Adaptive practical Byzantine fault tolerance consensus algorithm in permission blockchain network

G. Indra Navaroj

Department of Information Technology,
Jayaraj Annapackiam CSI College of Engineering, India
Email: indrajesus@gmail.com

E. Golden Julie*

Department of Computer Science and Engineering,
Anna University Regional Campus,
Tirunelveli, India
Email: goldenjuliephd@gmail.com
*Corresponding author

Y. Harold Robinson

School of Information Technology and Engineering,
Vellore Institute of Technology,
Vellore, India
Email: yhrobinphd@gmail.com

Abstract: Blockchain is a distributed ledger or data structure. Combined with many other technologies, it uses the internet of things, cloud computing, artificial intelligence, big data, and machine learning. Several industries, especially governments, have employed blockchain technology to overcome a variety of security challenges. Blockchain focuses on double-spending and distributed consensus. However, blockchain networks are inefficient and scalable. Communication overhead occurs due to many replications. This paper proposes an adaptive practical Byzantine fault tolerance algorithm in permission blockchains. This method divides the node into trust nodes and faulty nodes. The nodes with faulty reputations are excluded from voting. Also, the identified trust node has a high reputation in the consensus process. A majority of voting values select the master node. This adaptive PBFT algorithm is excellent for long-term periodicity and increased scalability, and lower overall communication costs. Finally, the performance of adaptive PBFT is compared to other algorithms.

Keywords: blockchain trust node; fault node; Byzantine; practical Byzantine.

Reference to this paper should be made as follows: Navaroj, G.I., Julie, E.G. and Robinson, Y.H. (2022) 'Adaptive practical Byzantine fault tolerance consensus algorithm in permission blockchain network', *Int. J. Web and Grid Services*, Vol. 18, No. 1, pp.62–82.

Biographical notes: G. Indra Navaroj received her BTech degree in Information Technology in 2007 from Anna University Chennai and MTech degree in Computer and Information Technology in 2010 from Manonmaniam Sundaranar University, Tirunelveli. Currently, she is pursuing her PhD in Anna University Chennai. Presently, she has been working as an Assistant Professor in Jayaraj Annapackiam CSI College of Engineering, Nazareth, India. She has 13.10 years of teaching experiences. She has written a book chapter and published many research papers in various fields. Her research areas include wireless sensor networks, blockchain and IoT.

E. Golden Julie is currently working as a Senior Assistant Professor in the Department of Computer Science and Engineering, Anna University, Regional Campus, Tirunelveli. She received her PhD degree in Information and Communication Engineering from Anna University, Chennai in 2017. She completed her ME degree in Computer Science and Engineering in Nandha Engineering College, Tamilnadu in 2008 and did her BE in Computer Science and Engineering in Tamilnadu College of Engineering, Coimbatore Tamilnadu. She has more than 12 years of experience in teaching. She has published more than 34 papers in various international journals.

Y. Harold Robinson is currently working in the School of Information Technology and Engineering, Vellore Institute of Technology, Vellore. He received his PhD degree in Information and Communication Engineering from Anna University, Chennai in 2016. He has more than 15 years of experience in teaching. He has published more than 50 papers in various international journals and presented more than 45 papers in both national and international conferences. He is acting as a reviewer of many journals like *Multimedia Tools and Applications*, *Wireless Personal Communication* by Springer Publication.

1 Introduction

The wireless sensor clustering network contain self-organising node, is an intelligent self-organising network, which collect the information from the environmental monitoring area. The important design issues of wireless sensor clustering network is to prolong the lifetime of network, balance the network consumption, note that the sensor node emerged only by batteries in important condition (Alfandi et al., 2020). There are three type of WSN such as centralised data gathering, many to one traffic pattern and multi-hop communication. These characteristic can lead to packet loss, packet collision and network congestion. To needed of the high energy, this is cause premature death of entire network and sensor nodes (Anisi et al., 2012). In this WSN each sensor node acts as a router to send or forward the data to the sink node by multi-hop path. But each sensor node has battery power without the facility of recharging (Cai et al., 2020). Here the sensor node energy is reduced for the process of communication. Due to energy shortage of sensor node lead to fail the node or that sensor node are died. So the operation of forward the data packet is loss that reduces the life time of network (Chatterjee et al., 2017).

Internet of things (IoT) contain different kind of sensor nodes that create, process, store and communicate large amount of critical and security data as well as sensitive information. So privacy sensitive data are communicate from one node to another node, node to human and sensor node to cluster head (CH) are appealing for various

communication attack (Feng et al., 2018). Large amount of new wearable and networkable device, which is IoT sensor nodes are lightweight and low energy. In WSN, IoT technology, highly affect the network communication. In centralised system are used for communication of IoT technology. At the same time IoT device increase the risk of communication to various privacy and security threats. This contradicts with communication privacy principles and security. Communication is an important concept of the human life and network also. These IoT systems cannot fully satisfy the task of reliable, uninterrupted, secure communication.

Now introduce the blockchain technology to overcome the critical issues of secure communication and information exchange in WSN and IoT. It provides the security in WSN peer- to- Peer networks and IoT. Transparent, secure, decentralised and distributed communication is supported with the blockchain Technology. In the case of WSN, IoT the blockchain can be used to manage the truth of information for nodes because any node can access the history of date in the public blockchain. The scheme is to determine the information truth worthiness and sensor node truth worthiness in the WSN, IoT. Then concentrate a blockchain for information exchange among the node or IoT devices. After that use a public blockchain that store the information worthiness and sensor node trust worthiness in a distributed ledger that is appropriate for secure communication. Also introduce different kind of blockchain consensus mechanism support scalability and security of a blockchain.

In the recent years, many researchers concentrate on blockchain consensus algorithms. The consensus algorithm gives the solution for deciding a distributed environment challenges (Nakamoto, 2009). Why consensus is needed in a distributed environment? If only one node is present in the network, then do not need the agreement (decision). But more number of node current in the distributed environment has several decision-makers and hence that needs the consensus (Dinh et al., 2018). Blockchain is a distributed, decentralised, and public digital ledger in which transaction records between two nodes in the blockchain network these records cannot alter by any other node (Puthal et al., 2020). In distributed systems, consensus mechanisms have been a significant problem. Deploying and designing consensus algorithms are a vital task as it needs several critical issues like resiliency against node failure, network partitioning, node failure, corrupt or out of order input, network latency (Baliga, 2017). Consensus algorithm is also used to avoid the fork in the blockchain networks. In the blockchain network forking problem is defined simultaneously by two miners mine the same block of the transaction (Sankar et al., 2017).

The vital problem of consensus algorithms are consistency, availability and fault-tolerant. But the entire consensus algorithm cannot satisfy all three properties in an asynchronous environment. In distributed systems, these algorithms tend to provide safety and liveness over fault-tolerant. There are two major fault-tolerant problems in distributed environments such as crash fault and Byzantine fault (Cachin and Vukolić 2017). Crash fault happen the nodes fail due to some software and hardware failure. It may occur at any time without any warning. After that network cannot perform any further action and the failure nodes remain unresponsive. Byzantine fault tolerance (BFT) determines the reliability of a distributed system where nodes might be unsuccessful, and the output is flawed data. One of the critical issues have been addressed by the blockchain network is BFT. BFT is two sensor nodes that can transmit safety across the system when the sensor nodes are displaying the similar information. In a Byzantine failure, a node can inconsistently fail, but at each time could present different symptoms

(Bano et al., 2017). In this case, active nodes have to agree on a sign to evade the entire system failure. BFT is also called as error avalanche congruency, Byzantine agreement and Byzantine general issues too.

Without BFT, a sensor node can broadcast false data to the network making data on the blockchain untrustworthy. When implementing the BFT, ensure that all nodes prevent with the practical Byzantine fault tolerant (PBFT). In PBFT, all the nodes in the blockchain network participate in the voting process for adding the new block in the system. Here this is need 2/3 nodes consensus to add a new block in the network. PBFT concept is more economical; it is suitable for permitting blockchain. It would perform fault tolerance against malicious nodes. There are two types of blockchain, such as permissionless blockchain and permission blockchain. A permissionless blockchain is open; the participating nodes cannot be known by each other. It contains the un-trusted nodes. Here consensus is critical because each node creates a validated block, then the block is under the conclusion of entire nodes in the blockchain. There are two types of blockchain: permission blockchain and permission less blockchain. *Proof of Work*, *Proof of Stack*, and *Proof of Burn* are permission less blockchain consensus algorithms. These algorithms need more communication cast in the blockchain network. PAXOS, RAFT, BFT, Practical BFT algorithms are permission blockchain consensus algorithms (Biswas et al., 2019).

The major contribution of the proposed work is

- An adaptive consensus algorithm is performed by using three parameters to select a master node (chain head).
- Find the trust node (honest node) among the client nodes based on their valid transaction which allows the trust node for the voting process and also remove the concept of view.
- Find the faulty node (Byzantine node or malicious node) among the client nodes based on their invalid transaction. After finding the faulty nodes, that node's reputation value decreases and are removed from the voting process.
- Threshold measure is calculated at a specific time interval, and the master node (CH) will rotate in equal chance.

The rest of the paper is presented as follows. In Section 2 contain the related work. In Section 3 contain proposed method of APBFT algorithm, flowchart of APBFT technique, Master node replacement. In Section 4 compare the performance evaluation of our proposed APBFT algorithm with existing PBFT algorithm. Finally the conclusion of the paper is demonstrated in Section 5.

2 Related work

In a message-passing system, a general behave maliciously. Consider the four generals as General A, General B, General C, General D. All the Generals communicate using the phone call. General B sent his own opinion of attack. So the soldiers are ready to attack. Now General C makes the phone call to A and B and sends his personal decision to attack. If General A sends the attack message to General B but sends the retreat message to General D, in this situation, General A acts like a malicious. Now General D gets

confused when making the decision. In a distributed or decentralised message passing environment, achieving consensus is very difficult. This kind of node acts like a malicious and is called as malicious node. This malicious node is called as Byzantine time node, and failure is called as Byzantine time failure (Lamport et al., 2019). If no failure occurs in the system, it is easy to reach the consensus.

The Byzantine fault-tolerant protocol supports the synchronous environment. But our real-time system is asynchronous. So that is need another mechanism called PBFT Algorithm which is termed as practical because it ensures safety over an asynchronous network, but not aliveness on a simple asynchronous system. Otherwise, it will remain inviolate the impossibility theorem or the impossibility principle. So, to ensure aliveness, those have a weak asynchronous assumption where this is deviate from a simple asynchronous system. The system can support Byzantine failure, and it has low overhead. So, that is why the proposed technique call a system as an asynchronous method. PBFT algorithm can be applied in the permission blockchain network. The model contains 3f + 1 replicas, in an asynchronous environment, and the method require 3f + 1 replica (Kiki et al., 2020).

Federated Byzantine Agreement (FBA) algorithm has achieved robustness by quorum slice. Each node decision is used for determining a system-level quorum. The author has also proposed Steller consensus algorithm (Mazieres, 2015). In FBA each node in the network knows other node. So before completion, the node transaction waits for the other node agreement. Here the FBA ensures the integrity of the blockchain transaction. BFT algorithm requires an additional cost of resource; algorithm complexity and performance. BFT does not control the faulty node and do not support the message delivery schedule scheme due to too many attackers present. The author has proposed cross fault tolerance or XFT; this approach is used to increase reliability and security (Liu et al., 2016). XFT SMR algorithm is providing safety in an asynchronous system. It provides BFT in an asynchronous network as well.

The honey badger BFT is use to provide liveness with a novel atomic broadcast protocol that also provides optimal efficiency. It is a first atomic broadcasting algorithm which helps to achieve optimal asymptotic efficiency as the system contains designated nodes and all nodes agree to the transaction of all nodes. 3f + 1 <= N is the lower bound for broadcast protocol (Miller et al., 2016). This proposed algorithm contains three steps, such as purely asynchronous network, static Byzantine fault, and trusted setup. This approach is used to improve the scalability of the network. The author has described the PBFT algorithm to identify faulty nodes and difficult to remove faulty in time; Here, the primary node faces many of the attacks. Proposed reputation BFT (RBFT) algorithm is to evaluate all the node operations in the consensus process. In the voting process, faulty nodes have lower right to vote for candidate nodes. If any Byzantine faulty node is detected, then the node reputation will be reduced. Then the author introduces an innovative reputation based primary change scheme. The sensor nodes have a high reputation, to give a higher chance for the voting process and to create new valid blocks. The proposed algorithm is to ensure system security, reliability and better performance compared to PBFT. Here removed the view concept from the voting process and the primary node elected based on the rotation (Lei et al., 2018).

An innovative collective decision algorithm is for using the blockchain consensus process. This algorithm is used to find the faulty node; at the same time, it provides corrective decisions even in the presence of malicious nodes (Iyer et al., 2019). This algorithm generates a peer to peer network of PMUs and can find malicious data.

Proposed PMU is used to provide the consensus of nodes. As described in the PBFT algorithm, master node selection is one of the significant problems and communication overhead has increased. In regard to the proposed Egalitarian PBFT algorithm, the primary node selection process is removed. So, the nodes create a block and through broadcast request, other nodes are constructed, broadcast to the network and broadcasting the current status to the system is also carried out. The entire node in the network is treated as equal and efficient (He and Hou, 2019). Here the efficiency of consensus is improved, and communication overhead is reduced half by less view change. By this algorithm being compared with PBFT the throughput has increased, communication overhead and delay also has been reduced.

The PBFT algorithm cannot satisfy the scalability in an extensive blockchain network. So the author proposed a novel optimised PBFT consensus algorithm based on the concept of eigen trust model. TPBFT is a multistage algorithm. Here the node is divided into two types such as transaction node and non-transaction node. Then it would be feasible to calculate global trust value, and then to calculate trust node value between nodes (Gao et al., 2019). The highest trust values of nodes can construct the consensus group. This algorithm reduces the view change process and replaces the single primary node by a group of central nodes – this proposed algorithm is used to reduce the communication overhead. PBFT algorithms cannot stimulate positive nodes effectively and large numbers of communication resources are needed. Here, the offered credit delegated BFT (CDBFT), works based on two techniques. First one is voting rewards, and punishment schemes are a credit evaluation scheme (Wang et al., 2019). That scheme is used to reduce the faulty node in the consensus process and support reliable node simulation. The second one is consistency, and checkpoint protocol requires improving the efficiency and flexibility of the system. This CDBFT algorithm is used to reduce the faulty node in 5% and improve the stability and effectiveness of the network.

Delegated Adaptive BFT algorithm is an empowered approach in which a more flexible DBFT can elect BFT flavours suitable for parallel tasks. A concept of adaptiveness extends DABFT. The new block is generated based on the validator by a task validator – this proposed method is used to improve efficiency and reduce the system complexity (Deng, 2019). SBFT is a scalable and decentralised trust infrastructure. BFT works in a group of hundreds of replicas especially in a world-scale deployment. The proposed SBFT is to work in a 209 model with f = 64 Byzantine fault in a world scale geo-replicated deployment (Gueta et al., 2019). SBFT increases the performance and scalability of the network. Compared to PBFT, SBFT provides 2 × better throughputs, and 1.5 × latency is a highly optimised network. This SBFT approach contains four steps such as collector, use an optimistic fast path, utilising redundant servers for the quick way and reduce client communication. The author introduces the delegated BFT algorithm, and hence forth an improvement of PBFT. In this proposed approach, node selection is delegated to other nodes (Ray et al., 2020). DBFT algorithm is used to improve network efficiency and scalability.

The author proposed secure and highly efficient PBFT–SG PBFT for the internet of vehicle application. This algorithm based on the concept of distributed structure. It is reduce the issues of single node attack and reduce the complexity of central storage. Introduce score grouping mechanism to improve higher consensus efficiency and also reduce the overhead of PBFT consensus. Geographic PBFT algorithm is used to overcome the Sybil node attack, low scalability and also suffer from high computational cost. A new scalable consensus protocol and location based algorithm is designed for

IoT-blockchain application. After that described the fixed sensor node need more computational power than mobile sensor node (Guan et al., 2010). Sometime the mobile sensor nodes act as malicious node. It is also reduce the overhead of recording transaction and validating transaction. G-PBFT algorithm is reduced network overhead, reduce consensus time.

BFT algorithm is selects a collection of authenticated node in the blockchain network. It provide the energy efficiency compare to other consensus protocol selected authenticated nodes are ensure the integrity of data block in the blockchain. And also avoids the concurrent blocks that have malicious data in the blockchain network. Dynamic reputation PBFT algorithm works like a credit based consortium node selection technique. Chain head is elected as a monitoring node, that node is divide the remaining participated node into the consensus node and the secondary node based on the reputation value of the node. These nodes are participated in the Block generation process (Lao et al., 2020). This protocol is increase the transaction speed and is also suitable for blockchain energy field. The author proposed SDMA-PBFT algorithm, that the number of consensus node increase in PBFT algorithm. scalable dynamic multi-agent hierarchy PBFT algorithm to reduce the communication cost from $O(n^2)$ to $O(n \times k \times \log kn)$. Internal nodes voting result can be collected effectively. The nodes are includes client node, consensus node and storage node. Author modified the reply phase and pre-prepare phase in the network. Each node in the network sends a message to all other participated nodes. The author present mixed BFT (MBPFT) algorithm adopts layered technology. MBFT algorithm introduces a credit mechanism and a random node election technique to improve fault tolerance and security. The nodes are divided into client node, backup node and verifying node. Verifying node is verified all the node transaction and store the transaction list. Then the verification node reports the malicious node details to the blockchain network (Li and Li, 2018). MBFT provide high throughput, scalability and good security. It mainly concentrates on fault tolerance of the blockchain and improves the throughput and scalability with the detailed comparison is illustrated in Table 1.

Table 1 Comparison of PBFT algorithms

Algorithm	*Advantages*	*Disadvantages*
PBFT	1 In PBFT, all the nodes in the blockchain network participate in the voting process for adding the new block in the system. 2 PBFT concept is more economical; it is suitable for permitting blockchain. 3 PBFT algorithm which is termed as practical because it ensures safety over an asynchronous network.	1 The PBFT algorithm cannot satisfy the scalability in an extensive blockchain network. 2 The PBFT algorithm to identify faulty nodes and difficult to remove faulty in time. 3 The PBFT algorithm, master node selection is one of the significant problems and communication overhead has increased.

Table 1 Comparison of PBFT algorithms (continued)

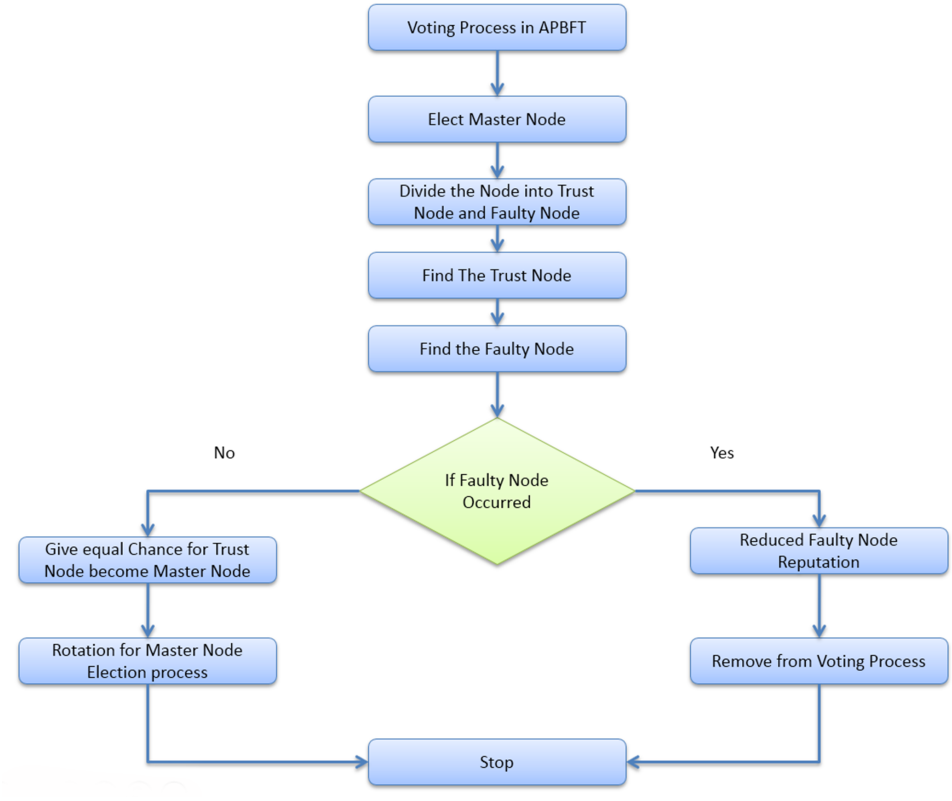
Algorithm	*Advantages*	*Disadvantages*
Honey badger BFT	1 It provides optimal efficiency. 2 All nodes agree to the transaction of all nodes. 3 It is used to improve the scalability of the network.	1 Difficult to remove faulty node in time; 2 The primary node faces many of the attacks. 3 If any Byzantine faulty node is detected, then the node reputation will be reduced.
XFT	1 This approach is used to increase reliability and security. 2 It provides Byzantine fault tolerance in an asynchronous network as well. 3 Algorithm is providing safety in an asynchronous system.	1 BFT algorithm requires an additional cost of resource; 2 Algorithm is complexity and performance. 3 BFT does not control the faulty node and do not support the message delivery schedule scheme due to too many attackers present.
FBFT	1 Algorithm has achieved robustness by quorum slice. 2 The node transaction waits for the other node agreement. 3 The FBA ensures the integrity of the blockchain transaction.	1 The FBFT is increase the Network overload. 2 It is reduces the system performance. 3 It is not support the scalability and security
RBFT	1 Ensure system security, reliability and better performance 2 The nodes have lower right to vote for candidate nodes. 3 The sensor nodes have a high reputation, to give a higher chance for the voting process and to create new valid blocks.	1 Communication overhead increase. 2 Time complexity. 3 the amount of the participating nodes is large
TPBFT	1 It would be feasible to calculate global trust value, and then to calculate trust node value between nodes. 2 This algorithm reduces the view change process and replaces the single primary node by a group of central nodes. 3 Algorithm is used to reduce the communication overhead.	1 PBFT algorithms cannot stimulate positive nodes effectively. 2 The large numbers of communication resources are needed. 3 The claimed performances of these consensus protocols are worrisome and unconvincing

3 Proposed method

The proposed algorithm is an adaptive practical Byzantine fault tolerant algorithm. The blockchain network consensus is essential for distributing messages among the nodes.

Here, that algorithm cannot concentrate on one node in the network consensus as several nodes present in the distributed environment consensus are essential. In permission blockchain, different types of consensus algorithms are used, such as PAXOS, RAFT, and PBFT algorithms. Now that is could concentrate on the agreement in the blockchain network using PBFT protocol. In recent years, many of the authors have proposed different kinds of methods for changing the PBFT protocol. But algorithms cannot support the network lifetime. Here, the proposed technique mainly focuses on the network lifetime of nodes and it is demonstrated in Figure 1. In our proposed system model there are three nodes such as master node (chain head) {M = M_1, M_2 …}, trust node {$T = T_1$, T_2…..}, faulty node {$F = F_1$, F_2 …} In a group of nodes especially in a blockchain network, master node monitors the transactions and checks whether the transactions are valid or invalid transactions. Both trust nodes and faulty nodes send transactions and participate in the voting process. These client nodes can select the master node in the blockchain network.

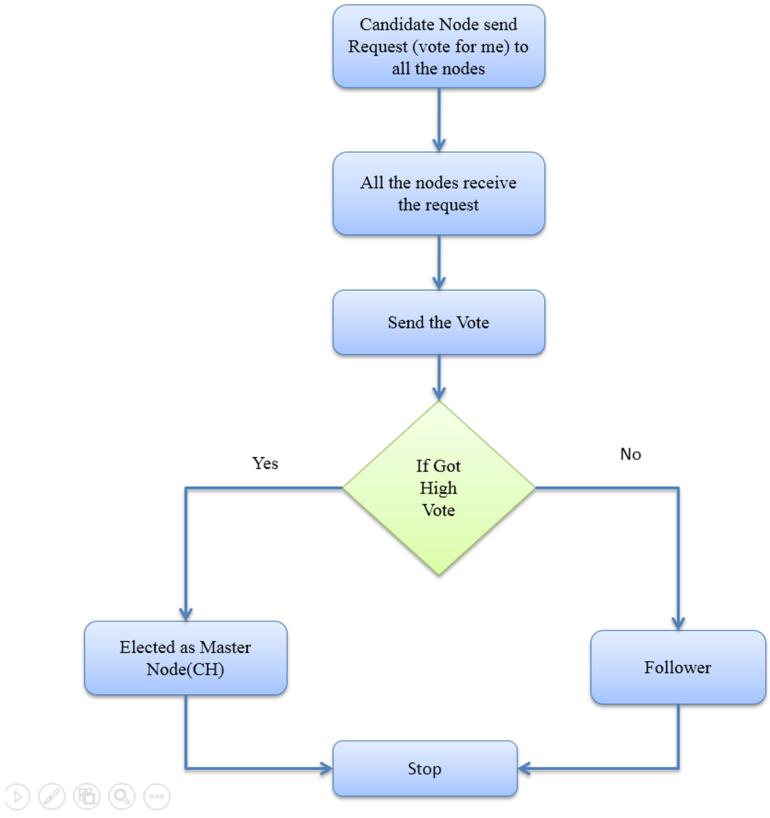
Figure 1 Flowchart of the APBFT technique (see online version for colours)



Based on the method of proposed adaptive PBFT algorithm, first, select the master node (chain head) among the client nodes with the majority of voting in the voting process. Here the candidate nodes can send the request message to the client nodes in the network. Master node monitor could control each node transaction and helps to find the trusted node and fault node. The trust nodes are allowed to participate in blockchain, and faulty

nodes are removed from the voting process. Finally, the master node (chain head) is elected among the rotation basis. Next, the trust nodes act as candidate nodes for the election of master node (chain head). The proposed adaptive PBFT algorithm is used to improve the efficiency of the network and it is illustrated in Figure 2.

Figure 2 Master node election (see online version for colours)



The consensus algorithm in the blockchain is a technique in which the nodes reach the consensus on the same result when there is a time change in satisfying the information in the blockchain network. The nodes collect the transaction history and share the blocks with the nodes. The nodes execute the consensus method to prevent the faulty, malicious and validate the shared blocks. Suppose the node will modified or delete the data as well as ensuring reliability. If the faulty nodes attempt to modify the transaction history, attempt to break block rights, it is not at all possible that the block history would be modified. Many of authors proposed different method of PBFT consensus algorithm to reach the consensus in blockchain network.

Algorithm 1 Master node selection

Let Vn(N)= Threshold voting value of client node N

Vn= number of client in the voting process

Transaction flooding ()

```
{
    Vn(N) = voting of client node(N) in the network
while(time < Vn(N)
{
    candidate node send the request to all the client node in the network
    Receive vote(V) from the client node
calculate number of vote for client node in the network
{
    calculate MV(node(N)
if(time α MV(node N) then
{
    MV (node n) >= Vn (N)
    calculate MV = maximum number of vote/number of client in the voting process
}
    MV candidate node selected as a master node
}
```

Algorithm 1 explained the Master Node selection process in the blockchain network. APBFT protocol is based on the concept of PBFT protocol. Here the proposed Adaptive PBFT algorithm selects the Master node as Chain Head. In Adaptive PBFT algorithm there are number of client nodes in the distributed blockchain network. Some of the client node acts as a candidate node for the voting process, and the remaining nodes act as a participation in the voting process. Then the entire client node can select the Master Node as Chain Head. Then the candidate node can send the request message to the client node Vote for me. An all the client nodes in the blockchain network receive the request message from different candidate's node and would vote for anyone of the candidate nodes. At the same time, the candidate node will receive voting from various client nodes. Finally, calculate which candidate node got a majority of voting in the voting process, that winner is elected as a Master Node and another client node acts as a follower of Master Node. In this proposed algorithm, the Master node monitors the transaction of the client nodes in the network.

Algorithm 2 Divide nodes

```
Let Tn =Trust node
Let Fn =Faulty node
N= number of node in the network
{
    Trust node<- ϕ, Faulty node<- ϕ
{
for (node j ∈ N )
{
if (node j > valid transaction with node) then
    Trust node ← node j
```

```
    else Faulty node ← node j
}
}
}
```

Algorithm 2 explained the Adaptive PBFT algorithm, here the client nodes in the network are divided into two types of nodes such as Trust node (Tn) which is also called as honest node in the blockchain network, Faulty node (Fn) also called as malicious node in the blockchain network. Master Node monitors the client node activities such as finding the valid transaction and invalid transaction. Fault node means continuously sending the faulty information in the network. Trust refers to the client node which sends the valid transaction in the network and that client node is called as Trust node.

Algorithm 3 Find trust node

```
Let TijV = Trust node transaction value
TijV ← 0
{
for( node ∈ Trust node)
    SijV = trust(i, j) – untrust(i, j)
    St = Σmax(SijV, j, 0)
}
{
if(St = 0) then
    set TijV = 1/N
else
{
for( node j ∈ Trust node)
    TijV = max (Sv, 0)/St
}
}
```

Algorithm 3 explained the Trust node calculation based on the valid transaction of the entire node in the network. In the Adaptive PBFT algorithm, Third step is found as the Trust node. If the client node is a trust node, then the nodes are allowed to be involved in the voting process and send the valid transaction in the network. In future, these trust nodes can act as candidate nodes for the voting process. This algorithm allows only the trust node and so the network lifetime increases, packet delivery ratio increases and throughput increases too.

Algorithm 4 Find fault node

```
Let FijV = Faulty node transaction value
TijV ← 0
{
    find the transaction value between nodes for node j ∈ faulty node
```

```
{
for (node j ∈ faulty node)
{
if (node ik ∈ Trust node and node ik ∈ trust node)
{
    TijVi,j = ΣTikV Cjk
else
    compare TijV iteratively
}
}
}
}
```

Algorithm 4 explained the finding of the fault node in the network from the trust node. This algorithm is used to find the fault node, at the same time remove the faulty node from the voting process. Here the fault nodes are not able to participate in the voting process. This method is used to reduce the network partition.

Algorithm 5 Master replacement

```
Manage Master Node
{
    Rotation check = Master Node Round % update interval
if Rotation check = 0
{
    Master Node Rotation = change Master Node()
if (Energy< Energy of Threshold node in the network) then
{
    Reconstruct Master Node
    update Routing table
}
else
    Master Node Rotation = same Master Node ()
}
}
```

Algorithm 5 explained that periodically, master node should be replaced by rotation technique. It is used for increasing network lifetime. If the same node is the Master node for a long time, it will lose energy leading to a dead node. So here, replace Master node and update the routing table periodically and this method will increase network lifetime. This method also allows the other Trust Node to act as a Master Node for the rotation basis. A blockchain network needs a minimum number of nodes for the network to make accurate decisions and run properly. Quorum contains an honest node (trust node). In this blockchain network, to avoid the network from stalling, there must be at least one non-malicious node in equation (1)

$$Q <= N - f \tag{1}$$

In the blockchain network, to avoid the network from splitting into different decisions, the majority of honesty should be present. Here need a majority trust node and Quorum size should be greater than half of the total number of nodes present in the network.

$$Q > N / 2 \tag{2}$$

$$2Q - N > 0 \tag{3}$$

by combining the two conditions

$$N < 2Q = 2(N - f), f < N / 2, N > 2f \tag{4}$$

$$\text{if } N_{min} = 2f + 1, \text{ then } 2Q > f + 1 \text{ or } Q > f + 1 / 2 \tag{5}$$

Therefore non-Byzantine failure Quorum size $Q_{min} = f + 1$

Given N nodes in a blockchain network, with f fault nodes might have Byzantine failure. Then find the Quorum size. Byzantine failure nodes can vote for an invalid transaction or decision can lead to split the network and as a result forking emerges. Byzantine nodes make different statements to different node consequences. It may lead to a non-failed node into divergent state and stuck state. In a blockchain network to avoid the network from stalling there should be at least one non-faulty node. It is possible that faulty nodes may not reply to transactions.

$$Q <= N - f \tag{6}$$

To avoid the network from splitting into different decisions, the majority should be present. In Byzantine failure nodes can also vote. So those need to consider the faulty node in the voting process.

$$2Q - N > 0 \tag{7}$$

This equation is used to find the maximum number of failed nodes that can be present in the blockchain network.

$$2Q - N > f \tag{8}$$

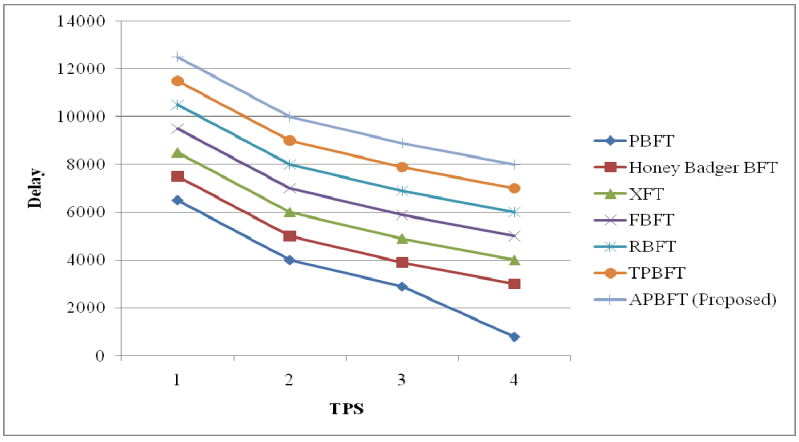
where f is the maximum number of faulty node

4 Performance evaluation

Initially the proposed technique has been analysed the efficiency, scalability and effectiveness of the proposed APBFT, and then compare the APBFT with other existing BFT consensus algorithms in blockchain. The performance of our proposed APFT consensus algorithm with the existing BFT type consensus algorithm like PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT can improve the throughput, reduce the delay, minimise the communication overhead and compare the performance of the Faulty node rate. The proposed technique also discusses the algorithm so as to complete the consensus work quickly. The fastest algorithm is the best consensus algorithm. Throughput means the total amount of transactions processed per unit of time. The result is shown in Figure 3. It can find that our proposed APBFT algorithm has a inferior

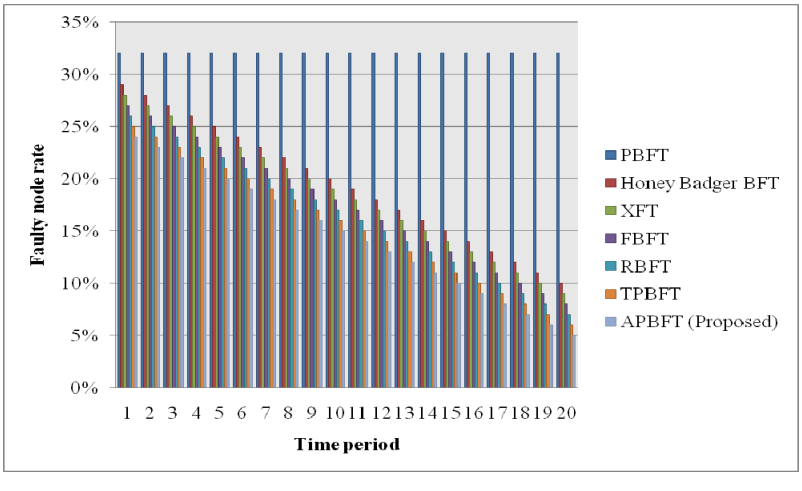
consensus delay than the PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT algorithm below the similar TPS.

Figure 3 Delay (see online version for colours)



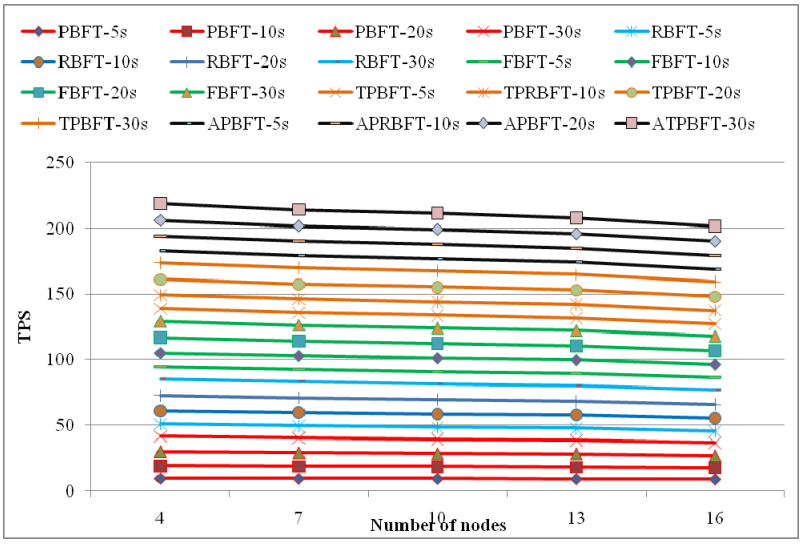
The faulty node rate of the proposed APBFT consensus algorithm with the existing BFT approach has been compared. Figure 4 shows the change in the faulty node rate. The output shows the existing PBFT algorithm; faulty node rate is not changed in the running time of the network, while the proposed APBFT consensus algorithm can effectively identify malicious nodes or faulty nodes and eliminate the faulty node in the blockchain system. So that our proposed APBFT consensus algorithm is to ensure the availability and security of the network, here the faulty node rate is significantly reduced when compared to other existing BFT type consensus algorithm. The author analysed the throughput (TPS) and the delay of our blockchain system with the different number of nodes and block production period. Here the number of error nodes is 1, 2, 3, 4 and 5, and the number of trust nodes is 4, 7, 10, 13and 16, respectively. In the blockchain network, the calculated throughput is called as TPS.

Figure 4 Faulty node rate (see online version for colours)



Here, that is measure the new block creation period from 5 to 30 second in every 5 seconds when the system is running continuously. This is calculating the 30 blocks and measure the average delay and throughput. Figure 5 illustrates the evaluation result of our proposed APBFT consensus algorithm under different number of nodes and different block generation period.

Figure 5 Throughput (see online version for colours)



Under the same node and at time intervals, the block generation ratios are set. Here the proposed technique measures the TPS of APBFT with existing BFT type consensus algorithms like PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT algorithms. The APBFT algorithm is higher than the existing BFT type consensus algorithms like PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT algorithms. Here the delay value of our proposed APBFT algorithm is measured as the delay value is lower than the existing BFT type consensus algorithm like PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT algorithm. So the overall performance of APBFT consensus algorithm in blockchain network is high when the faulty node is present.

The communication overhead of our proposed APBFT algorithm with other existing BFT consensus algorithms is PBFT; the consensus process of the total number of communication overhead is $2n^2 - n - 1$. Here $C = 2n^2 - n - 1 + Pn(n - 1)$. The number of communications is $(n - 1)^2$. Here $c = n(n - 1) + P(n^2 - 1)$. Here the proposed technique discusses the communication overhead of the proposed APBFT algorithm with other consensus algorithms.

Here focus on four parameters like 4 nodes, 6 nodes, 8 nodes, 10 nodes for calculating the communication overhead of the consensus algorithms. However, there are no view changes in the consensus process of our proposed APBFT and the total number of communication time is 40 times. The result is shown in Figure 6. Our proposed APBFT algorithm is a low communication overhead when compared to the existing BFT consensus algorithms.

Figure 6 Communication overhead (see online version for colours)

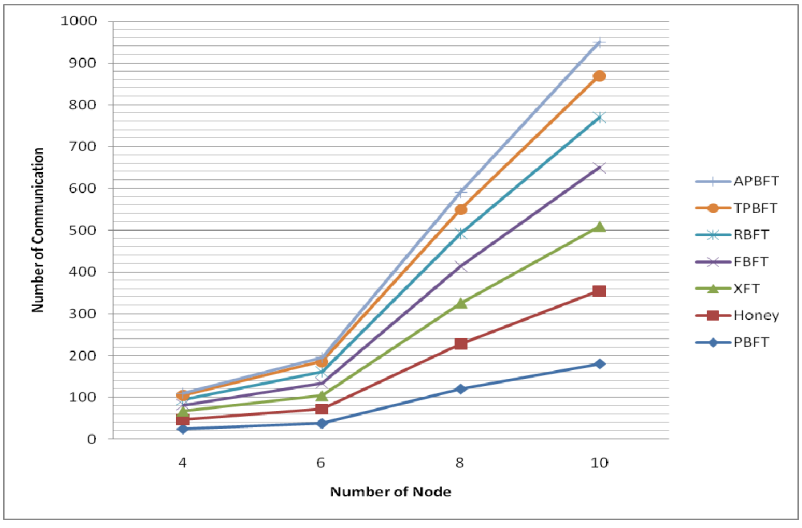
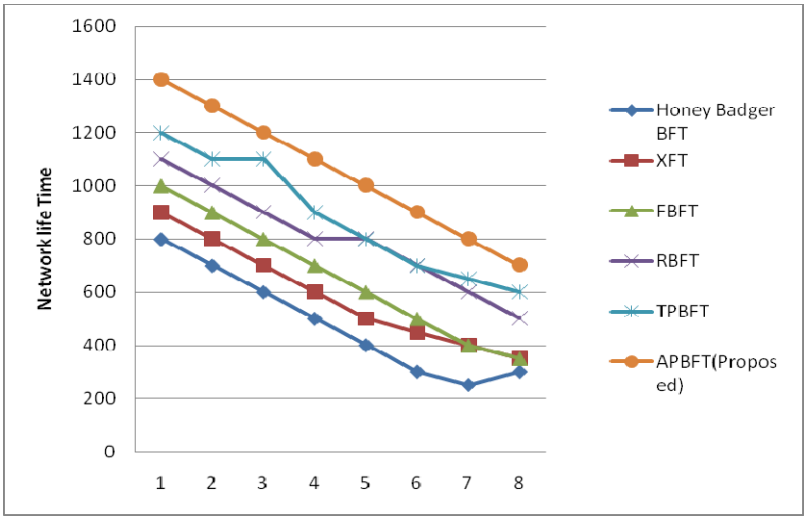
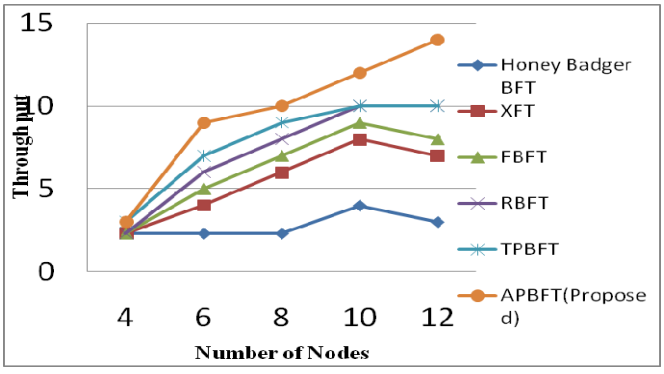


Figure 7 Network lifetime (see online version for colours)



The comparison for the performance of network life time is completed when network dimension increase from 100 m–500 m. In this Honey Badger provide low network lifetime performance. In this XFT algorithm has small improvement than Honey Badger. But the FBFT has the better performance compare to the PBFT, Honey Badger because the node to be selected trust node in the future rounds. Next the RBFT and TPBFT performance is better than that of PBFT, FBFT, RBFT and TPBFT. However the proposed APBFT approach is provide the better network lifetime. This is lead to increase the network lifetime and save the energy of nodes in the WSN.

Figure 8 Through put (see online version for colours)



Here the proposed technique focuses on five parameters like 4nodes, 6nodes, 8nodes, 10 nodes and 12 nodes for calculating throughput of the consensus algorithm. However there are not at all possible to participate faulty node in consensus process of our proposed APBFT and total number through put is 14. The result is shown in figure 8. Our proposed APBFT algorithm is high throughput when compared to the existing PBFT consensus algorithm. The proposed Adaptive PBFT algorithm has been limited by the consensus process, so the number of packets delivered during the consensus process is reduced. Here, assume that the N numbers of nodes have participated in the voting process. The node has high trusted value t ($o < t <= 1$) that participated in the voting process. Then the number of nodes in the Master node is found as y ($1 <= y <= tN$). The size of the Master node $y = 1$, it degrades the PBFT process with the consensus node tN. The number of packets is $O((tN)^2)$. Then the value $t = 1$, the node participated in the voting process, the number of packets delivered $O(N^2)$. In this $O(N^2)$ is the highest number during the voting process. In the blockchain network, the communication complexity is reduced. The communication complexity $O(N^2)$) compared to PBFT is the same.

The BFT rate is $\frac{N-1}{3}$ in the existing algorithm of XFT. BFT of existing BFT consensus algorithms can deal with faulty nodes that work arbitrarily. In PBFT the BFT is in balance. Other consensus algorithms such as RBFT, FBFT is same BFT as $\frac{N-1}{3}$. Honey badger BFT is an extension of the PBFT consensus algorithm so the total communication complexity is the same as the PBFT consensus algorithm as $\frac{N-1}{3}$. Next the proposed technique has considered the XFT consensus algorithm, assuming that a BFT node cannot control faulty nodes and networks simultaneously. Here the BFT is $\frac{N-1}{2}$. Then evaluate the BFT of TPBFT is $\left(1-\frac{2}{3}d\right)N-\frac{1}{3}$. In this algorithm the primary node is replaced by the first group and the probability of view change process get reduced and the proposed Adaptive PBFT is $\frac{N-1}{3}$ BFT.

After that, many of the consensus algorithms based on the BFT consensus algorithm have been in low scalability. PBFT is not suitable for the extensive scale network, because of more number of nodes participating in the voting process. In the Adaptive

PBFT algorithm, the fault node is removed from the consensus process and node reputation is reduced. It reduces the number of faulty nodes participating in the consensus process. This method makes our APBFT consensus algorithm more suitable for a large scale blockchain network. So the APBFT, scalability is not reduced in an extensive scale network.

Our proposed APBFT consensus algorithm is based on the PBFT algorithm, which has a multistage consensus process. Generally, a proposed APBFT algorithm is different from a PBFT algorithm. Firstly, this is considering the high trust value node which is allowed to participate in the voting process. This method is used to improve the scalability and efficiency of the APBFT. Secondly, the method reduce the faulty node reputation in the voting process and also reduce the view change process as the Master node has been selected as the majority voting value. Here the equal chance is given to all trust nodes in the consensus process. So another node gets an opportunity to elect Master node. This will be reducing the consensus complexity. Here the Master node monitors the node activity and finds the faulty node and finds the faulty node transaction process. So the faulty node is not participating in the consensus process. This method avoids the process of view change.

Our proposed APBFT is compared with existing type of BFT algorithm such as PBFT, honey badger BFT, XFT, FBFT, RBFT, and TPBFT. Because the PBFT algorithm and RBFT algorithm are three-phase committing voting processes, their communication complexity is $O(N)^2$. FBFT simplifies PBFT, it easily adopts the primary node request and immediately sends the client request, their communication complexity is $O(N)^2$. The total communication complexity of honey badger BFT is $O(N^2 + N^3 \log N)$. The communication complexity of PBFT is the same as XFT, PAXOS, RAFT $O(N)^2$. Our proposed APBFT communication complexity is the same as PBFT $O(N)^2$. In BFT, existing BFT consensus algorithms can deal with faulty nodes that could work arbitrarily. In PBFT the BFT is $\frac{N-1}{3}$. Other consensus algorithms such as RBFT, FBFT is same as BFT is $\frac{N-1}{3}$. Honey badger BFT is an extension of the PBFT consensus algorithm so, the total communication complexity is the same as the PBFT consensus algorithm as $\frac{N-1}{3}$. Next, the proposed technique has considered the XFT consensus algorithm, assuming that a BFT node cannot control faulty nodes and networks simultaneously. Here the BFT is $\frac{N-1}{2}$. Then evaluate the BFT of TPBFT is $\left(1-\frac{2}{3}d\right)N-\frac{1}{3}$. In this algorithm the primary node is replaced by the primary group, and the probability of view change process gets reduced as the proposed Adaptive PBFT is $\frac{N-1}{3}$ BFT.

The proposed method of APBFT is has trust node, that node is only allow to participate in the voting process. At the same time the identified faulty nodes is removed or discard from the voting process. That proposed consensus algorithm is mainly used in permission blockchain network. APBFT algorithm is used to reach the node consensus very quickly and remove the faulty node easily. So the consensus overhead is reduced in blockchain network. The node lifetime is very important in WSN, IoT and blockchain network. The network is easily reach the consensus process and all the trust node actively

participated in the voting process, node energy is saved and time also reduced. Many of the existing PBFT algorithm, all the node is participated in the consensus process so consensus overhead increase. The proposed APBFT algorithm is allows only the trust node is participating in the consensus process so consensus overhead is reduced.

5 Conclusions

In this article, the proposed blockchain consensus algorithm named APBFT to improve the improvement of PBFT blockchain consensus algorithm. In our proposed APBFT consensus algorithm, we construct the Master Node selection based on the equal chance of the entire candidate node. This approach improves the efficiency of consensus and reduces the communication overhead. Then the finding has been carried out on the trust node and faulty node among candidates participated in the voting process based on their valid and invalid transaction. Here the proposed technique must reduce the view change probability of the consensus process. Further, high trust value node is allowed to participate in the voting process. Henceforth the identified faulty node removed from the consensus process. Finally, this APBFT consensus algorithm ensures the improvement of network performance in the presence of faulty nodes in the network. However, the proposed APBFT algorithm with the other existing type of BFT algorithm like PBFT, honey badger BFT, XFT, FBFT, RBFT, and T-PBFT which can improve the BFT and efficiency. Our proposed APFT algorithm can ensure the improvement of fault tolerance algorithm for the BFT type algorithm which will deploy in a blockchain network.

References

Alfandi, O., Otoum, S. and Jararweh, Y. (2020) 'Blockchain solution for IoT-based critical infrastructures: Byzantine fault tolerance', *NOMS EEE/IFIP Network Operations and Management Symposium*, pp.1–4.

Anisi, M.H., Abdullah, A.H., Razak, S.A. and Ngadi, M.A. (2012) 'An overview of data routing approaches for wireless sensor networks', *Sensors*, Vol. 12, No. 4, pp.3964–3996.

Baliga, A. (2017) *Understanding Blockchain Consensus Models* [online] https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf (accessed 5 February 2021).

Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S. and Danezis, G. (2017) 'Consensus in the age of blockchains', *arXiv Preprint*, arXiv: 1711.03936.

Biswas, S., Sharif, K., Li, F., Maharjan, S., Mohanty, S.P., Wang, Y. (2019) 'PoBT: a light weight consensus algorithm for scalable IoT business blockchain', *IEEE Internet of Things Journal*, Vol. 7, No. 3, pp.2343–2355.

Cachin, C., Vukolić, M. (2017) 'Blockchain consensus protocols in the wild', *arXiv Preprint*, arXiv: 1707.01873.

Cai, W., Jiang, W., Xie, K., Zhu, Y., Liu, Y., Shen, T. (2020) 'Dynamic reputation-based consensus mechanism: real-time transactions for energy blockchain', *International Journal of Distributed Sensor Networks*, Vol. 16, No. 3, pp.1–13.

Chatterjee, P., Ghosh, S.C. and Das, N. (2017) 'Load balanced coverage with graded node deployment in wireless sensor networks', *IEEE Transactions on Multi-Scale Computing Systems*, Vol. 3, No. 2, pp.100–112.

Deng, Q. (2019) 'Blockchain economical models, delegated proof of economic value and delegated adaptive Byzantine fault tolerance and their implementation in artificial intelligence blockcloud', *Journal of Risk and Financial Management*, Vol. 12, No. 4, pp.177–197.

Dinh, T.T.A., Liu, R. and Zhang M. (2018) 'Untangling blockchain: a data processing view of blockchain systems', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 7, pp.1366–1385.

Feng, L., Zhang, H., Chen, Y. and Lou, L. (2018) 'Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain', *Applied Sciences*, Vol. 8, No. 10, pp.1–21.

Gao, S., Yu, T., Zhu, J. and Cai, W. (2019) 'T-PBFT: an eigen trust-based practical Byzantine fault tolerance consensus algorithm', *China Communications*, Vol. 16, No. 12, pp.111–123.

Guan, X., Guan, L., Wang, X.G. and Ohtsuki, T. (2010) 'A new load balancing and data collection algorithm for energy saving in wireless sensor networks', *Telecommunication Systems*, Vol. 45, No. 4, pp.313–322.

Gueta, G.G., Abraham, I., Grossman, S., Malkhi, D., Pinkas, B., Reiter, M. and Tomescu, A. (2019) 'SBFT: a scalable and decentralized trust infrastructure', *9th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp.568–580.

He, L. and Hou, Z. (2019) 'An improvement of consensus fault tolerant algorithm applied to alliance chain', in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp.1–4.

Iyer, S., Thakur, S., Dixit, M., Agrawal, A., Katkam, R. and Kazi, F. (2019) 'Blockchain based distributed consensus for Byzantine fault tolerance in PMU network', in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp.1–7.

Kiki, A., Rahayu, W., Hara, T. and Taniar, D. (2020) 'Backup gateways for IoT mesh network using order-k hops Voronoi diagram', *World Wide Web*, Vol. 24, No. 3, pp.955–970.

Lamport, L., Shostak, R. and Pease, M. (2019) 'The Byzantine generals problem', in *Concurrency: The Works of Leslie Lamport*, pp.203–226, Association for Computing Machinery, New York, NY, USA.

Lao, L., Dai, X., Xiao, B. and Guo, S. (2020) 'G-PBFT: a location-based and scalable consensus protocol for IOT-blockchain applications', *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp.664–673.

Lei, K., Zhang, Q., Xu, L. and Qi, Z. (2018) 'Reputation-based Byzantine fault-tolerance for consortium blockchain', in *IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp.604–611.

Li, L. and Li, D. (2018) 'An energy-balanced routing protocol for a wireless sensor network', *Journal of Sensors*, Vol. 2018, pp.1–12, Article ID 8505616, https://doi.org/10.1155/2018/8505616.

Liu, S., Viotti, P., Cachin, C., Quéma, V. and Vukolić, M. (2016) 'XFT: practical fault tolerance beyond crashes', in *12th USENIX Symposium on Operating Systems Design and Implementation*, Vol. 16, pp.485–500.

Mazieres, D. (2015) 'The stellar consensus protocol: a federated model for internet-level consensus', *Stellar Development Foundation*, 32pp.

Miller, A., Xia, Y., Croman, K., Shi, E. and Song, D. (2016) 'The honey badger of BFT protocols', in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp.31–42.

Nakamoto, S. (2009) *Bitcoin: A Peer-to-Peer Electronic Cash System* [online] https://bitcoin.org/bitcoin.pdf/ (accessed 21 February 2021).

Puthal, D., Mohanty, S.P., Yanambaka, V.P. and Kougianos, E. (2020) 'Poah: a novel consensus algorithm for fast scalable private blockchain for large-scale IoT frameworks', *arXiv Preprint*, arXiv: 2001.07297.

Ray, P.P., Dash, D., Salah, K. and Kumar, N. (2020) 'Blockchain for IoT-based healthcare: background, consensus, platforms, and use cases', *IEEE Systems Journal*, Vol. 15, No. 1, pp.85–94.

Sankar, L.S., Sindhu, M. and Sethumadhavan, M. (2017) 'Survey of consensus protocols on blockchain applications', *4th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp.1–5.

Wang, Y., Cai, S., Lin, C., Chen, Z., Wang, T., Gao, Z. and Zhou, C. (2019) 'Study of blockchains's consensus mechanism based on credit', *IEEE Access*, Vol. 7, No. 1, pp.10224–10231.