EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Trust Model for Effective Consensus in Blockchain

R. Shalini[1,*] and R. Manoharan[2]

[1]Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, INDIA.
[2]Professor in Computer Science and Engineering, Pondicherry Engineering College, Puducherry, INDIA.

Abstract

Blockchain technology is a revolution started as a new economy with an alternative currency namely Bitcoin. Besides the economical aspect, the technological capabilities of Blockchain such as distributed computing, record keeping, irrecoverability of transactions, reliability and etc., are harnessed by variety of real-world applications. Blockchain is a rising pool of records known as blocks linked using security procedure. It is typically managed by a group of nodes in a distributed network technology which integrates technologies such as distributed ledger, security and consensus algorithm to ensure reliability and immutability. In Blockchain, the access privileges are determined by a set of nodes called miners, which run the consensus algorithm to access and submit transactions in to the block after authentication. However, in the existing Blockchain, there is no mechanism to ensure the trust and robustness of the miners and eliminate the malicious miners which runs the consensus algorithm. Therefore, this paper proposes a trust model with an objective of eliminating untrusted nodes from the mining process to enhance the reliability and security of the Blockchain. Further, the proposed trust model is suitably analysed for transaction rate, efficiency and scalability with Hyper Ledger framework to ensure the robustness.

Keywords: Trust Model, Blockchain, Consensus Algorithm, BFT, Scalability.

Received on 30 November 2021, accepted on 12 January 2022, published on 01 February 2022

doi: 10.4108/eai.1-2-2022.173294

*Corresponding author. Email: shalinir856@gmail.com

1. Introduction

Blockchain is a series of blocks with records in a distributed environment and are managed by group of computing nodes called peers with characteristics like immutability, transparency, scalability, reliability and etc [1]. Blockchain is perhaps the biggest innovation with a revealing future. Though the principal use of Blockchain innovation is Bitcoin as a cryptographic money, different use cases of Blockchain innovation have picked up variety of applications viz., administration, healthcare, SCM, etc [2].
Blockchain stores the transactions in a hash-based structure referred as Block. Each block contains three significant components namely the information, hash block and hash value of the previous block. The hash value in every block ensures the integrity of the data and any malicious change in the data will certainly affect the hash value. To further strengthen the security the entire block is protected by a hash and the hash is stored in the previous block. Therefore, any change in one block will affect the entire Blockchain and thus it is more secured. In brief the Blockchain incorporates distributed ledger, cryptosystem, consensus process and smart contract to provide its characteristics [3].
Blockchain is a decentralized information base with improved security and integrity of data that are stored regardless of the type of Blockchain namely permissionless or permissioned Blockchain. Permissioned Blockchain requires the nodes to be authenticated and controlled by central authority preventing accessing data directly unlike the permissionless Blockchain [4]. The permissioned Blockchain is further classified into consortium Blockchain and private Blockchain where the read and write operations are controlled by group of nodes and a centralized node respectively. On the other hand, in permissionless Blockchain any node can access the data without any centralized administration. Though there are many application areas in the Blockchain, apparently the consensus mechanism is important that decides the access control of the data with the help of set of nodes in the network namely miners. Miners are the nodes that validate

the transactions in the Blockchain access control with the help of consensus algorithms that guarantees the consistency and validity. There are numerous consensus algorithms exists in the literature. Out of which Byzantine Fault Tolerant (BFT) consensus algorithm [5] and its variants are widely used in consortium Blockchain.

Generally, the BFT consensus algorithm and its variants do not offer scalability. Moreover, failure in the primary node and untrusted nodes on the consensus group drops the overall performances of the consensus process. Therefore, in this paper, a trusted consensus algorithm to restrict the consensus group based on the trust values of the individual nodes is proposed.

In the proposed model all the nodes in the consensus group are most trusted nodes which will ensure the security and scalability. The scalability issues are restricted only with the nodes perform consensus process. The trust value of the individual nodes is proposed to be calculated based on these factors namely, History Factor (HF), Risk Factor (RF) and Feedback Factor (FF) [6]. The HF is obtained based on the successful and unsuccessful transactions committed by the node. The RF is obtained based on risk of deviation between the peers and the FF is calculated based on the feedback value of the neighbour nodes. Finally, these three factors are combined with an appropriate proportion to compute the overall trust of a node. Then the trusted consensus group is formulated with nodes having certain trust value and above based on a threshold value.

The main contribution of the paper is to formulate a trust model to construct a trusted miner's group for BFT consensus algorithm to enhance the performance and scalability in the consensus process.

The forthcoming sections are organised with a detailed literature of existing consensus algorithms in section II and the framework proposed trust model in section III. Section IV presents the experimental setup and evaluation parameters. The experimental setup also proposes an application to demonstrate the trust model. Section V finally concludes the paper.

2. Related Work

The Blockchain technology is a highly secured technology where maintaining the security and integrity of the transactions mainly depends on the consensus process that are performed by the miners. Therefore, identifying the trust among the peers and allowing trusted peers to involve themselves into the mining process is the primary goal in the trusted consensus process [7]. This section highlights the related concepts that are present in the literature.

Eigen trust based PBFT consensus algorithm [8] was proposed to improve the efficiency and to separate the malicious nodes in the network to form a consensus group based on the trust value. The author computes the global trust value of a node by summing up the local trust of a node 'i' with all the remaining nodes in the network. The local trust is computed based on the direct and recommended trust value. The basis for the direct trust depends on the satisfactory and unsatisfactory transactions between the pair of nodes. In this paper, the consensus group is formed based on the trust value and then PBFT algorithm is used for consensus process with trusted nodes to improve the efficiency. The major limitation of this method is that, it gives absolute ordering of peers without interpretation of recent history and feedback from other peers.

An enhanced Eigen Trust algorithm was proposed in [9], to identify and isolate a malicious peer in a network. This is carried out by analysing the recorded values of global reputation rating and reputation given by certain peer after each transaction they are involved in, to track and identify the malicious peer. This work computes the trust purely based on the feedback from other peers without considering the direct trust with the past history. Therefore, feedback from malicious peer will have a negative impact on the trust calculation. Further this method was not suitably analysed in the Blockchain network for effectiveness.

Peer Trust [10], is a reputable model for trust calculation of a node which includes two factors such as feedback received from other nodes and the credibility factor of the node that delivers the feedback. This model considers the recent transactions to calculate the trust in order to avoid traitor of the problem. It has also a limitation by considering only feedback to compute the trust of a node and has been proposed in a P2P environment.

Power Trust [11], is another trust management system dynamically a set of reputable power nodes using a distributed ranking method which once again uses first hand and second-hand information to build the trust of a peer. However, this method is not suitably analysed in Blockchain network effectively.

Wang et al., [12] has proposed a trust model for access control in collaborative environment based on history and peer recommendations to evaluate the trustworthiness of the requester. The author has used the number of transactions as a history and number of recommendations for computing the trust. Based on that, the trust score is calculated for access control. However, the freshness of history and Direct Trust Tree based recommendations were not considered. Further, trust value is used for access control for providing privacy.

Monrat et al. [5], has elaborated various consensus mechanisms that are used in Blockchain network for consensus process. Further the paper portrays the complete functionalities of Blockchain network process along with vulnerabilities which are happening in the consensus mechanism. However, there exists no discussion about trust model to avoid malicious miner in a network which could be harmful during the consensus process.

Xiaoyong Li, et al. [6], proposed a multi-dimensional trust evaluation model for P2P computing based on cognitive behaviour by incorporating multiple factors such as history, feedback, risk, motivation and availability factors. The weights are dynamically assigned using ordered weighted average algorithms. However, little effort is paid on identifying and evaluating certain factors for the recently joined peer. Further, this model has not been completely explored practically on various P2P model and in Blockchain networks.

Many research groups are proposing fuzzy logic concepts for building effective trust models in distributed computing and Blockchain technologies. Wang et al. [13], proposed a fuzzy based trust model to access the interactions where confidence is derived from direct and indirect transactions in distributed network systems. In certain other paper [14], the trust and reputations of nodes were built based on the observations and recommendations while fuzzy logic is used

for judgements. Fuzzy based Certain Trust Model was proposed [15], with average rating, certainty and expectation to build the trust model specific to e-commerce applications. However, significant research on application of these models in Blockchain consensus group has not been explored in the literature and forms the motivation for proposing a trust model in this research work.

3. Proposed Trust Model

This paper proposes a Trust based model to select the trusted miners who will involve in the consensus process for Blockchain network in order to process the block transactions in a more secured manner.

3.1 Overview of BPFT

Consensus mechanism is one of the important elements of a Blockchain which is responsible for verification process for inserting a block consisting of the transactions into a chain. The major objectives of the consensus algorithms are to reach an agreement by taking the collective interests of the group by collaborating with all the nodes in the distributed network [1]. Though there are many consensus algorithms exists in the literature, this paper makes use of the practical Byzantine Fault Tolerance (pBFT) [3] as it has many characteristic benefits of pBFT such as transaction finality, scalability, energy efficiency, non-requirement of asset, less computation, low reward variance, etc., make it more suitable for Blockchain.

The pBFT algorithm is an agreement calculation based on the Byzantine fault tolerance (BFT) [3], which has the essence of characteristic of a distributed network of reaching consensus even when some of the nodes of the network do not respond or respond with incorrect information. The goal of a BFT mechanism is to protect against system failure by making collective decisions to reduce the influence of faulty nodes. This is achieved by applying the concepts of Byzantine Generals Problem, in which generals have to tka a common decision to fight or withdraw the war. If a few of the wanted to fight when others wanted to withdraw, they cannot win the war. The important thing is a common decision has to be taken for the fight, otherwise the war would turn into defeat.

The pBFT consensus process selects a group of nodes with one node as leader and other are members. The nodes in the group communicates with each other to get a common agreement while processing the ledger in the Blockchain. To perform the process correctly, pBFT two third of the total nodes should agree with the consensus verification. Thus, when there are more nodes and two third of them are agreeing result in more reliability in the network.

The pBFT process has four phases [3] viz., 1) Client sends request to the leader about the transaction. 2) The leader then collects the transactions, group them into a block and broadcast the block to the other nodes in the group. 3) The nodes in the group verifies the transactions in the block. Upon verification of valid block, the hash value for the block is computed and broadcasted. 4) Finally, two-third nodes in the group reply with the same hash for successful consensus and the block is then added to the Blockchain.

The pBFT algorithm is designed for asynchronous consensus systems and optimized in a more efficient way to deal with all problems. The level of communication is quite high because nodes want to verify all the information that is found on the network. However, this algorithm suffers with lack of scalability since large number of nodes will communicate with each other for the common consensus which may sometimes include untrusted nodes in the network.

3.2. Proposed Framework

This work proposes a trust enabled consensus algorithm to make the consensus more reliable and scalable. In the proposed model a group of trusted nodes are formed which will run the pBFT consensus algorithm to make the verification more reliable and scalable. The new trust model verifies the node before getting into the consensus group based on three characteristic parameters viz., history of past performances, risk associated based on the previous transactions and the feedback of the peers based on their transactions [6].

The new trust model will calculate the trust value for each and every node based on the above-mentioned factors and a normalised trust value will be computed by applying appropriate weights for the three factors. Based on the normalised trust value of each node in the Blockchain network, nodes with trust value greater than the predefined trust value will only be included into the trusted consensus group. The nodes in the trusted consensus group alone will participate in the pBFT consensus for adding a block into the chain. This will improve the transaction rate, reliability and scalability in the Blockchain network.

The architecture and the workflow of the proposed trust enabled algorithm in the Blockchain network is depicted in the figure 3.1. In the diagram, the components shown in the rectangular box viz., trust calculation, fusion and rating are the proposed components of the trust model. In the proposed work, once the transaction proposal is submitted to the Blockchain network for inserting a block, the execution of consensus process will be preceded by formation of trust group with the trust model. The trust model uses three factors namely History Factor (HF), Risk Factor (RF) and Feedback Factor (FF) for trust calculation [6].

The first factor in the proposed trust model is History Factor HF_{ij} for two nodes P_i and P_j and is defined as the rating of P_i on P_j based on the satisfaction degree of their transactions over a time period ‘h’. This rating is purely based on the satisfactory and unsatisfactory transactions between P_i and P_j at a specific timestamp within the time period ‘h’.

The ratings are given as:
Ratings R = {r(1), r(2), …r(t), …r(h)}
Where, ‘t’ is timestamp and ‘h’ is history window size.

The ratings quantization for P_i will give score for P_j based on the performance as below:

$$r(t) = (sat(i,j) - unsat(i,j))/T_r \quad \text{----- (1)}$$

where, sat(i, j) is satisfactory transactions held between node i and node j, unsat(i, j) is unsatisfactory transactions held

between node i and node j, T_r is total transactions at the time of 't'. Based in the above ratings and an adjustable positive constant h, the HF_{ij} is calculated using equation (2).

$$HF_{ij} = (\sum_{t=1}^{h} r(t) * \alpha(t))/h \quad \text{-----(2)}$$

The HF_{ij} is calculated using the equation when h is not equal to zero otherwise it is assumed as 0. Also in the above equation $\alpha(t)$ is equal to 1 when t = h, otherwise $\alpha(t-1) = \alpha(t) - (1-\mu)\hat{}h$ where h acts as an adjustment constant which can be tuned and μ is a constant.

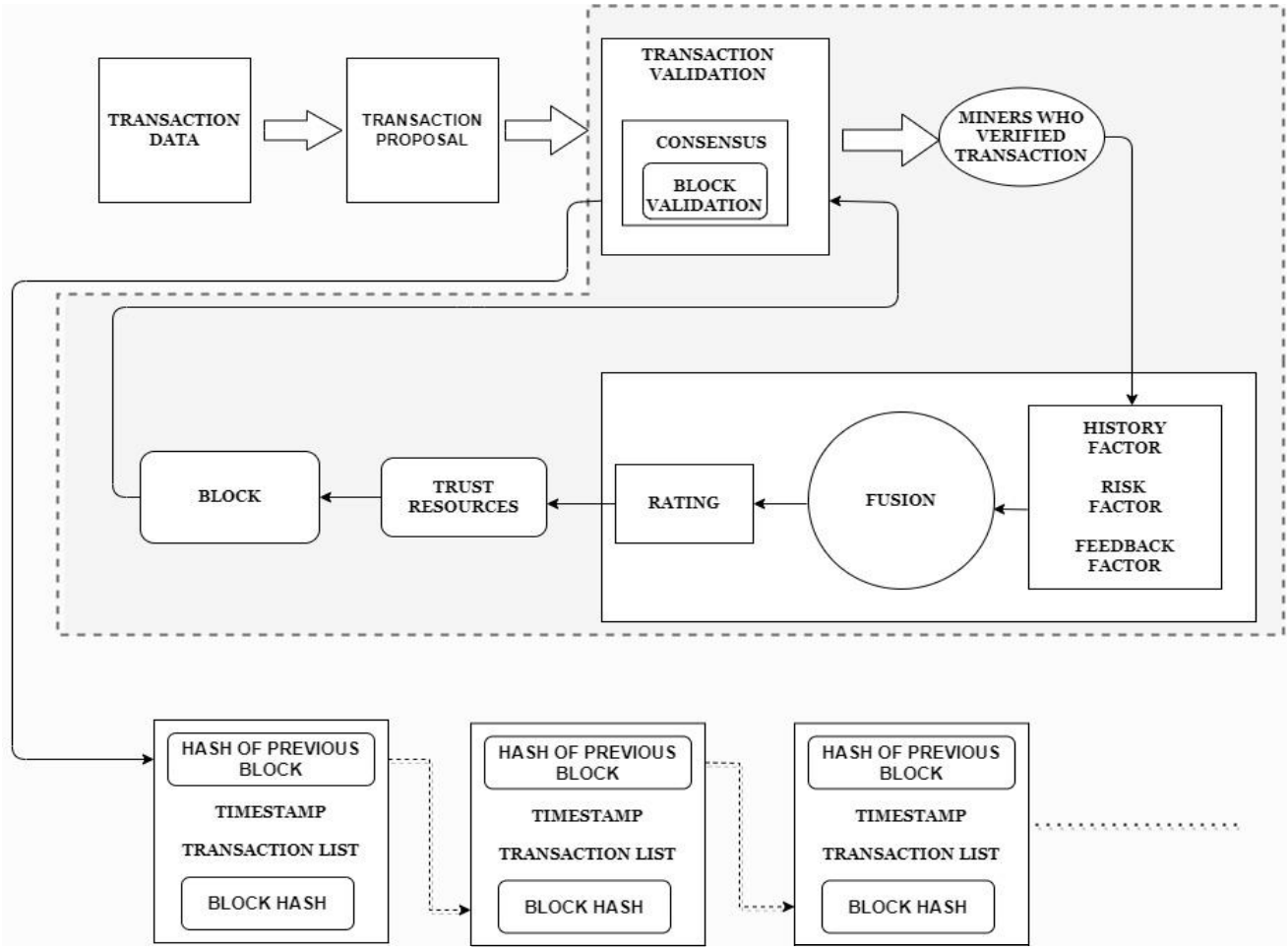

Figure 3.1. Proposed Architecture and workflow

Algorithm for the History Factor:

Algorithm 1. CalcHistoryFactor

Input : Node 'i' and a transaction node of i viz., 'j'

Output: History based Trust Factor HF_{ij}

1. h = n *// is a history evidence period and be set with a value*
2. S_{ij} = 0 *// the difference between the satisfactory and unsatisfactory transactions between i and j*
3. HF_{ij} = 0 *// Weighted average of past experiences between i and j*
4. t = x *// timestamp and the value is between 1 and h*
5. Initialize the values for $\alpha(t)$ for every t between 1 to h
6. μ is a positive adjustment constant between 0 and 1
7.
8. *// ------ for computing the probabilistic ratings based on the satisfactory and unsatisfactory transactions between i and j ------*
9. for every 't' in h do
10. for node j ∈ nodes do
11. S_{ij} = sat (i, j) – unsat (i, j)
12. $T_r = \sum_{k=0}^{n} \max(S_{ij}, 0)$
13. end
14. r(t) = S_{ij} / T_r
15. end
16.
17. *//------ for computing the history factor between i and j -------*
18. for every t = h to 1 do
19. $HF_{ij} = HF_{ij} + (r(t) * \alpha(t))/h$

20. $\alpha(t) = \alpha(t) - (1 - \mu)\hat{}h$
21. end
22. return HF_{ij}

The second factor is the Risk Factor RF_{ij} which is defined as the risk of deviation between Pi and Pj from the expected outcome based on the past results. Here, the risk rating is purely based on the number of failed transactions during time period 'h'. RF_{ij} is given by the equation (3).

$$RF_{ij} = 1 - \frac{r(h)}{h+\delta} \quad \text{----(3)}$$

where r(h) is the number of failed or unsatisfactory transactions during h risk window and δ is a constant adjustment factor.

Algorithm for Risk Factor:

Algorithm 2. CalcRiskFactor

Input : Node 'i' and a transaction nodes of i viz., 'j'
Output: Risk based Trust Factor RF_{ij}

1. z = n // *is a risk period and be set with a value*
2. RF_{ij} = 0 // *Weighted average of past experiences between i and j*
3. δ is a positive adjustment constant factor
4.
5. //------ *for computing the Risk based trust factor between i and j* -------
6. $RF_{ij} = 1 - (n(z) / (z + \delta))$
7.
8. return RF_{ij}

The third and final factor in the proposed model is Feedback Factor FF_{ij} which is defined as P_i has rated P_j based on the feedback using the Direct Trust Tree path function β(W(k)) and is defined by the following equation (4).

$$FF_{ij} = \sum_{N=1}^{Q} (\beta(W_N) * T(W_N, P_j)) / \sum_{N=1}^{Q} \beta(W_N)$$

------(4)

Where, W = {W_1, W_2, .., W_N, …..W_Q} is a set of feedback providers for P_j. FF_{ij} is calculated using the above equation for any Q value which is greater than 0, otherwise it is assumed 0.

For calculating the feedback factor using eqn. 4, the value of N ranges from 1 to Q, β(W_N) is said to be a path function in the Direct Trust Tree [11] where it reflects the feedback trust value. The feedback trust value is defined by the below equation.

$$\beta(W_N) = \prod_{Q=1}^{level} T(P_Q, P_{Q+1}), \quad level > 0$$

$$= 1 \quad level = 0 \quad \text{---(5)}$$

where, $T(P_Q, P_{Q+1})$ is defined as the trust value of node P_Q with its next descendant node P_{Q+1} in the direction of the trust path [11]. Further details and the construction of the direct trust tree and the principle can be read from the reference [11].

Algorithm for Feedback Factor:

Algorithm 3. CalcFeedbackFactor

Input : Node 'i' and a transaction node of i viz., 'j'
Output : Feedback based Trust Factor FF_{ij}

1. M = n // is the number of Feedback rater for node 'I'
2. FF_{ij} = 0 // between i and j
3. T (P_m, P_{m+1}) is the trust between the P_m and its descendent P_{m+1}
4. β(W(k)) is the path function in the DTT
5.
6. //---- *for computing the Feedback based trust factor between i and j* –
7.
8. for every feedback rater 'k' in M nodes
9. If level == 0 then β(W(k)) = 1
10. otherwise
11. for every m in level from 1
12. B(W(k)) = β(W(k))* T (P_m, P_{m+1})
13. end for
14. end if
15. end for
16.
17. for every feedback rater 'k' in M nodes
18. $F_n = F_n + \beta(W(k)) * T(W_k, P_j)$
19. $F_d = F_d + \beta(W(k))$
20. end for
21.
22. if M is not equal to 0 then $FF_{ij} = F_n / F_d$
23.
24. return FF_{ij}

The aforementioned trust factors for every nodes is calculated and are combined into a single Trust Value using the equation (6). The calculation for final trust value is nothing but normalizing the trust factors by applying weight values for the three factors in an appropriate fraction. This fraction may be changed based on the preference for the three trust factors. Here all trust parameters values are combined to give an overall trust value for a particular node in a network. The calculation is done using the following equation.

$$T(P_i, P_j) = w_1 * HF_{ij} + w_2 * RF_{ij} + w_3 * FF_{ij}$$

---- 6

Where, W = (w_1, w_2, w_3) are the weight of the related trust factor which is in the range between 0 and 1.

3.3 Consensus Group Formation and Consensus Process

In the trust enabled consensus process, before the actual process of consensus, the consensus group is formed with the trusted nodes in the underlying network. In the network the trust values for every node calculates the transaction behaviour, risk and feedback values from the other connected nodes. Further, these three trust values of individual nodes are normalised to get a single trust value. The normalised value is computed by suitably assigning weight value for the above said three factors namely history factor, risk factor and feedback factor. After computing the normalised trust value of every node, nodes which are higher trust value than the predefined threshold value is selected and included into the consensus group. The nodes with lesser trust value are identified as untrusted nodes and hence they are not allowed to enter in the consensus group. After forming the consensus group with only trusted nodes the actual consensus process is started. The transactions initiated by nodes are entered in to the transaction pool. Transactions from the transaction pool are grouped into a block and is subjected to the pBFT consensus processes. After the clear verification and consensus obtained from the pBFT algorithm, the Block is inserted in to the Blockchain. The pBFT consensus algorithm will allow only the nodes in the trusted group to participate in the consensus process. This process will ensure that the trusted nodes will be involved in the block creation and hence the reliability and security will further be maintained. As such in the proposed research, the trust model before the pBFT consensus process ensures more security in the Blockchain network.

4. Experimental Results and evaluation

4.1. Experimental setup

In order to evaluate the performance of the proposed trust model, simulation experiments were conducted using Hyperledger Fabric version 2.3. The proposed trust model is implemented using Go language. To analyse the performance, an application namely Project Management System for the funding agencies was considered. This system implements a consortium Blockchain in which the funding agency, monitoring agency and research institutions who are getting the grant from the funding agencies are members and form the Blockchain network. The proposal from the research institution will be received by the funding agency and approved based on the merit of the proposal. The approved institutions will get research grant from the funding agency and the institutions spend the money to carry out the research project. The entire workflow during the project period and the transactions generated were monitored by the monitoring agency. All the transactions were maintained in a secure manner in the Blockchain with the proposed trust model. This application was implemented using Hyperledger and Go lang using Docker environment. The smart contract and chain code was implemented using Go language. Docker and Docker compose was used to run the images of the Distributed Ledger/Nodes.

A distributed network with 15 nodes were created for experimentation. Each transaction of 54 KB was inserted into the Blockchain. To analyse the performance of the proposed trust model, experiments were conducted in three different scenarios namely, without trust model (all the 15 nodes), with trust model having threshold value 0.75 and with trust model having threshold value 0.6. The proposed trust model has been evaluated for three parameters namely, transaction rate, transaction delay and processing time.

Experiments were conducted for several trails to get the results. The obtained results were averaged over several runs and are presented as data points in the graphs for better accuracy.

4.2. Experimental results and analysis

The performance of the proposed trust model is analysed in three different scenarios for the following evaluation metrics.

4.2.1. Transaction Rate

Transaction rate is calculated based on the transactions per block against the processing time of the block. The transaction is given in terms of ms. The following graph in figure 4.1 shows the transaction rate for three different scenarios. The first scenario is without transaction model, the second scenario is with trust model having threshold value 0.75 and the third scenario is with trust model having threshold value 0.6. The data points were averaged by obtained data from three set of experiments. The experimental results shows that there is an increase of 21% transaction rate on an average.

4.2.2 Transaction Delay

Transaction delay is calculated based on the transactions per blocks compared with basic network of three nodes. The transaction delay is given in terms of ms. The following graph in figure 4.2 shows the transaction delay for three different scenarios. The first scenario is without transaction model, the second scenario is with trust model having threshold value 0.75 and the third scenario is with trust model having threshold value 0.6. The data points were averaged by obtained data from three set of experiments. The experimental results shows that there is a decrease of 37% transaction delay on an average.

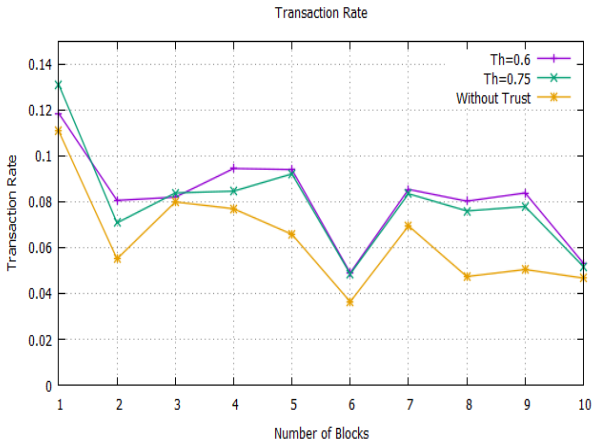

Figure 4.1 Transaction rate

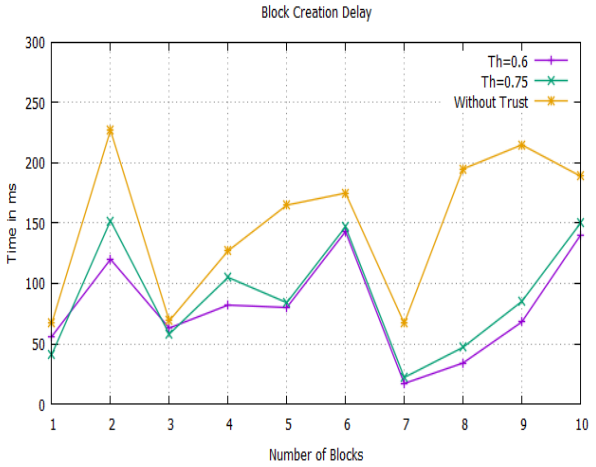

Figure 4.2 Transaction Delay

4.2.3 Block Processing Time

This metric evaluates the actual processing time of a block for the above three different scenarios. The following graph in figure 4.3 shows the transaction delay for three different scenarios. The first scenario is without transaction model, the second scenario is with trust model having threshold value 0.75 and the third scenario is with trust model having threshold value 0.6. The data points were averaged by obtained data from three set of experiments. The results show that there is a significant decrease in block processing time with trust model.

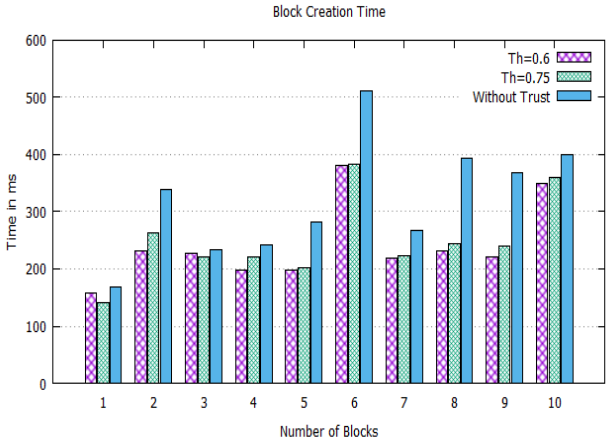

Figure 4.3 Block Processing Time

5. Conclusion

A new trust enabled consensus mechanism is proposed to enhance the performances of the BPFT consensus algorithm. The proposed trust-based consensus mechanism creates a trusted group of miners based on history factor, risk factor and feedback factor of the nodes to improve the efficiency. Since the trusted nodes are only present in consensus the communication complexity is also reduced to limited nodes and the transaction rate is also increased. This will address the scalability to some extend in the consensus process. The proposed trust model was experimentally analysed with a suitable application implemented in Hyperledger framework. The experimental results shown significant improvement in transaction rate, transaction delay and block processing time.

References

[1] Hosseini Bamakan, Seyed Mojtaba, Motavali, Amirhossein and Babaei, Alireza, “A survey of blockchain consensus algorithms performance evaluation criteria,” International Journal of Expert Systems with Applications, vol.154, no.4, pp. 1-39, Apr 2020.

[2] H. Wu et al., “Data Management in Supply Chain Using Blockchain: Challenges and a Case Study,” 28th International Conference on Computer Communication and Networks (ICCCN), vol.1, no.1, pp.1-8, Jul .2019.

[3] Fran Casinoa, Thomas K. Dasaklisb and Constantinos Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues”, Telematics and Informatics, vol. 36, no. 1, pp. 55-81, Nov 2018.

[4] Paul J. Taylor, et al., “A systematic literature review of Blockchain cyber security,” Digital Communications and Networks, vol. 6, no. 2, pp. 147-156, May 2020.

[5] A. A. Monrat, O. Schelén and K. Andersson, “Survey of Blockchain from the Perspectives of Applications, Challenges and Opportunities,” IEEE Access, vol.7, no. 1, pp. 117134 - 117151, Aug. 2019.

[6] Xiaoyong Li, Feng Zhou, Xudong Yang, “A multi-dimensional trust evaluation model for large-scale P2P computing,” Journal of Parallel and Distributed Computing, vol.71, no.6, pp.837-847, Jun 2011.

[7] Ahmadpanah Seyed Hossein, Jamili oskoue and Abdullah. “P2P Network Trust Management Survey,” Journal of Advances in Computer Engineering and Technology, vol.3, no. 2, pp.1-13, Apr 2017.

[8] Gao, Sheng, Yu, Tianyu, Zhu, Jianming and Cai, Wei, “T-PBFT: An Eigen Trust-based practical Byzantine fault tolerance consensus algorithm,” China Communications, vol.16, no.12, pp.111-123, Dec 2019.

[9] Alhussain, Alanoud and Kurdi, Heba, “EERP: An enhanced EigenTrust algorithm for reputation management in peer-to-peer networks,” Procedia Computer Science, vol.141, no.1, pp. 490-495, Jan 2018.

[10] Xiong, Li and Liu, Ling. “PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities,” IEEE Transactions on Knowledge and Data Engineering, vol.16, no. 7, pp. 843 – 857, Jul 2004.

[11] Xiaoyong, Li and Gui, Xiaolin, “Engineering trusted P2P system with fast reputation aggregating mechanism,” Journal of Advances in Computer Engineering and Technology, vol.3, no.4, pp.2007-2012, Jan 2008.

[12] M. Li, H. Wang and D. Ross, "Trust-Based Access Control for Privacy Protection in Collaborative Environment," IEEE International Conference on e-Business Engineering, vol. 1, no. 1, pp. 425-430, Oct 2009.

[13] Wang Yonghao and Zeng Guangping, “The Research on Trust Model of Internetware Based on Fuzzy Logic,” Procedia Engineering, vol.29, no. 1, pp.1356-1361, Dec 2012.

[14] Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia and Xingwei Wang, “TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things,” Computer Science and Information Systems, vol.8, no. 20, pp. 1207-1228, Oct 2011.

[15] K. W. Nafi, T. S. Kar, M. A. Hossain and M. M. A. Hashem, “A Fuzzy Logic Based Certain Trust Model for E-Commerce,” International Conference on Informatics, Electronics and Vision, ICIEV 2013, vol.1, no. 1, pp.1-6, May 2013.