter within the shard. The smaller the network diameter, the shorter the average broadcast time between any two nodes, thus effectively reducing the broadcast latency.

These properties are applied to the proposed sharding modified scheme. We implement the sharding method to organize the nodes across the network into several clusters based on the well-known AP (Affinity Propagation) algorithm [67]. For most existing algorithms for data clustering, the number of clusters must be predetermined before running the clustering procedure. However, predetermining shard groups limits the acquisition of the optimal solution in sharding detection. Therefore, the AP algorithm realizes a method to determine the optimal number of intra-shards based on reputation value and network distance. Meanwhile, researchers have shown that the square-error of the clustering result and efficiency of algorithm are superior to the traditional K-Means algorithm, C-means algorithm, etc.

Affinity propagation takes as input a collection of similarities between data points, where the similarity $s(i, k)$ indicates how well the data point with index k is suited to be the exemplar for data point i [67]. We optimize the calculation method of similarity by cosine similarity. The cosine similarity $Sim_{i,k}$ of two data vectors of node i and k is expressed as Equation (15):

$$Sim_{i,k} = \frac{\vec{x}_i \bullet \vec{x}_k}{|\vec{x}_i| \times |\vec{x}_k|} \tag{15}$$

where $\vec{x}_i = (S_i, D_{i,k})$, $\vec{x}_k = (S_k, D_{k,i})$, and $D_{i,k}$ the network distance from node i to node k , which is equal to the network latency. It should be noted that the two network latencies $D_{i,k}$ and $D_{k,i}$ are not necessarily equal. The description of the algorithm is exhibited as follows (Algorithm 1):

Algorithm 1. AP (Affinity Propagation) algorithm

Input:

$X = \{x_1, x_2, \dots, x_n\}$: The cosine similarity.

p : The optimal preference.

λ : Damping factor.

Output:

$[C_1, C_2 \dots, C_m]$: The sharding clustering labels.

$[\text{Cluster}(x_1), \dots, \text{Cluster}(x_n)]$: $\text{Cluster}(x_i)$ represents which cluster(group) x_i belongs to.

1: Calculate the similarity $s(i, k)$ between data point x_i and x_k ($i, k = 1, 2, \dots, n$). All the similarities form the similarity matrix $S = [s(i, k)]_{n \times n}$

2: $p \leftarrow$ preference according to a priori convention.

3: Initial the availability matrix $A = [a(i, k)]_{n \times n}$ as a zero matrix: $a(i, k) = 0$.

4: **repeat** 5,6,7,8,9 **until** R and A converge and $\sum_{i=1}^K \sum_{j=1, j \neq i}^K \frac{2 * |AT_i - AT_j|}{K(K-1)} \leq \theta$

5: Update the matrix $R = [r(i, k)]_{n \times n}$ using the rule follow:

$$r(i, k) = s(x_i, x_k) - \max_{j:j \neq k} [s(x_i, x_j) + a(x_i, x_j)]$$

6: Update the matrix $A = [a(i, k)]_{n \times n}$ using the rule follow:

$$a(i, k) = \begin{cases} \sum_{i':i' \neq k} \max(0, r(x_{i'}, x_k)), & \text{for } k=i \\ \min(0, r(x_i, x_k), \sum_{i':i' \neq k} \max(0, r(x_{i'}, x_k))), & \text{for } k \neq i \end{cases}$$

7: $R_i \leftarrow (1 - \lambda) \times R_i + \lambda \times R_{i-1}$

8: $A_i \leftarrow (1 - \lambda) \times A_i + \lambda \times A_{i-1}$

9: Calculate the aggregate trust using rule: $AT_k = \sum_{i=1}^{N_k} S_{ki}$

10: Calculate the exemplar labels $C = [C_1, C_2, \dots, C_n]$ using rule: $C_i = \underset{x_k}{\text{argmax}} [a(i, k) + r(i, k)]$

11: Calculate the cluster labels using rule: $\text{Cluster}(x_i) = \text{Cluster}(C_i)$

After the affinity propagation, the exemplar labels C_i represent the sharding labels of consensus nodes. The believable league is selected by v_1 validators whose reputation value is over tr_{th} and the aggregate trust of each believable league meets the Equation (14). The normal league consists of v_2 nodes which are randomly selected from the remaining validators in the shard. Meanwhile, the aggregate trust of each normal league should be more than threshold Φ_2 shown as Equation (14). Then, the possible combinations of leagues are shown as Equations (16) and (17):

$$l_1 = C_r^{v_1}, S_{v_1} \geq tr_{th}, \sum_{i=1}^{v_1} S_{ki} \geq \Phi_1 \tag{16}$$

$$l_2 = C_{N_k-r}^{v_2}, \sum_{j=1}^{N_k-r} S_{kj} \geq \Phi_2 \tag{17}$$

$$L = l_1 \times l_2 \tag{18}$$

where l_1 and l_2 are the total numbers of believable league and normal league. L is the total number of combinations which can complete transaction verification. We select an alternative randomly within an election period, and the alternative can be replaced in the next election period of one epoch. At the same time, in order to increase randomness and fairness, the nodes that complete the consensus in each round reduce a random number from the existing reputation value. This does not mean that the reputation value of the node is really reduced, but rather the transformation in the calculation of clustering parameters, which is similar to the Servi mechanism of PoB [53].

Compared with a fixed number of validators, our approach is more flexible and random in the selection of validation groups. It provides a possible solution that the epoch of shards can be prolonged without increasing the security risk, even in days. In the next

section, we prove the randomness of the scheme is bias resistant and give the security analysis of the whole scheme.

5. Security Analysis

In this section, we first give the most concerned bias-resistant proof in the sharding scheme, and then give a security analysis of the reputation evaluation algorithm on which the sharding algorithm depends.

5.1. Bias-Resistant Proof

For proving the bias resistance of our model, we introduce the proof that the corresponding shard distribution problem is NP-hard. Therefore, there is no exact algorithm that is guaranteed to find the optimal solution within polynomial time. The proof process is as follows:

Proof. The shard distribution problem (denoted as D) is to find k disjoint and non-empty shard set G_i , which satisfies the condition as Equations (13) and (14). $G = G_1 \cup G_2 \cup \dots \cup G_k$. The value of θ is infinitely close to zero. In this case, the core problem of D is reduced to a partition problem finding, that is, a group of numerical values are divided into k mutually disjoint groups according to the constraint Equations (13) and (14), so as to minimize the difference between subsets. The result of grouping ensures that the aggregate reputation trust difference of each partition is as small as possible. According to the definition of the K -partition problem, it can be seen that this is a NP complete problem [68]. On the other hand, the D problem is an optimization problem to find the optimal set G . For a given arbitrary solution set G^* , there is no polynomial time algorithm to verify that the given G^* can obtain the best set G satisfying Equations (13) and (14). Therefore, the D problem is not the NP problem. Since the problem $D \notin NP$, it is NP-hard not in NP-complete (NPC). \square

5.2. Security Analysis of the Reputation Algorithm

The multi-factor reputation model extends the evaluation dimension of reputation value. The QoS and QoS parameters are the reputation obtained by the network measurement information and trusted third party certification. Any consensus node cannot cheat on these two parts of reputation by hiring larger ghostwriters. Meanwhile, we design the time attenuation factor $T(i)$ and correction function $f(x)$ to prevent some malicious nodes from improving reputation value through a small number of successful interactions. The accumulation of high reputation value requires a certain period and cannot be obtained by cheating. Any new node cannot significantly improve reputation value by performing well in a short time. We also prove this in the experiment in the next section. Meanwhile, the cost of malicious behavior is powerfully expensive so that the validator has no incentive to misbehave under any circumstances. Therefore, when the reputation value of a consensus node reaches the trust threshold S_{th} , it is almost impossible to conduct malicious behavior.

6. Performance Evaluation

We built our experiments based on the OMNET++ simulation platform for evaluating the performance of our model. The experimental environment of simulation is configured as following: Intel i7-4790 CPU @ 3.60ghz (8 CPU cores), memory 4096 MB, system model Dell OptiPlex 9020. In this paper, the simulation of network topology is built by importing different scenarios and topology types from INET supported by OMNET. Communication link delay is set as 100 ms and network bandwidth is 20 Mbps, which is consistent with the measurement values of bitcoin and Ethereum [69]. The total number N of nodes in the blockchain network is 1000. The experiment result is tested 5 times and each test included 100 rounds of consensus.

The minimum and maximum values of the node's real reputation value is set to 1 and 10, respectively. The variation range of initial reputation value is set between 3 and 5. It is measured by whether a node provides recommendation reputation. The initial reputation

of a node that does not provide recommendation reputation is 3. Qos reputation is counted by the the packet loss rate, latency and jitter, which is recorded in the log information of the consensus node. The simulation parameters used in the proposed scheme are presented in Table 2.

Table 2. Simulation setup.

Simulation Parameters	Value
Simulation tool	OMNeT++
Number of consensus nodes	1000
Communication link delay	100 ms
Network bandwidth	20 Mbps
w_{rep} , w_{pr} , and w_{Qos}	1/3
τ , δ	1
damping factor λ	0.5
α , β	0.5
θ	0.01

In the consensus commitment and preparation stage of PBFT, the packet loss rate is set to a random value from 0.05 to 0.1. We set the false response probability (FRP) as an indicator to simulate the robustness of response message. The parameter represents the probability that the commitment or preparation message of a consensus node is corrupted. The FRP is set by Equation (19), which includes two conditions:

1. An error message from consensus nodes should be rejected by the model.
2. A false positive message indicates that normal feedback is incorrectly detected as malicious feedback by the model

$$FRP = \frac{1}{1 + S_i} \quad (19)$$

The non-consensus evaluation reputation value is obtained by adding the real reputation value and a Gaussian noise with zero mean value. The variance of the noise is a dynamic variable $k\sigma$ in the experiment, where k is the scale factor of the noise variance and σ is the variance unit of noise. In the experiment, the value of K is 1, 2 or 3, which can test the influence of different feedback noise on the reputation model. Since reputation score feedback is a subjective uncertainty value, unit noise variance σ is set to 1, which indicates a relatively large noise [70].

$$SNC_{new} = R_{real} + \frac{1}{\sqrt{2\pi k\sigma}} \exp\left(-\frac{x^2}{2k\sigma}\right) \quad (20)$$

The correction parameter of past reputation is set as fitting curves $f_1(x)$, which is shown as Equation (9). The fitting curve is calculated by the following conditions and the image of the curve is shown in Figure 5

1. The curve tends to be stable at the 10 times interaction;
2. The coefficient of initial accumulation value is 0.1;
3. The coefficient of the fifth interaction is 0.5.

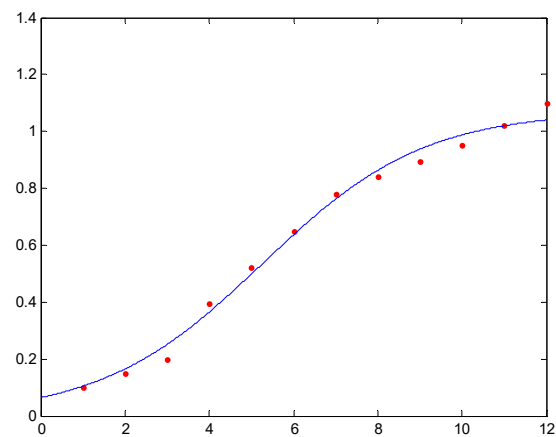


Figure 5. The correction parameter curve of past reputation.

At the beginning of the experiment, we regard all nodes as new nodes and set the reputation value of nodes to 4 for clustering. In total, 1000 nodes are randomly generated in the range of $10^4 \times 10^4$ (m²). In order to facilitate the visualization of clustering results, in our simulation settings, the network coordinate distance is directly proportional to the network latency. However, in real-world scenarios, the network latency of nodes close to each other is not necessarily lower. Therefore, in the similarity calculation, we measure the network latency as a distance parameter rather than the real geographical distance. The damping factor λ is 0.5. The clustering results in Figure 6 show that the number of clustering groups is 28, and the number of nodes in each group is between 30–40. Then, we build a PBFT consensus simulation through the OMNeT++ platform to simulate the consensus process in each shard and analyze the evaluation reputation through additional packets between nodes. The change in reputation value is recorded every two rounds.

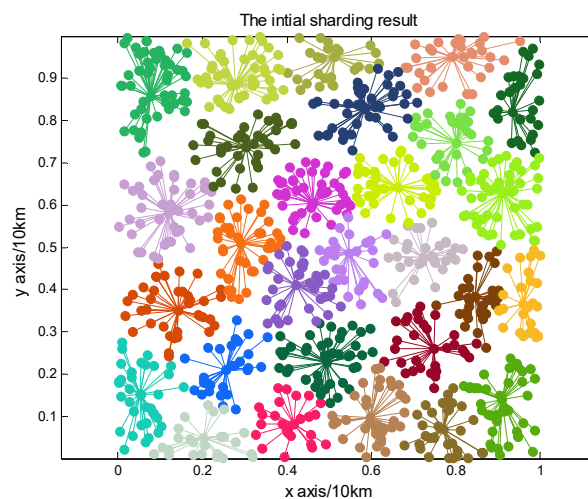


Figure 6. The initial clustering results.

Figure 7 validates the average convergence time of our reputation evaluation mechanism. We select three groups of nodes with real reputation values of 5, 7 and 9 as representatives for comparison. As shown in Figure 7, the higher reputation value requires longer convergence time, which also proves our view in Section 3 that high reputation value can not be obtained by cheating in a short time. Figure 8 shows the comparison of the convergence time of reputation value from 1 to 10. We can see that for malicious nodes with reputation value less than 4, the reputation algorithm can be found within 6 rounds, while for trusted nodes with reputation value greater than 6, at least 15 rounds of reputation accumulation are required. Meanwhile, we also compared the reputation

value statistical algorithm proposed in [32,38]. The comparison parameter is through the normalized mean square error (NMSE), which is defined by Equation (21). Figure 9 shows the evaluation results of the three trust algorithms. It can be seen that our reputation algorithm is obviously closer to the actual reputation value and the evaluation effect is better than [32,38].

$$NMSE = \frac{1}{x_{real}} \sqrt{\frac{\sum_{i=1}^m (x_i - x_{real})^2}{m}} \tag{21}$$

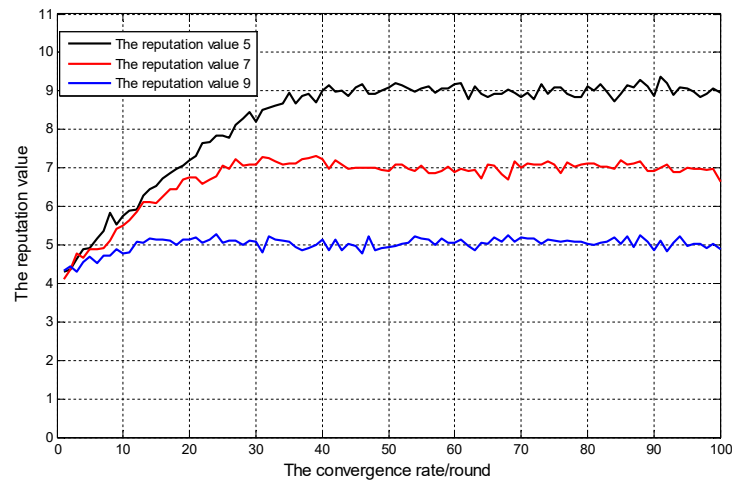


Figure 7. The average convergence speed with real reputation values of 5, 7 and 9.

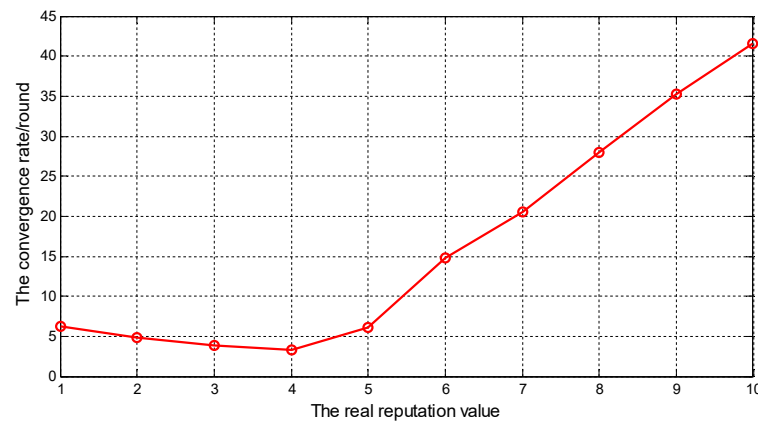


Figure 8. The comparison of the convergence time of reputation value from 1 to 10.

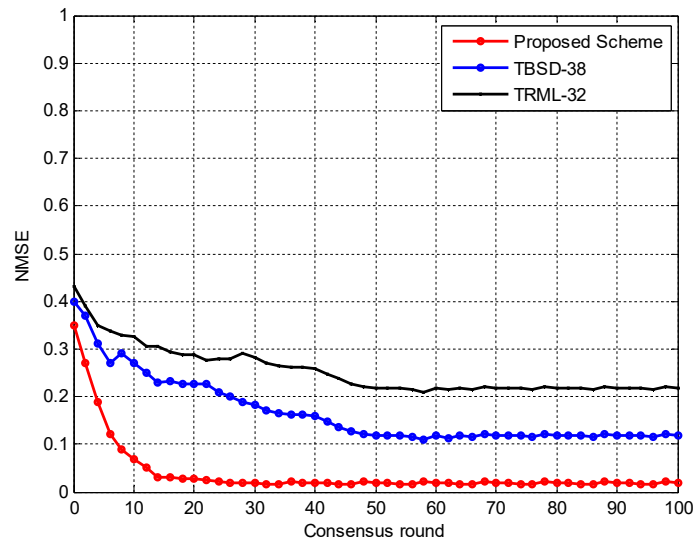


Figure 9. The NMSE results of the three trust algorithms.

The number of transactions packaged in a block is set as 2000 [69]. T_s indicates the time point of the block arriving at the intra-shard, and T_e represents the time point at which the PBFT consensus completion. Then, the TPS may be measured as the transaction throughput as Equation (22)

$$TPS = \frac{1}{T_e - T_s} \times 2000 \tag{22}$$

We compare the proposed model with a classical sharding scheme Omniledger [18]. The experimental results are shown in Figure 10. It can be seen from the experimental results that due to the addition of network latency parameters, the broadcast latency in intra-shard is significantly reduced, which significantly improves the transaction throughput and proves the advantages of our model. The clustering results after 100 rounds of consensus are shown in Figure 11.

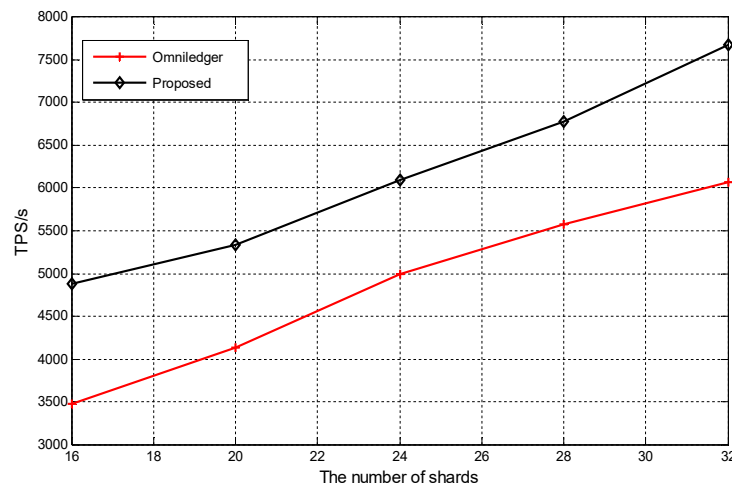


Figure 10. The experimental results of TPS.

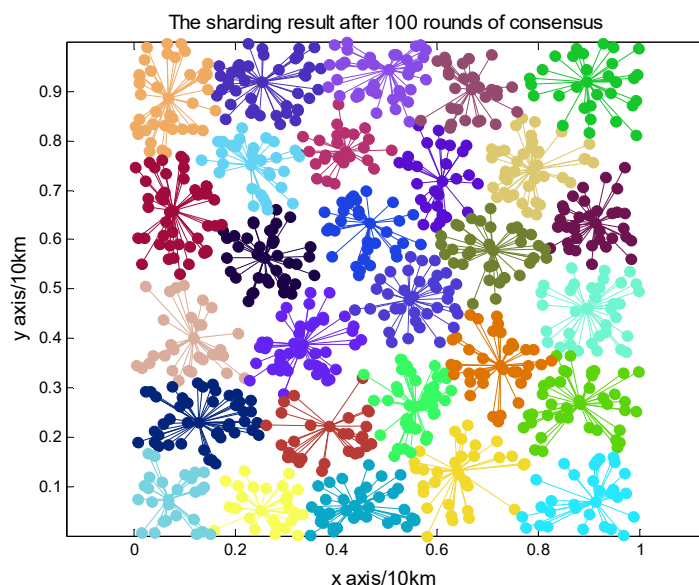


Figure 11. The clustering results after 100 rounds of consensus.

7. Conclusions

In this paper, we proposed a reputation-based sharding blockchain consensus model in ICN. Our model focused on the key challenges in the existing sharding mechanism, which are reliability and efficiency. Most the researchers devote their attention to the optimization of cross-shard transactions or fairness shard, while lacking consideration of the potential sharp increases of malicious attacks in intra-shards and the influence of underlying network parameters on consensus efficiency. To address these challenges, we optimized a multi-dimension reputation model and leveraged the AP algorithm for gathering consensus nodes into shards. We prove the robustness of the multi-dimension reputation model since the high reputation value cannot be obtained by cheating or bribery attack. Given the minimal possibility to be at fault in the security of validation, the model achieves real-time processing speed with assurance. The experimental results demonstrate that in comparison with existing classical sharding schemes, our model exhibits better security and efficiency.

As future work, we intend to more deeply research the integration of ICN and blockchain. Using the characteristics of the ICN, the blockchain can sink various technologies into the network. The performance of the blockchain in communication and storage can be optimized by the ICN approach. Therefore, the efficiency and availability of the sharding blockchain will be further improved to better serve the IoT, AR/VR, or 5G/6G scenarios.

Author Contributions: Conceptualization, J.S. and Y.L.; Data curation, J.S.; Formal analysis, J.S.; Funding acquisition, Y.L. and X.Z.; Investigation, J.S.; Methodology, J.S., Y.L. and X.Z.; Software, J.S.; Supervision, X.Z.; Validation, J.S.; Writing—original draft, J.S.; Writing—review and editing, Y.L. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Strategic Leadership Project of Chinese Academy of Sciences: SEANET Technology Standardization Research System Development (Project No. XDC02070100).

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

References

1. Wu, H.; Dwivedi, A.D.; Srivastava, G. Security and Privacy of Patient Information in Medical Systems Based on Blockchain Technology. *ACM Trans. Multimedia Comput. Commun. Appl.* **2021**, *17*, 1–17. [[CrossRef](#)]
2. Dwivedi, A.D.; Singh, R.; Kaushik, K.; Mukkamala, R.R.; Alnumay, W.S. Blockchain and artificial intelligence for 5G-enabled Internet of Things: Challenges, opportunities, and solutions. *Trans. Emerg. Telecommun. Technol.* **2021**, e4329. [[CrossRef](#)]

3. Yang, R.; Yu, F.R.; Si, P.; Yang, Z.; Zhang, Y. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Commun. Surveys Tuts.* **2019**, *21*, 1508–1532. [[CrossRef](#)]
4. Awais, M.; Shah, M.A. Information-centric networking: A review on futuristic networks. In Proceedings of the 2017 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK, 7–8 September 2017. [[CrossRef](#)]
5. Liao, S.; Wu, J.; Li, J.; Konstantin, K. Information-Centric Massive IoT based Ubiquitous Connected VR/AR in 6G: A Proposed Caching Consensus Approach. *IEEE Internet Things J.* **2020**, *8*, 5172–5184. [[CrossRef](#)]
6. Chen, C.; Wang, C.; Qiu, T.; Lv, N.; Pei, Q. A Secure Content Sharing Scheme Based on Blockchain in Vehicular Named Data Networks. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3278–3289. [[CrossRef](#)]
7. Sharma, V.; You, I.; Jayakody, D.N.K.; Reina, D.G.; Choo, K.-K.R. Neural-Blockchain-Based Ultrareliable Caching for Edge-Enabled UAV Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5723–5736. [[CrossRef](#)]
8. Pan, Q.; Wu, J.; Li, J.; Yang, W.; Guan, Z. Blockchain and AI Empowered Trust-Information-Centric Network for Beyond 5G. *IEEE Netw.* **2020**, *34*, 38–45. [[CrossRef](#)]
9. Lai, K.; Zhang, Q.; Lou, J.; Bai, B.; Xu, K. Securing ICN-Based UAV Ad Hoc Networking with Blockchain. *IEEE Commun. Mag.* **2019**, *57*, 26–32. [[CrossRef](#)]
10. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 23 April 2013).
11. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [[CrossRef](#)]
12. Poon, J.; Buterin, V. Plasma: Scalable autonomous smart contracts. In *Lightning Netw*; Tech. Rep. Working Draft; Ethereum: San Francisco, CA, USA, 2017; pp. 1–47.
13. Jourenko, M.; Kurazumi, K.; Lorangeira, M.; Tanaka, K. SoK: A taxonomy for layer-2 scalability related protocols for cryptocurrencies. *IACR Cryptol. ePrint Arch.* **2019**, *2019*, 352.
14. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* **2019**, *7*, 22328–22370. [[CrossRef](#)]
15. Yu, G.; Wang, X.; Yu, K.; Ni, W.; Zhang, J.A.; Liu, R.P. Survey: Sharding in Blockchains. *IEEE Access* **2020**, *8*, 14155–14181. [[CrossRef](#)]
16. Manshaei, M.H.; Jadliwala, M.; Maiti, A.; Fooladgar, M. A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains. *IEEE Access* **2018**, *6*, 78100–78112. [[CrossRef](#)]
17. 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, Vienna, Austria, 24–28 October 2016; pp. 17–30. [[CrossRef](#)]
18. Kokoris-Kogias, E.; Jovanovic, P.; Gasser, L.; Gailly, N.; Syta, E.; Ford, B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 583–598. [[CrossRef](#)]
19. Zamani, M.; Movahedi, M.; Raykova, M. Rapidchain: Scaling blockchain via full sharding. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 931–948.
20. The ZILLIQA Team. The Zilliqa Technical Whitepaper. Available online: <https://docs.zilliqa.com/whitepaper.pdf> (accessed on 20 August 2017).
21. Gilad, Y.; Hemo, R.; Micali, S.; Vlachos, G.; Zeldovich, N. Algorand: Scaling byzantine agreements for cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, 28 October 2017. [[CrossRef](#)]
22. Feldman, P. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), Los Angeles, CA, USA, 12–14 October 1987; pp. 427–438. [[CrossRef](#)]
23. Stadler, M. Publicly verifiable secret sharing. In *Advances in Cryptology—EUROCRYPT*; Maurer, U., Ed.; Springer: Berlin/Heidelberg, Germany, 1996; pp. 190–199.
24. Syta, E.; Jovanovic, P.; Kogias, E.K.; Gailly, N.; Gasser, L.; Khoffi, I.; Fischer, M.J.; Ford, B. Scalable Bias-Resistant Distributed Randomness. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 444–460. [[CrossRef](#)]
25. Wesolowski, B. Efficient verifiable delay functions. In Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, 19–23 May 2019; Volume 11478, pp. 379–407.
26. Ometoruwa, T. Solving the Blockchain Trilemma: Decentralization Security & Scalability. Available online: <https://www.coinbureau.com/analysis/solving-blockchain-trilemma/> (accessed on 10 October 2021).
27. Wang, J.; Wang, H. Monoxide: Scale out blockchains with asynchronous consensus zones. In Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Boston, MA, USA, 26 February 2019; pp. 95–112.
28. Al-Bassam, M.; Sonnino, A.; Bano, S.; Hryczyszyn, D.; Danezis, G. Chainspace: A sharded smart contracts platform. *arXiv* **2017**, arXiv:1708.03778.
29. Buterin, V. Ethereum Sharding FAQ. Available online: <https://github.com/ethereum/wiki/wiki/ShardingFAQ> (accessed on 1 August 2019).
30. McCorry, P.; Hicks, A.; Meiklejohn, S. *Smart Contracts for Bribing Miners*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–18.

31. Shawn, D. 1% Shard Attack Explained. Available online: <https://www.mangoresearch.co/1-shard-attack-explained-ethereum-sharding-contd/> (accessed on 11 March 2018).
32. Halgamuge, M.N.; Hettikankanamge, S.C.; Mohammad, A. Trust model to minimize the influence of malicious attacks in sharding based blockchain networks. In Proceedings of the IEEE International Conference on Artificial Intelligence and Knowledge Engineering, Laguna Hills, CA, USA, 9–13 December 2020; pp. 162–167.
33. Wang, G. RepShard: Reputation-based Sharding Scheme Achieves Linearly Scaling Efficiency and Security Simultaneously. In Proceedings of the IEEE International Conference on Blockchain, Rhodes, Greece, 2–6 November 2020; pp. 237–246.
34. Kang, J.; Xiong, Z.; Niyato, D.; Ye, D.; Kim, D.I.; Zhao, J. Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2906–2920. [[CrossRef](#)]
35. Yang, Z.; Yang, K.; Lei, L.; Zheng, K.; Leung, V.C.M. Blockchainbased decentralized trust management in vehicular networks. *IEEE Internet Things J.* **2018**, *6*, 1495–1505. [[CrossRef](#)]
36. Bugday, A.; Ozsoy, A.; Öztaner, S.M.; Sever, H. Creating consensus group using online learning based reputation in blockchain networks. *Pervasive Mob. Comput.* **2019**, *59*, 111–125. [[CrossRef](#)]
37. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q.; Lim, L.; Yang, M.-H. Hedged deep tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4303–4311.
38. Yun, J.; Goh, Y.; Chung, J.-M. Trust-Based Shard Distribution Scheme for Fault-Tolerant Shard Blockchain Networks. *IEEE Access* **2019**, *7*, 135164–135175. [[CrossRef](#)]
39. Hao, W.; Zeng, J.; Dai, X.; Xiao, J.; Hua, Q.-S.; Chen, H.; Li, K.-C.; Jin, H. Towards a Trust-Enhanced Blockchain P2P Topology for Enabling Fast and Reliable Broadcast. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 904–917. [[CrossRef](#)]
40. Jia, Z.; Chen, R.; Li, J. Delottery: A novel decentralized lottery system based on blockchain technology. In Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications, Xi'an, China, 9–11 December 2019; pp. 20–25.
41. Boneh, D.; Bonneau, J.; Bünz, B.; Fisch, B. Verifiable delay functions. In *Advances in Cryptology—CRYPTO*; Shacham, H., Boldyreva, A., Eds.; Springer: Cham, Switzerland, 2018; pp. 757–788.
42. Zhang, Q.-Y.; Wang, X.-W.; Huang, M.; Li, K.-Q.; Das, S. Software Defined Networking Meets Information Centric Networking: A Survey. *IEEE Access* **2018**, *6*, 39547–39563. [[CrossRef](#)]
43. Chang, D.; Kwak, M.; Choi, N.; Kwon, T.; Choi, Y. C-flow: An efficient content delivery framework with OpenFlow. In Proceedings of the International Conference on Information Networking 2014 (ICOIN2014), Phuket, Thailand, 10–12 February 2014; pp. 270–275. [[CrossRef](#)]
44. Vahlenkamp, M.; Schneider, F.; Kutscher, D.; Seedorf, J. Enabling ICN in IP networks using SDN. In Proceedings of the Enabling ICN in IP Networks Using SDN, Goettingen, Germany, 7–10 October 2013; pp. 1–2. [[CrossRef](#)]
45. Trajano, A.F.R.; Fernandez, M.P. ContentSDN: A Content-Based Transparent Proxy Architecture in Software-Defined Networking. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 532–539. [[CrossRef](#)]
46. Fan, Z.; Yang, W.; Tian, K. An edge computing service model based on information-centric networking. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 498–505.
47. Mastorakis, S.; Mtibaa, A.; Lee, J.; Misra, S. ICedge: When Edge Computing Meets Information-Centric Networking. *IEEE Internet Things J.* **2020**, *7*, 4203–4217. [[CrossRef](#)]
48. Sarros, C.-A.; Lertsinsrubtavee, A.; Molina-Jimenez, C.; Prasopoulos, K.; Diamantopoulos, S.; Vardalis, D.; Sathiaseelan, A. ICN-based edge service deployment in challenged networks. In Proceedings of the 4th ACM Conference on Information-Centric Networking, Berlin, Germany, 26–28 September 2017; pp. 210–211.
49. Koponen, T.; Chawla, M.; Chun, B.-G.; Ermolinskiy, A.; Kim, K.H.; Shenker, S.; Stoica, I. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 181–192. [[CrossRef](#)]
50. Liao, Y.; Sheng, Y.; Wang, J. A deterministic latency name resolution framework using network partitioning for 5G-ICN integration. *International Journal of Innovative Computing. Inf. Control.* **2019**, *15*, 1865–1880.
51. Wang, J.; Chen, G.; You, J.; Sun, P. SEANet: Architecture and Technologies of an On-site, Elastic, Autonomous Network. *J. Netw. New Media Tech.* **2020**, *9*, 1–8.
52. Dannewitz, C.; Kutscher, D.; Ohlman, B.; Farrell, S.; Ahlgren, B.; Karl, H. Network of Information (NetInf)—An information-centric networking architecture. *Comput. Commun.* **2013**, *36*, 721–735. [[CrossRef](#)]
53. The Cryptocurrency Technical Whitepaper. Available online: https://cryptorating.eu/whitepapers/IOST/Tech_white_paper_EN (accessed on 12 October 2021).
54. Qi, Z.; Zhang, Y.; Wang, Y.; Wang, J.; Wu, Y. A Cascade Structure for Blockchain. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 252–253.
55. Yu, S.; Lv, K.; Shao, Z.; Guo, Y.; Zou, J.; Zhang, B. A High Performance Blockchain Platform for Intelligent Devices. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 260–261.
56. Yoo, H.; Yim, J.; Kim, S. The Blockchain for Domain Based Static Sharding. In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering (Trustcom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 1689–1692.

57. Zou, J.; Dong, Z.; Shao, A.; Zhuang, P.; Li, W.; Zomaya, A.Y. 3DDAG: A High-Performance DAG Network with Eventual Consistency and Finality. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 262–263.
58. Fadlullah, Z.M.; Wei, C.; Shi, Z.; Kato, N. GT-QoSec: A game-theoretic joint optimization of QoS and security for differentiated services in next generation heterogeneous networks. *IEEE Trans. Wireless Commun.* **2017**, *16*, 1037–1050. [[CrossRef](#)]
59. Luna, J.; Flouris, M.; Marazakis, M.; Bilas, A. Providing security to the Desktop Data Grid. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing, Miami, FL, USA, 14–18 April 2008; pp. 2779–2786.
60. O Gundoyin, S.O.; Kamil, I.A. A Fuzzy-AHP based prioritization of trust criteria in fog computing services. *Appl. Soft. Comput.* **2020**, *97*, 106789. [[CrossRef](#)]
61. Almaiah, M.A. A New Scheme for Detecting Malicious Attacks in Wireless Sensor Networks Based on Blockchain Technology. *Artif. Intell. Blockchain Future Cybersecur. Appl.* **2021**, 217–234. [[CrossRef](#)]
62. Adil, M.; Almaiah, M.A.; Alsayed, A.O.; Almomani, O. An Anonymous Channel Categorization Scheme of Edge Nodes to Detect Jamming Attacks in Wireless Sensor Networks. *Sensors* **2020**, *20*, 2311. [[CrossRef](#)]
63. Ali, A.; Almaiah, M.A.; Hajje, F.; Pasha, M.F.; Fang, O.H.; Khan, R.; Teo, J.; Zakarya, M. An Industrial IoT-Based Blockchain-Enabled Secure Searchable Encryption Approach for Healthcare Systems Using Neural Network. *Sensors* **2022**, *22*, 572. [[CrossRef](#)]
64. Kerrouche, A.; Senouci, M.R.; Mellouk, A. QoS-FS: A new forwarding strategy with QoS for routing in Named Data Networking. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–7. [[CrossRef](#)]
65. McCarthy, J.; Chaudhry, S.R.; Kuppuudaiyar, P.; Loomba, R.; Clarke, S. QoS-ICN: An information-centric approach to QoS in vehicular environments. *Veh. Commun.* **2021**, *30*, 100351. [[CrossRef](#)]
66. McCarthy, J.; Kuppuudaiyar, P.; Loomba, R.; Clarke, S. Towards an ICN Approach to QoS in a Dynamic Edge Environment. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [[CrossRef](#)]
67. Frey, B.J. Dueck Delbert Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [[CrossRef](#)] [[PubMed](#)]
68. Korf, R.E. A complete anytime algorithm for number partitioning. *Artif. Intell.* **1998**, *106*, 181–203. [[CrossRef](#)]
69. Ethereum Sharding FAQ. Available online: <https://github.com/Ethereum/wiki/wiki/Sharding-FAQs> (accessed on 10 August 2021).
70. Maybeck, P.S. Stochastic models, estimation, and control. *Math. Sci. Eng.* **1979**, *141*, 1–423.