NITG Chain: A Scalable, Private and Permissioned Blockchain with Proof of Reputation Consensus Method

Alok Jaiswal, Sheetal Chandel, Ajit Muzumdar, Chirag Modi, Madhu G. M., and C. Vyjayanthi

1 Introduction

Blockchain technology has grown for various IT applications. It can be defined as the append only distributed database that is practically immutable, maintained by decentralized P2P network using consensus method, cryptography and back referencing blocks to order and validate the transactions [1]. In contrast to traditional databases, blockchain offers data immutability, data transparency, user anonymity, trust among the untrusted entities, decentralization etc. Bitcoin [2] is the first popular implementation of blockchain for financial application. Since then, other blockchains such as Ethereum, various Hyperledger project solutions, IOTA, Ripple, R3. Corda, Hashgraph etc. have been popularized. Blockchain is explored in various markets such as B2C, B2B, online trading, auctioning, energy, e-KYC etc. In addition, it is evolved by integrating smart contracts as business logic. As per the deployment and access scope, there are mainly three types of blockchains viz; public, private and consortium. Public blockchains are open to all nodes for read and write. The widely used public blockchains are Bitcoin and Ethereum. The public blockchains are more secure and fully decentralized as the transaction validation and confirmation is done by a large number of nodes in the network. However, scalability and blockchain forks are major issue [3]. In private blockchain, a node requires a permission to take

A. Jaiswal (✉) · S. Chandel · A. Muzumdar · C. Modi · M. G. M. · C. Vyjayanthi
National Institute of Technology Goa, Farmagudi, Ponda 403401, Goa, India
e-mail: ajitmuzumdar@nitgoa.ac.in

C. Modi
e-mail: cnmodi@nitgoa.ac.in

M. G. M.
e-mail: madhugm@nitgoa.ac.in

C. Vyjayanthi
e-mail: c.vyjayanthi@nitgoa.ac.in

U. P. Rao et al. (eds.), *Security, Privacy and Data Analytics*, Lecture Notes in Electrical Engineering 848, https://doi.org/10.1007/978-981-16-9089-1_1

part in the blockchain network. Here, a write permission is given to the authorized nodes only, whereas read permission may be public or restricted. Here, data security and trust are dependent on the credibility of the authorized nodes. In consortium blockchain, the data write operation is performed by a pre-selected set of nodes from multiple organizations, while read permission may be public or restricted. As like a private blockchain, the data security and trust are dependent on the credibility of the selected nodes.

Although blockchain has high potential in various applications, it faces the problem of scalability due to the underlying computationally expensive consensus methods such as Poof of Work (PoW), Proof of Stake (PoS) etc. [4]. With the increasing number of nodes, the transaction throughput is decreasing as a large number of nodes are involved in transaction validation and confirmation. Although these consensus methods help in improving the trust among the untrusted entities in the blockchain network, they pose the scalability issue. In literature, researchers have proposed on-chain and off-chain solutions [5, 6] to address the scalability issue in blockchain. However, still there is a room for further improving the scalability and security of the blockchain.

In this paper, we design NITG Chain (NITGoa Chain) with the improved scalability and security. It is a private and permissioned blockchain which can be more suitable for business to business applications such as trading, asset management etc. Here, a node needs a permission from the existing nodes to participate in the blockchain network. It applies Proof of Reputation (PoR) consensus method for appending the block in NITG Chain. Here, the block mining is performed by only the dedicated nodes (authorized nodes) which have high reputation and selected by each organization. A block created by any authorized node is verified by other authority nodes along with underlying transactions, and that block is appended to NITG Chain after receiving and verifying the confirmation from 2/3 of authority nodes. The selection of authority node for block creation is done based on its liveliness and reputation. Thus, NITG Chain achieves fairness among the nodes for block creation. In addition, it ensures reliability, security from blockchain fork and affordable throughput and scalability. The functional and performance validation of NITG Chain is done using a testbed at NIT Goa by applying different size of transactions and blocks. From the experimental results, it is observed that NITG Chain achieves throughput 712 *tps* on average, while generating a block of transaction at an interval of 10 s.

In following, Sect. 2 discusses background on blockchain followed by the existing solutions to improve the scalability of blockchain. A detailed discussion of the NITG Chain is given in Sect. 3, while the experimental results and analysis of the NITG Chain is given in Sect. 4. Section 5 concludes our work with references at the end.

2 Background and Related Work

A common workflow of blockchain in peer to peer network is depicted in Fig. 1. Here, a transaction initiator broadcasts a transaction into peer to peer network of blockchain after digitally signing it. After receiving a number of transactions, each node starts creating a block of transactions by applying Proof of Work (PoW) [2]. A node which is successful of mining a block broadcasts that block in peer to peer network for the validation. After receiving the confirmation from a majority of nodes (51% nodes), each node updates their copy of the blockchain by appending that block.

In contracts to distributed database, blockchain establishes the trust among the nodes through consensus. However, scalability and fault tolerance of blockchains depend on underlying consensus method. A summary of the existing consensus methods is given in Table 1.

To address the scalability problem in blockchain, different solutions such as on-chain and off-chain including side chain and child-chain have been reported. On-chain solutions attempt to increase the block size, reduce the transaction size or sharding to improve the scalability. Big block and Bitcoin Unlimited [18] are the examples of the increased block size. Here, more number of transactions can be confirmed in a single run of the blockchain update. However, the block propagation speed decreases, which may result into blockchain forking. In Bitcoin's Segregated Witness (Segwit) [19], the signatures and transactions are separated. The witness data structure stores the signatures. This helps to address the problem of transaction malleability. Here, the size of transactions is reduced to increase the throughput. As like in the distributed databases, sharding in blockchain attempts to group nodes into different shards, and thus, allowing parallel processing of the transactions. Elastico [20], OmniLedger [21], RapidChain [22], Zilliqa [23], Harmony [24] and Monoxide [25] are applying the blockchain sharding. However, sharding causes the data integrity issue, if an attacker can have control over shards.

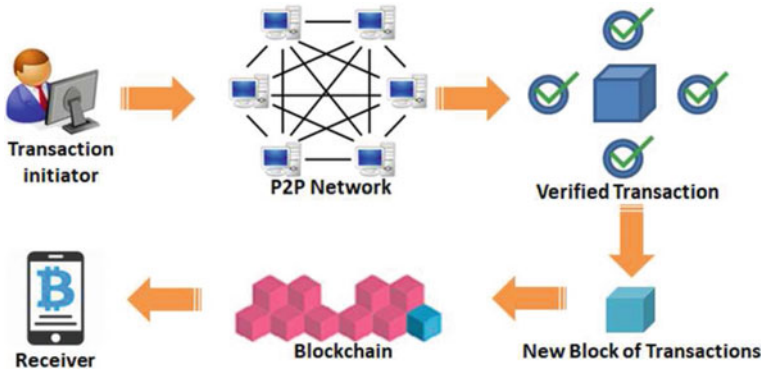

Fig. 1 A common workflow of blockchain [7]

Table 1 Summary of the existing consensus methods

Consensus	Applicable blockchain/DLT	Node identity	Block mining	Scalability	Fault tolerance
PoW [2]	Bitcoin	Permissionless	R	Low	50%
PoS [8]	Ethereum	Permissionless	D	Medium	$3f + 1$
PBFT [9]	Hyperledger	Permissioned	R	Medium	$3f + 1$
DPoS [10]	Bitshares	Permissioned	D	High	$3f + 1$ to $2f + 1$
FBA [11]	Ripple and Stellar	Permissioned	D	High	$5f + 1$ to $3f + 1$
PoAu [12]	None	Permissioned	D	High	$3f + 1$
PoET [13]	Intel's Sawtooth	Permissionless	R	High	$2c + 1$
PoAc [14]	None	Permissionless	R	Low	50%
PoB [15]	None	Permissionless	R	Low	50%
PoC [16]	Zcash	Permissionless	R	Low	50%
PoR [17]	None	Permissioned	D	High	$3f + 1$

R Random, *D* Deterministic, c number of nodes, f Byzantine faults

In off-chain solutions, the transaction processing is performed outside the chain for the frequent transactions. Typical solutions under this category are payment channels [26, 27], side chain [28] and Child chain [6]. Bitcoin's Lightning network [26] and Ethereum's Raiden network [27] have adopted the payment channel (off-chain) to process frequent transactions, and thus, reducing the number of transactions to be processed at main chain. This helps to improve the throughput and reduce the transaction fees. However, it may affect the ecosystem due to the reduced transaction fees and profit to miners. The goal of side chain is to transfer the cryptocurrencies among different blockchains. For example, the exchange of cryptocurrencies among different blockchains. The typical example of side chain is Pegged Sidechain [28]. Child chain follows the parent-child structure, in which transaction processing is performed at child chain, while parent chain maintains the record of the confirmed transactions. Plasma [6] follows the parent-child structure for transaction processing and record maintenance. A summary of the existing solutions to improve blockchain scalability is given in Table 2. As per our observation, there is a need of re-investigating or extending the existing consensus methods (on-chain solutions) to improve the throughput, scalability and security in blockchain.

3 NITG Chain: Proposed Blockchain Framework

3.1 Objective and Design Goals

The main objective is to design a scalable and secured private blockchain with the improved throughput for business to business applications. The proposed framework should achieve higher throughput in context of different size of transactions and block.

Table 2 Summary of the existing solutions to blockchain scalability

Category	Solutions	Advantages	Limitations
On-Chain	Increasing the block size [5, 18]	Improved throughput	Blockchain forking
	Reducing the transaction size [19]	Improved throughput	Causes fungibility
	Sharding [20–25]	Parallel processing	1% attack
Off-Chain	Payment channel [26, 27]	Reduced transaction fee and waiting time	Can affect the token ecosystems
	Side chain [28]	Allows to exchange cryptocurrencies among different chains	Difficult to control different cryptocurrencies due to their price differences
	Child chain [6]	Parent-child chain structure to improve the throughput	Complexity increases for parent-child verification

3.2 Design of the NITG Chain Framework

The design of NITG Chain is given in Fig. 2. Here, the created transactions are transmitted to all the peers (i.e. connected nodes to the transaction initiator) in peer-to-peer network. Further, these peers transmit such transactions to their peers and thus, all the created transactions are propagated in whole network for verification. Each node in P2P network has a list of all the verified transactions which are created in a particular time. The verification of transactions helps to remove any malicious transaction. In NITG Chain, only authorized nodes verify and validate the transactions. At a time, only one authorized node creates a block of such verified transactions and checks for other live authority nodes to make an active list. The active list of the authorized nodes is sorted as per their reputation score calculated over the period of time and appended to the block. The generated block by an authorized node is signed using the elliptic curve cryptography based digital signature algorithm. This generated block is broadcasted in the network for the verification. Block verification is performed by each authorized node in order to introduce valid block in the network. Upon verification, each authorized node broadcasts the signed confirmation transaction containing the block's hash value. Then, this block is added to blockchain by each node after receiving the confirmation from 2/3 of authority nodes. For the next block creation, the next authorized node is selected from the active list. Thus, all the authorized nodes have a fair chance of mining the block. In general, NITG Chain performs new node insertion, transaction flow, transaction verification, creating an active list of the authorized nodes, block creation, verification, and confirmation, time synchronization and API service. In following, a detailed discussion on these activities is given.

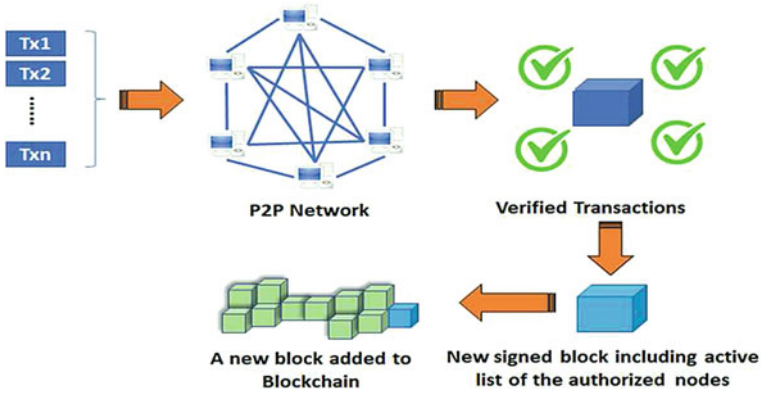

Fig. 2 Design of the NITG Chain framework

3.2.1 New Node Insertion in Network

When a new node wishes to participate in NITG chain, it broadcasts its public key to all the peers. A new node insertion in NITG chain is performed through 5-way handshaking and mutual authentication. A new node first sends a request to the existing node for adding it to the existing node's peer list, as shown in Fig. 3. In response, the existing node adds this node to its peer list and returns a node ID. Further, this new node sends a request for getting the existing node's peer list. The existing node sends its peer list to the new node. New node adds the returned peers to its peer list. Finally, a new node becomes a part of the network. For a new node insertion, following messages are involved.

1. $N \rightarrow P$: *na* //Request to add in peer list
2. $P \rightarrow N$: $(na, nb, ID)E_{kP}$ //Return Node ID (Existing node adds this new node to its peer list and returns a node ID)
3. $N \rightarrow P$: $(na, nb)E_{kN}$ //Request for peer list (New node sends request for the existing node's peer list)
4. $P \rightarrow N$: $(na, nb, PL)E_{kP}$ //Returns peer list
5. $N \rightarrow P$: $(nb, PL)E_{kN}$ //Add peers to peer list (New node adds the returned peers to its peer list and acknowledges the same)

Here, N is a new node and P is the existing peer node. *na* and *nb* are nonces of new node and peer node respectively. E_{kN} and E_{kP} are the private keys of new node and peer node respectively for signing the messages using elliptic curve cryptography. After 5-way handshaking process, all the peers add mapping of new node ID to its public key to their local cache. All further communication with this new node uses this public key.

3.2.2 Transaction Flow and Verification

The transactions created by the users are transmitted to all the nodes in the network. For this, we are applying Breadth First Search (BFS), where a node creates a transaction, it is transmitted to all its peers. Further, those peer nodes transmit that transaction to their peers as maintained in their peer list and so on. At the end, all nodes in the network receive the created transaction. Like this, each node in the network has a list of all the transactions created during time interval *t*. However, there may be malicious nodes which can spread the forged transactions into the network. To prevent this, the propagated transactions are verified before packaging them into the block. In NITG Chain, authorized nodes verify the transaction by checking its attributes, digital signature of sender, balance and size of the data field. Such verified transactions are collected as valid transactions for block creation.

3.2.3 Active List of the Authorized and Reputed Nodes

Each authorized node maintains a list of authorized nodes along with their reputation score calculated during time interval *T*. To know whether the authorized nodes are active or not, the current authorized node sends a ping request message to all other authorized nodes, as shown in Fig. 4. If it gets reply from a list of authority node, it adds that node to the active list. At the end, all the available authority nodes are identified for next block creations, as shown in Fig. 5. Consider, one of the authorized nodes is not available or crashed to send a reply. In this case, a crashed node will not be added to the active list of the authorized node. Thus, the next authorized and reputed node (potential miner) waits for a fixed time slice (2 s) and then it creates

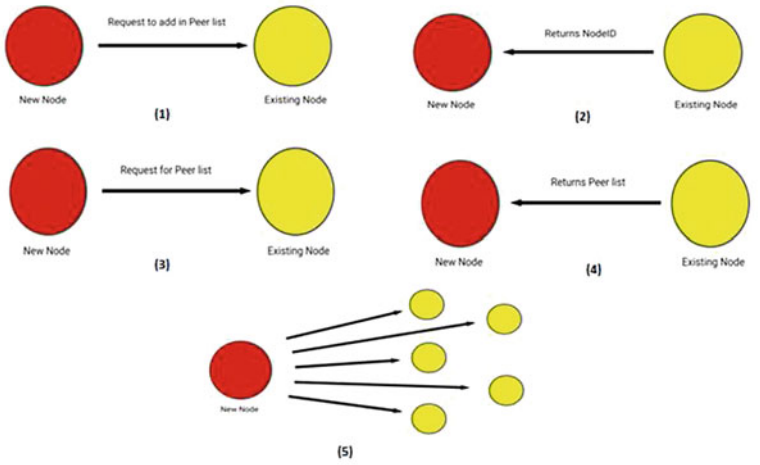

Fig. 3 New node insertion process in NITG Chain

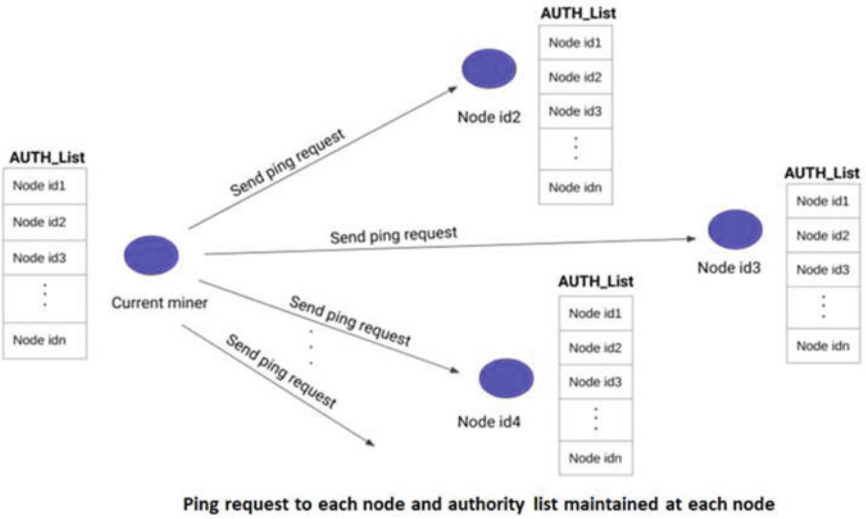

Fig. 4 Checking liveliness of the authorized nodes in NITG Chain

the block itself. This helps to achieve the reliability of the network and fast creation of block. For each of the authority nodes which have not participated in creating the recent *m* blocks, the reputation score is calculated based on their liveliness and participation in creating the recent blocks. Thus, reputation score of each node is determined as the number of times a node has generated valid blocks with respect to a chance given to that node for block creation. For this, an authorized node traverse the blockchain. Like this, all the authority nodes maintain a list of the other authority nodes and their reputation score and keeps updating it.

3.2.4 Block Creation, Verification and Confirmation

An authorized node verifies the transactions and collects them as valid transactions during the time *t*, as discussed earlier. It then creates a block (please refer Fig. 6) with the active list of the authorized nodes with their reputation score, as discussed earlier. Here, all the authority nodes are sorted and added to the active list of the authority nodes. This active list along with the reputation score of each node is appended in the current block so that the next authority node (potential node to create the next block) can be found from the current active list. This helps to secure the reputation score. Therefore, the active list determines the next miner automatically. If an immediate node crashes, the next potential miner waits for a fixed time slice (2 s) and then it creates the block itself and thus, making the network reliable, more decentralized and solving the problem of monopoly attack. It then signs the block using its private

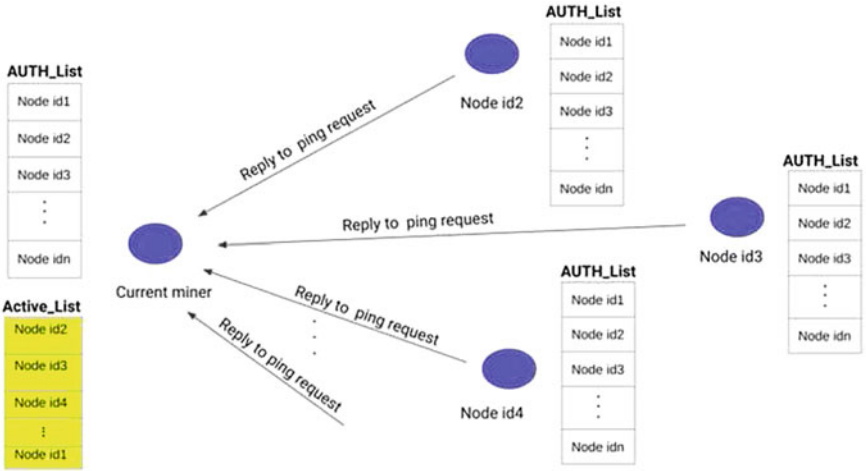

Fig. 5 Creating an active list of the authorized and reputed nodes in NITG Chain

key and appends signature to the block. Finally, the created block is broadcasted for the verification.

Block verification is done by each authority node in order to introduce valid block in the network. Here, each authority node checks the presence of the block creator in the authority list. If present, then the signature of that node is verified. If the creator node is an authority node, the signature of each transaction using the signer's public key is verified. Then, each authority node creates a confirmation transaction

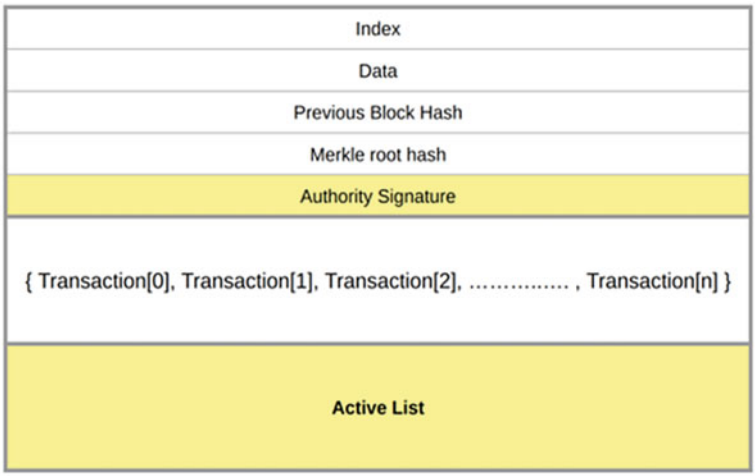

Fig. 6 Block format in NITG Chain

containing the hash value of the created block and signs it using its private key. Such transaction is broadcasted into the network as a confirmation of the block.

Block confirmation is done by authority nodes in the network in order to append the created block to their copy of the chain. Each node waits for the confirmation from 2/3 of the authority nodes. If confirmation from 2/3 of the authority nodes are received, the confirmation transaction's fields viz; signature and hash of the block are verified. If signature and block hash is valid for the confirmation transaction from 2/3 of authority nodes, that block is appended to the node's chain. Thus, to add a block in NITG Chain, a confirmation from at least 2/3 of authority nodes is required. This helps to reduce the verification cost as each node in the network do not require to verify the block.

3.2.5 Time Synchronized Block Generation

After each fixed duration, a new block is created automatically and added to the chain. This new block contains all the transactions which are committed in particular time *t*. In NITG Chain, we have kept a time slice of 10 s for the next block generation and to append it to the chain.

3.2.6 API Service to Interact with NITG Chain

Many low storage devices like IOT sensors, mobile phones, etc. are unable to interact with the blockchain due to their storage and computing limit. We have developed an API service through which such devices can interact with the NITG Chain. To build an API, we have used Express.js and Node.js. The network communication is enabled through protocols like HTTPS/HTTP.

4 Experimental Results and Analysis

4.1 Experimental Setup

For the functional validation of the NITG Chain, a small-scale testbed is created at NIT Goa by setting up three servers as shown in Fig. 7. We have considered three validator nodes such as VM1.1, VM2.1 and VM3.1 as authority nodes. The functionalities of NITG Chain are written in Python.

For the experimental evaluation of the proposed NITG Chain, the peer itself is considered as CA to generate its own public key. Here, transaction certificates are not provided with the assumption that all the nodes are having the valid keys. In future, the role of CA can be implemented for the real time deployment. Each node generates a preloaded transaction (please refer Fig. 8). In Fig. 8, "vk" is the public key of the

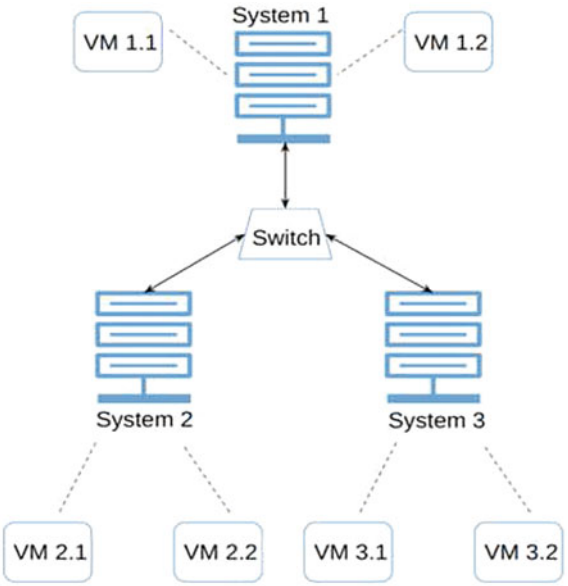

Fig. 7 Experimental setup for functional and performance validation of NITG Chain

{
"nonce":0,
"from_account":'7e2e44bddbd60e50d53f328d61d2f731058851ce9c06e34bd35ea3b77b0a39f2',
"to_account":'fbfeec63c97591f8bf182a95d8a914e20dc9691dff4e59c8060f705166bd99a9',
"amount": 1233,
"vk": 'c46064f3ef5327e74a3ac514443503fbac5e50437f2f046eeea17a4d2558138c43e919a3ea5815ea254483c1eccfec6b44b9569586fd4d038eb15a50867dec3d',
"data": '"a":"sdfsdfsdf","b":"sdfsdf"',
"sig":'38757f8cff5554949a83cd3d9df0ab91f447bfb6ef800d6257cdca00bc259ee68b1193a64e8df223645385e9f21de7196a475b6e3e19704e27c1a7ebe5e8d7aa',
"timestamp":1591549899.9142396,
}

Fig. 8 Pre-loaded transaction on each node

sender, "sig" is the signature of the transaction signed by a sender, "form account" is the id of the sender and "to account" is the receiver's id. The transactions are automatically generated using a script running on all the nodes. For the performance evaluation, NITG Chain is evaluated in terms of throughput on different size of transactions and blocks.

4.2 Results and Analysis

During performance evaluation, the transactions are automatically generated by all the nodes and the authority nodes have followed the defined consensus rules to create the block in the time interval of 10 s. We have applied different size of the transactions with different payload sizes (1–4096 bytes) and evaluated the throughput of NITG Chain. Figure 9 shows the transaction throughput of NITG Chain with varying size of the payload. From the results, it is observed that the NITG Chain confirms more than 690 transactions per second, if the transaction payload size is up to 1000 bytes.

In addition, a range of blocks with varying number of transactions are applied to NITG Chain to test its performance. Such blocks are sent in the network by a randomly selected node. The performance results of NITG Chain in the context of different size of the blocks are shown in Fig. 10. It is observed that NITG Chain has throughput more than 545 transactions per second, while adding up to 500 transactions per block.

Although performance results of NITG Chain are derived in a limited resource environment, these results are more encouraging to use NITG Chain in large scale network as it requires less computation for performing the transactions. As per our observation, NITG Chain addresses the problems of blockchain forks and scalability at an affordable level. In NITG Chain, a block created by an authority node is verified by other authority nodes and it requires confirmation from at least 2/3 of other authority nodes to append it in blockchain. There is no chance of having another

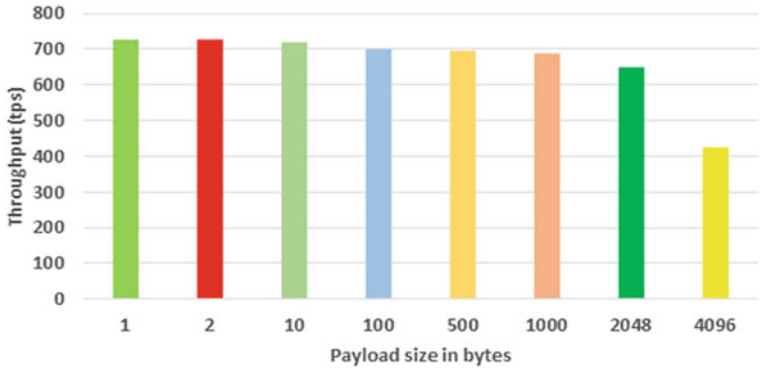

Fig. 9 Throughput of NITG Chain with different payload size (in bytes)

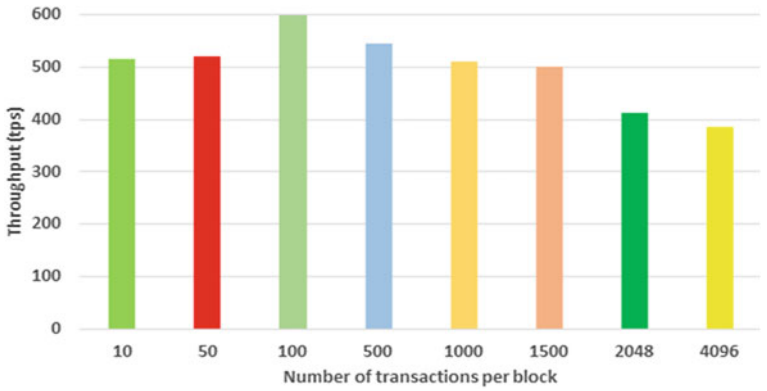

Fig. 10 Throughput of NITG Chain with different size of block

Table 3 Comparison of NITG Chain with the existing well-known blockchains

Platform/type	Consensus method	Smart contract support	Anonymity	Average throughput (tps)	Block generation interval
Bitcoin [2]/Permissionless	PoW	No	Yes	3.5	10 min
Ethereum [8]/Permissionless or Permisioned	PoS/PoW	Yes	No	15	20–30 s
Zcash [16]/Pemissionless	PoW	No	Yes	23	1.25 min
Ripple [29]/Pemissionless	RPCSA	No	No	1300	3.84 s
Hyperledger [9]/Permisioned	BFT	Yes	No	3500	Customizable
Litecoin [30]/Permissionless	PoW	No	No	56	2.5 min
NITG Chain/Permissioned	PoR	Under development	Yes	712	10 s

confirmed block in the network at same time as each authority node signs only one block at a time, and thus, it addresses the problem of blockchain forks. It assumes that at least 2/3 of reputed authority nodes behave honestly. Table 3 shows a comparison of NITG Chain with the existing well-known blockchains.

5 Conclusions

Blockchain has high potential in future IT based domains. In this paper, we have designed and implemented NITG Chain, a scalable, private and permissioned blockchain for business to business applications. It applies a proof of reputation consensus, in which the liveliness of authority nodes and their reputation are considered to select an authority node for block creation. In addition, each authority node gets a fair chance of mining the block. This helps to achieve the reliability of the network. The verification of block, underlying transactions and confirmation transaction from 2/3 of authority nodes helps to address the problem of blockchain fork as well as scalability. From the experimental results, it is observed that NITG Chain has throughput 712 *tps* on average, while applying different size of payload and blocks. The experimental results are very encouraging to use NITG Chain for large scale business to business applications, in which the authority node can be selected from each of the participating organizations. In future, NITG Chain will be further investigated to support the smart contracts and to offer on-chain data privacy.

Acknowledgements This work is part of a research project titled "Developing Smart Controller for Optimum Utilization of Energy and Trustworthy Management in a Micro Grid Environment (IMP/2019/000251)" with funding support under IMPacting Research INnovation and Technlogy-2C.1 (IMPRINT-2C.1) by Science and Engineering Research Board, Department of Science and Technology, Government of India.

References

1. Muzumdar A, Modi C, Madhu G, Vyjayanthi C (2021) A trustworthy and incentivized smart grid energy trading framework using distributed ledger and smart contracts. J Netw Comput Appl 103(074):183–184
2. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf
3. Conti M, Sandeep Kumar E, Lal C, Ruj S (2018) A survey on security and privacy issues of bitcoin. IEEE Commun Surv Tutor 20(4):3416–3452
4. Andoni M, Robu V, Flynn D, Abram S, Geach D, Jenkins D, McCallum P, Peacock A (2019) Blockchain technology in the energy sector: a systematic review of challenges and opportunities. Renew Sustain Energy Rev 100:143–174
5. Jeff G (2015) Making decentralized economic policy. http://gtf.org/garzik/bitcoin/BIP100-blocksizechangeproposal.pdf
6. Poon J, Buterin V (2017) Plasma: scalable autonomous smart contracts. https://www.plasma.io/plasma.pdf
7. Swathi P, Modi C, Patel D (2019) Preventing sybil attack in blockchain using distributed behavior monitoring of miners. In: 10th international conference on computing, communication and networking technologies, pp 1–6
8. Proof-of-stake (pos) (2021). https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/
9. Hyperledger architecture, vol 1 (2017). https://www.hyperledger.org
10. Delegated proof of stake (dpos) (2021). https://how.bitshares.works/en/master/technology/dpos.html

11. Mazieres D (2016) The stellar consensus protocol: a federated model for internet-level consensus. https://www.stellar.org/papers/stellar-consensus-protocol
12. Proof of authority (2021). https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains
13. Poet 1.0 specification (2021). https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html
14. Bentov I, Lee C, Mizrahi A, Rosenfeld M (2014) Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract]y. SIGMETRICS Perform Eval Rev 42(3):34–37
15. Karantias K, Kiayias A, Zindros D (2019) Proof-of-burn. https://eprint.iacr.org/2019/1096.pdf
16. Hopwood D, Bowe S, Hornby T, Wilcox N (2021) Zcash protocol specification. https://zips.z.cash/protocol/protocol.pdf
17. Zhuang Q, Liu Y, Chen L, Ai Z (2019) Proof of reputation: a reputation-based consensus protocol for blockchain based systems. In: Proceedings of the 2019 international electronics communication conference, IECC '19, pp 131–138
18. Bitcoin unlimited (2021). https://www.bitcoinunlimited.info/
19. Eric L, Johnson L, Pieter W (2015) Segregated witness. https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
20. Luu L, Narayanan V, Zheng C, Baweja K, Gilbert S, Saxena P (2016) A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, CCS '16, pp 17–30
21. Kokoris-Kogias E, Jovanovic P, Gasser L, Gailly N, Syta E, Ford B (2018) Omniledger: a secure, scale-out, decentralized ledger via sharding. In: 2018 IEEE symposium on security and privacy (SP), pp 583–598
22. Zamani M, Movahedi M, Raykova M (2018) Rapidchain: scaling blockchain via full sharding. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, CCS '18, pp 931–948
23. Team Z (2017) The zilliqa technical whitepaper. https://docs.zilliqa.com/whitepaper.pdf
24. Harmony (2021). https://www.harmony.one/
25. Wang J, Wang H (2019) Monoxide: scale out blockchains with asynchronous consensus zones. In: 16th USENIX symposium on networked systems design and implementation (NSDI 19), pp 95–112
26. Poon J, Dryja T (2016) The bitcoin lightning network: scalable offchain instant payments, draft version 0.5. https://lightning.network/lightning-network-paper.pdf
27. Raiden network (2021). https://raiden.network/
28. Back A, Corallo M, Dashjr L, Friedenbach M, Maxwell G, Miller A, Poelstra A, Timón J, Wuille P (2014) Enabling blockchain innovations with pegged sidechains. https://blockstream.com/sidechains.pdf
29. Ripplenet (2021). https://ripple.com/
30. Litecoin (2021). https://litecoin.com/en/