Comparison of proof-of-work based blockchains against federated consensus and proof-of-validation based blockchains

ambili_p180002cs@nitc.ac.in , jimmy@nitc.ac.in

October 2019

Abstract

This paper reports the results of survey done on the architecture and functionalities involved in blockchains. Moreover, it reports the results of comparison between proof-of-work-based blockchains, Bitcoin and Ethereum, against federated consensus-based blockchain, Ripple, and proof-of-validation-based blockchain, Tendermint, along the parameters like peer to peer network setup and maintenance, cryptocurrency involved, details of transaction execution and validation, block creation, block validation and consensus protocol and application development.

1 Introduction

Centralized architecture stores all major parts required to run a software system at a central location [1]. Most of the existing IT applications run on centralized architecture [2] wherein data is stored in central database. These databases have characteristics which enable features like core banking [3]. The importance of storing in databases is easy update and retrieval. The loss of a database will result in several issues like loss of customer data, application unavailability and inconsistency of data compared to pre-existing state. Therefore, backups of databases are stored in multiple geographic areas/locations. The backups are uniformed and brings in the concept of distributed systems.

Distributed systems are implemented by many techniques. The architecture built over peer to peer network is a highly successful and powerful one. In this method, the participants get equal priority and arrive at common decision through the use of consensus algorithms. Variations of conventional consensus algorithms are being used by various service providers to satisfy specific needs.

The trust is maintained in centralized architecture by relying on one or more intermediaries as in a hierarchy of digital certificates. This dependency sets stage for issues like double spending, forgery of transaction, reversal of transaction and censorship of transaction [4].

There is a solution to reduce dependency on third parties and remove the associated issues - distributed ledger. If the distributed ledger is strong under cryptography, forgery becomes almost impossible. Depending on the application use cases, reversal and censorship of transaction can be done by setting appropriate priority.

One such distributed ledger that is built using cryptographic primitives is blockchain. The data structure involved in blockchain is a chain of blocks which grows in the forward direction only. It is technically the back end database that maintains a distributed system openly [5]. Transaction data is stored in databases by each participating full node. Each block is linked strongly using cryptographic techniques with the previous one and contains data of all transactions within a period of time. The data in centralized architecture may be updated and

maintained by commit protocols. In contrast, integrity of data in blockchain is based on consensus on computation.

2 Background

Blockchain based on proof of work, Bitcoin, first came up as the backbone of cryptocurrency [6]. It involves a transaction validation mechanism which does not require intermediary assistance. It has zero downtime and is irreversible. Due to its decentralized nature and admissible anonymity, the transaction fee involved while transferring currency or item of value is very low. The ledger is public and hence ensures transparency. The entire blockchain can be traversed and every single transaction ever made can be traced.

The blockchain may be applied over existing web application or as a separate private application. For practical purposes, blockchains may be implemented over private networks or the Internet, as private standalone applications or hybrid applications. The categorization depends on two criteria - authorization and access control.

Authorization classifies the blockchain as permissioned and permissionless [7]. Permissioned blockchain provide special privileges to specific nodes and permission less blockchain is absolutely anonymous wherein anybody can step in and participate at any time. Access control describes access to blockchain data itself as public or private. Often, in reality, permissionless blockchains are implemented as public and permissioned ones as private.

To map the real world items on to the blockchain, relevant measurements and rules are to be determined and then embedded. The item then becomes a smart property and deal can be determined using the smart contract [8]. There is an active attempt to adopt blockchain based architecture in various domains.

This paper provides a comparison of the four blockchains under consideration, that is, Bitcoin, Ethereum, Ripple and Tendermint. The parameters considered include peer to peer network setup and maintenance, cryptocurrency involved, details of transaction execution and validation, block creation, block validation and consensus protocol and application development. These are elaborated in section 3.

3 Comparison

3.1 Peer to peer (P2P) network setup and maintenance

Blockchains run over peer to peer network. There are no privileged nodes. The nodes interact in a mesh network with a flat topology. Bitcoin is the largest and most successful application of P2P technologies. The second largest beneficiary

of P2P is file sharing applications like Napster[9] and BitTorrent[10].

Peer to peer network is built as an overlay over existing TCP/IP network. The messages are transferred in existing formats like Google Protocol buffers [11] over the network. Ethereum and Bitcoin use UDP (User Datagram Protocol) for transferring messages. Ripple network uses multicast IP. Tendermint uses UDP.

Peer discovery is done in Bitcoin with DNSseeds. These are DNS servers that return addresses of full nodes on the Bitcoin network and helps in finding peers. Any node may be selected at random for connectivity. Th extended Bitcoin network has different kinds of nodes like full nodes, simplified payment verification nodes, miners and mining pools. Messages are broadcast over the network using P2P protocols like bitcoinj [12]. Stratum [13] is an algorithm used at mining node.

Ethereum uses a distance based approach to maintain the network structure [14]. Kadelmia distributed hash table [15] is used to calculate the distances. Initialization is done by sending query to the bootstrap node which provides connection information for regular nodes to connect to each other. Whisper decentralised communication protocol is used to transfer messages.

In Ripple network, messaging is based on multicast IP [16]. Gnutella based approach is followed [17]. The roles of peers are of leaf, or Superpeer, and clienthandler. Each peer maintains multiple outgoing connections and optimal incoming connections to other peers. Network defines a fully connected directed graph of nodes. Bootstrapping is done by learning from configuration files, domain name lookups as well as messages received from overlay. Each participating TCP/IP socket occupies a slot. Slots have state and properties associated with it. Three algorithms used are fixed slot, live cache and bootcache.

Tendermint uses peer exchange protocol (PeX) [18]. There are seeds as well as persistent peers in the system. The seed nodes in Tendermint network reply to initial queries of participating nodes and then disconnect. The other participating nodes remain as persistent peers. All peers relay messages to other peers they know by default.

3.1.1 Comparison

All the four blockchains use UDP in the broadcast messages. The difference lies in the protocol used to achieve the same.

3.2 Cryptocurrency

An electronic coin is defined as a chain of digital signatures in [6]. Each owner transfers the coin, that is the bitcoin (BTC), to the next by digitally signing a

hash of the previous transaction and public key of next owner. The cryptocurrency in Ethereum is ether (ETH).

The difference between ETH and BTC token generation is that BTC generation halves approximately every 4 years whereas ETH continues to be generated at a constant rate every year, perhaps only until the Serenity version [19]. Greedy Heaviest-Observed Sub-Tree (GHOST) protocol is used to determine valid blocks and reward miners. This is a lot more complicated than Bitcoin. The total ETH in existence is sum of pre-mine, block rewards, uncle rewards and uncle referencing rewards,which are described below.

Around 72 million ETH were created for the crowdsale in the middle of 2014. This is sometimes called a 'pre-mine'. It was decided that post-crowdsale, future ETH generation would be capped at 25 percent of that per year. That is, no more than 18m ETH could be mined per year, in addition to the one-off 72m ETH generated for the crowdsale.

Currently each block mined creates 5 fresh ETH. If a block is mined every 14 seconds, and there are 31.5m seconds in a year (365x24x60x60), 2.25m blocks are mined per year. 2.25m blocks at 5 ETH per block is 11.3m ETH per year. This meets the commitment of less than 18m ETH generated per year.

Some blocks are mined a little late and don't form part of the main blockchain. In Bitcoin these are called 'orphans' and are entirely discarded, but in Ethereum they are called 'uncles' and can be referenced by later blocks. If uncles are referenced as uncles by a later block, they create about seven-eighth of the full ETH reward, that is 4.375 ETH for the miner of the uncle. This is called the uncle reward. Currently around 500 uncles are created per day, adding an additional 2,000 ETH into circulation per day, which is approximately, 0.7m ETH per year at this rate. A miner may also refer an uncle and can get about 0.15 ETH per uncle.

XRP is the token of the Ripple network. It is not created on the fly or using any reward system as in Bitcoin and Ethereum. Ripple network started off with 100 billion XRP [16].

Initially, Tendermint had a built-in currency and bonds to keep the system active. It has now evolved into a general purpose blockchain and can accommodate code base of other systems like Ethereum [14].

3.2.1 Comparison

Bitcoin will create its cryptocurrency till 2140. Ethereum will keep creating new ether as long as the system is live. Ripple payment system has XRP token which was created at its inception. No more tokens can be added to it. Tendermint is a blockchain application oriented development model and currently does not

have an in-built cryptocurrency.

3.3 Data handling

In Bitcoin, participating entities are identified using addresses. It is the public key of the elliptic key pair (ECC) [21]. The corresponding private key is used for signing. The network is thus pseudonymous. Addresses may be modified to have characters of our choice in the suffix portion. Such addresses are called vanity addresses. Bitcoin signature verification is done through scripts. Turing incomplete language cannot compute all Turing computable functions [20]. Script is a Turing incomplete language used in bitcoin. It does not allow loops.

Transactions are hashed and stored in the binary hash tree or the Merkle tree. Only the root of the Merkle tree is stored in the block. Transaction data is stored in database maintained at the client side. Two types of transactions are coinbase and normal transactions. The first transaction in each block is called a coinbase transaction. It starts a new coin owned by the creator of the block. Satoshi Nakamoto coded Bitcoin to start with 50 BTC in genesis block. Other transactions are normal ones which can be used to transfer Bitcoins or other items of trade . A transaction depends on several transactions and those depend on many more. There is never the need to extract a complete standalone copy of a transaction's history. Miners receive their part from the transaction fee which is the difference in total input and output value. Unspent transaction outputs (UTXO's) handle the flow of electronic coin.

Ethereum also uses asymmetric key pair generated using ECC . Every account is represented by an address generated from public key. Transactions are used to transfer ether from an account to another or to a contract. Transaction is signed using ECDSA [22].

A transaction in Ethereum contains recipient of the message, a signature identifying the sender, the amount of ether to transfer, maximum number of computational steps the transaction execution is allowed to make, called the gas limit, and the cost the sender of the transaction is willing to pay for each computational step, called the gas price. The product of gas used and gas price is called transaction fee.

The transactions in Ethereum are called message call and contract creation. To send ether or to execute a contract method, the transaction is broadcast to the network. The recipient verifies and receives ether. Transaction verification is done independently by participating nodes.

Ethereum uses trie data structure to store state, transactions and receipts [16]. Transaction receipts are also stored in ethereum. This eliminates the need of UTXO's. Transactions between different Ethereum accounts move global state of Ethereum from one to the other. Apart from details of transaction, state and

transaction receipts are also stored in Ethereum blocks.

The states can be categorised as world state and machine state [16]. World state is a mapping between addresses (160-bit identifiers) and account states (a data structure serialised as RLP). The account state comprises of following four fields, nonce - scalar value equal to number of transactions sent from this address, balance - scalar value equal to number of Wei owned by this address, storageRoot and codeHash - hash of EVM code of this account which gets executed when this address receives a message call.

With the introduction of smart contracts, loops are supported in Ethereum. The languages used in blockchains which support smart contracts and other Turing computable functions are Turing complete.

A transaction is the only way to move data in Ripple's XRP ledger [23]. They enter the ledger only if signed and validated by node and accepted by consensus process. Pseudo transactions are permitted by the ledger and are generated as per ledger rules. Such transactions aren't signed or submitted but must still be accepted by consensus. Transactions that fail are also included in ledger because they modify balances of XRP to pay for anti spam transaction cost.

Ripple-API (ripple-lib) is the official client library to XRP ledger[43]. The basic types include address which is base58 encoding of hash of account's public key, account sequence number that is used to keep transactions in order, amount which may be expressed in native currency or XRP and transaction fee which defaults to XRP and is specified in RippleAPI constructor. Authentication to rippled server, certificate containing key of client, time-out etc. are few of the other parameters in the constructor which is a Node.js object. Methods are defined to handle transactions. Few of these can be executed offline like sign, generateAddress, preparePayment, prepareOrder etc.

Tendermint core is responsible for sharing blocks and transactions between nodes and establishing a canonical/immutable order of transactions. Tendermint application need to validate each transaction received with DeliverTx message against the current state, application protocol and cryptographic credentials of the transaction. A validated transaction then needs to update the application state by binding a value into key value store.

Accounts are given addresses as in other blockchains. Signature addition is done using the private key and verification using the public key of asymmetric key pair associated with each account.

Upon creation of a blockchain, Tendermint calls InitChain. From then on, the following sequence of methods are executed for each block -
BeginBlock, [DeliverTx....], EndBlock, Commit, where DeliverTx is called for each transaction and results in an updated application state.

Application blockchain interface (ABCI) provides a clean interface between state transition machines on one computer, represented as application logic, and the mechanics of their replication across multiple computers, represented using consensus engine. Application validates transaction.

ABCI design has few distinct components. Message protocol between application and consensus engine defined using protobuf. Consensus engine runs the client and application runs the server. Blockchain protocol between ABCI (connection oriented) and Tendermint Core are mempool connection, consensus connection and query connection. CheckTx method is used to connect to MemPool. Remote procedure calls without engaging consensus is used to connect to query connection.

3.3.1 Comparison

Ethereum and Bitcoin blockchains are stored in levelDB. Ripple and Tendermint uses RocksDb for storage. Ripple also provide other choices like HyperLevelDB, levelDB and SQLite. Tendermint uses C implementation of levelDB.

3.4 Block validation and consensus

Consensus algorithms [24] often arise in the context of replicated state machines. Each participating node or server compute identical copies of the same state. They should continue to operate even if some of them are down. Replicated state machines are used to arrive at a common state in distributed systems. Replicated state machines are typically implemented in a distributed system using a replicated log. Each server stores a log containing series of commands, which its state machine executes in order. The state machines are deterministic and hence each computes the same state and same sequence of outputs.

To achieve consensus in a distributed system, transaction logs are maintained by participating nodes. State of system is modified based on rules agreed upon and data in these transaction logs. The right to perform state transition is also distributed among participating nodes. There may be users who are given rights to collectively perform transitions through an algorithm. It should be securely decentralized. The result is that no single actor or group of actors can take up majority of the set.

Paxos algorithm [25] derived after a research of twenty years has almost become synonym for consensus algorithm in distributed systems. Different methods may be used to achieve consensus in blockchain like proof of work, proof of stake, delegated proof of stake, leader based consensus, federated consensus, proprietary distributed ledger, practical byzantine fault tolerance (PBFT) and derivatives and N2N [26]. Each of these basically tries to solve Byzantine generals' problem [26].

It is an agreement problem in which group of generals each commanding a portion of Byzantine army, encircle a city. These generals wish to come up with a plan for attacking a city. In its simplest form, generals must only decide whether to attack or retreat. All generals should agree on a common decision. The problem is complicated by the presence of traitors.

A fault may occur in the system presenting different symptoms to different users. This kind of fault is called the Byzantine fault. The loss of a system service due to a Byzantine fault is called Byzantine failure.

Any system built using blockchain should be Byzantine fault tolerant. It is possible for a system to perform reliably only when the number of traitors is less than one-third the total participating nodes. There are different ways through which consensus can be achieved.

Under proof of work, transactions are broadcast by the nodes. These are grouped together into a block and are added to the blockchain if the appropriate work can be exhibited by the miner by determining the answer to a very special mathematical puzzle. The so called miners use specialized hardware to run mining software and win a block. This includes block rewards and transaction fees. The other nodes accept the block only if all transactions in it are valid. It is expressed by including hash in the next block they create. The items of trade may be colored using colored coins [27] and transferred over the networks running on proof of work based blockchains. A successfully running example is the bitcoin. Few transactions are left uncolored for payment of transaction fee to the miner.

Proof of stake category of consensus algorithms takes the power of specific nodes known as validators to arrive at final agreement. Delegated proof of stake extends this with electing witnesses from the possible validators who will vote for blocks.

Federated consensus mechanism tries to arrive at a conclusion by picking opinion from overlapping subnets and converging them.

Proof of validation puts responsibility of validating transactions and forming the block on special nodes called validators. In Tendermint, the proof is included as a field called LastCommit in each block.

Bitcoin is the successful practical implementation of distributed ledger based on proof of work. Technology behind it is useful to move other systems to blockchain. But permissionless or discretionary systems may not help always. If domain of finance is considered, the validator cannot be pseudonymous or anonymous and know your customer (KYC) procedures need to be followed.

A distributed ledger is well suited for specific use case within financial industry but not as a complete replacement. Another drawback of proof of work based architecture is that power consumption is very high at the mining nodes. To overcome this, proof of stake and delegated proof of stake methodologies were put forward. But both of them have nothing at stake problem wherein validator behave maliciously and vote for unworthy blocks knowingly because they have nothing to lose for their faults. This has been avoided in Tendermint by designing penalizing techniques for misbehavior.

The method used to validate blocks and achieve consensus in the four blockchains under consideration is described below.

3.4.1 Bitcoin protocol

Consensus in Bitcoin follows a four step approach. It begins with independent verification of transactions by full nodes, based on a comprehensive list of criteria. A matching transaction must exist in the pool.

The second step is independent aggregation of the transactions into blocks by mining nodes. The grouped transactions are loaded to mining pool where they can be mined into a block.

The third step involves independent verification of new blocks by every mining node and assembly into a chain. The rules to be checked include syntactic validation of block data structure, the block header is less than the target difficulty, the block stamp is less than two hours in future, only the first transaction in a block is coinbase transaction and all transactions within the block are valid.

The fourth step involves independent selection by every node of the chain with the most cumulative computation demonstrated through proof of work. The main chain at any time is whichever chain of blocks has the most cumulative difficulty associated with it. Under most circumstances this is also the chain with the most blocks in it, unless there are two equal length chains and one has more proof-of-work. The main chain will also have branches with blocks that are siblings to the blocks of the main chain. These blocks are valid but not part of the main chain. They are kept for future reference, in case one of those chains is extended to exceed the main chain difficulty.

Consensus on the block mined is achieved using proof of work mining algorithm wherein the miner who solves the mathematical puzzle first by creating a block with hash value less than the target gets to include it in the main chain.

3.4.2 Ethereum protocol

Ethereum also uses proof-of-work based approach as of now. Efforts are on to develop a robust proof-of-stake method called Casper which will be used after

good amount of validation.

The protocol begins by validating ommers, wherein the uncle blocks which were left unmined in earlier rounds is first considered. The next step is to validate transactions. The miners are then given appropriate rewards. The application of rewards to a block involves raising the balance of accounts of the beneficiary address of the block by 3 ETH and each ommer by 1/32 of the block reward. The beneficiary of the ommer gets rewarded depending on the block number.Verification of the state and block nonce is the last step. The trie in the block is updated accordingly.

Ethash is the PoW algorithm for Ethereum 1.0 described in Ethereum yellow paper. There exists a seed which can be computed for each block by scanning through the block headers up until that point. From the seed, one can compute a pseudorandom cache, cacheinit bytes in initial size. Light clients store the cache. From the cache, we can generate a dataset, datasetinit bytes in initial size, with the property that each item in the dataset depends on only a small number of items from the cache. Full clients and miners store the dataset. The dataset grows linearly with time.

Mining involves grabbing random slices of the dataset and hashing them together. Verification can be done with low memory by using the cache to regenerate the specific pieces of the dataset that you need, so you only need to store the cache. The large dataset is updated once ev- ery epoch blocks, so the vast majority of a miner's effort will be reading the dataset, not making changes to it.

3.4.3 Ripple protocol

There are five components involved in Ripple [4, 18] protocol - servers which run Ripple server software , ledger which is the record of amount of currency in each users account , last closed ledger which is the most recent ledger ratified by the consensus process and thus represents current state of the network, open ledger which represents the current operating status of a node, Unique Node List (UNL) which is a list maintained by each server of other servers that it queries when determining consensus and proposer which is any server that tries to start the process. UNL is a list of public keys associated with validating nodes.

Ripple consensus algorithm proceeds in rounds. In each round, four steps occur. Initially, each server takes all valid transactions it has seen prior to beginning of consensus round that have not already been applied. It is declared to be public in the form of a list known as "candidate set".

The server has the responsibility to combine the candidate set of all servers on its UNL. It then votes for the transaction with "yes" or "no" votes after verifying its transactions. Receiving a minimum percent of yes votes is considered

to be the criteria to move into the next round. The minimum percent required in first round is typically 50 percent.

Transactions that receive more than the desired percent of "yes" votes for that particular round are passed on to the next round. Others are either discarded or included in candidate set for beginning of consensus process on next ledger. The final round of consensus requires 90 percent of all servers on UNL to agree on a transaction.

All transactions that meet this requirement are included in the ledger. It is then closed and thus becomes the new last closed ledger. This process continues and hence blocks get added to the distributed ledger after multiple validation rounds. Network consensus is achieved in Ripple using XRP LCP (XRP ledger consensus protocol) and goes through deliberation, validation and preferred branch phases. Multiple Ripple ledgers can communicate using Interledger protocol.

3.4.4 Tendermint protocol

Tendermint [28] tries to achieve consensus by taking account of stake of validators. It avoids the "nothing at stake" problem wherein validators have nothing to lose even if they misbehave over the network by using proper penalizing techniques. It relays new information by gossip.

The algorithm was initially based on DLS protocol [29] though there have been attempts to modify it. Every participating node keeps a complete copy of sequence of transactions in blocks included in blockchain. Each user keeps an account in the system and it is identified by users' public key or address. Each account can hold sum of coins. These may change with new transactions.

Nodes relay new transactions which were signed and submitted by users to a node of the network. Special users with accounts that have coins locked in a bond deposit by posting a bond transaction are the validators of the system. The voting power of a validator is equal to the amount of bonded coin his account holds. The voting power of a validator reduces only when its coins are unlocked later by unbonding transaction.

A set of validators with at least two-third of total voting power have the power to confirm a block. A block is said to be committed when a two-third majority of validators send commit votes for it. It is called "polka".

A fork is identified in the blockchain, when two blocks at the same height are each signed by two-third majority of validators. So a fork can happen only when one-third majority of validators signs in duplicate.

A short evidence transaction can be generated by anyone who gets two conflicting commit vote signatures. The guilty validator gets punished when this

is committed into the blockchain and it destroys their bonded coins. Validators participate in consensus process by signing votes for blocks. There are three types of votes - Prevote, Precommit and Commit.

A block is said to be committed by the network when a two-third majority of validators commit it (signed and broadcast commits). The block creation at a particular height is determined using round robin protocol. Each round has three steps -Propose, Prevote and Precommit and two special steps - Commit and NewHeight.

A round is started by a dedicated proposer. They are chosen in a round robin fashion such that frequency of getting chance to propose is in proportion to their voting power. It broadcasts a proposal to its peers via gossip.

All nodes gossip the proposal to their neighbouring peers. In the beginning of Prevote, each validator makes a decision. No locking happens in this step. In case validators receive more than two-third majority of Prevotes for a particular acceptable block, the validator signs and broadcasts a Precommit for that vote. It also locks on to that block and releases any prior locks. A node has a lock on utmost one block at a time. If a node had not received more than two-third of Prevotes for a particular block, then it does not sign or lock anything. All nodes gossip all Precommits for the round to all their neighbouring peers. If two-third of Precommits is obtained for a block, then node enters commit state. Else it goes to propose step of next round.

For commit, two parallel conditions are to be satisfied. Node must receive the block committed by the network if it had not received already. Once a block is received, it signs and broadcasts a commit for that block. Secondly, node must wait until it receives at least two-third of commits for the block precommitted by the network. Then CommitTime is set to current time and transitions to NewHeight. In effect, blocks are added when two-third majority of validators agree.

Cosmos has been designed to facilitate inter blockchain communication. Cosmos hub lies at its core and interacts with participating blockchains using cosmos hub. Cosmos hub plans to use Atom as staking token.[54]

3.4.5 Comparison

Blocks are validated by every full node in Bitcoin after verifying the well formed transactions and UTXO's. At the heart of Bitcoin and Ethereum is the proof of work validation concept. Transaction, ommer, state and nonce validation is done in Ethereum. Ripple begins with fifty percent trust vote and reaches consensus when nodes receive ninety percent votes from members in its unique node list. Block verification is done using State in Tendermint and relies on two-third voting for consesnsus.

The chain with heaviest path is chosen as the valid chain in Bitcoin and Ethereum. Since Ripple and Ethereum uses validator set and follow strategies different from that of proof of work, longest chain rule is not followed here.

The significant difference between Ripple and Tendermint is on the basic method they used to achieve consensus.

1. Ripple uses federated consensus while Tendermint uses proof of validation and stake.

2.Ripple achieves Byzantine fault tolerance of twenty percent while Tendermint is developed as one-third Byzantine fault tolerant.

3. For a block to be confirmed in Ripple, it takes multiple rounds. The initial round uses minimum acceptance percent of fifty and grows to 80 percent for final acceptance. Tendermint uses three levels of voting in a single round and accepts or rejects a block. The type of vote cast in ripple is "Yes" or "No".

4. Tendermint uses three types of votes - Prevote, Precommit and Commit. Consensus is achieved in Tendermint by validators collecting votes from nodes. In ripple, consensus is based on votes received from members in UNL of each server. UNL is a list of public keys associated with validating nodes. Ripple achieves accountability by flagging malicious nodes for removal.

5. Tendermint uses locking mechanism and evidence transaction to achieve accountability. The network split detection algorithm prevents forks.

6. Commit vote in Tendermint has highest significance. It can invalidate Prevote and Precommit of previous rounds and hence prevent fork.

7. Ripple and Tendermint provides assurance of convergence. In Ripple, an upper bound is set and nodes which do not satisfy it are removed from UNL. There is a lower bound of two seconds in each consensus round wherein node can propose their initial candidate sets. A latency bound heuristic is enforced on all nodes in Ripple network. Tendermint proceeds with the rounds. If two-third majority commits are not obtained, the algorithm proceeds to the next round. The commits of the latest round are considered most significant and hence ensures convergence.

8. Power to achieve consensus is intrinsic to the blockchain system in Ripple and Tendermint and hence are permissioned systems.

9. In Tendermint, power lies with validators. In Ripple, the configuration of servers and their UNL's has a major influence on architecture of the system.

10. Ripple focuses on blockchain solutions for financial domain and is a part of inter ledger protocol as well. Tendermint has several sub protocols and provides application development platform through Cosmos.

3.5 Algorithms involved

Consensus in bitcoin and ethereum are based on proof-of-work. Ripple uses federated consensus. Tendermint uses proof of validation for consensus.

3.6 Application development

Blockchain applications must implement deterministic finite state machines to be securely replicated by the consensus algorithms. The four steps followed by Bitcoin full nodes for consensus ensures that same order is maintained for transaction validation and block validation.

In Ripple, applications are built using methods provided in RippleAPI.

In Tendermint, given the same ordered set of requests, all nodes will compute identical responses, for all BeginBlock, DeliverTx, EndBlock and Commit. The responses are included in the header of the next block, either via Merkle root or directly. So all nodes must agree on exactly what they are. If there is some non-determinism in the state machine, consensus will eventually fail as the nodes disagree over the correct values for block header. The nondeterminism must be fixed and nodes restarted.

4 Conclusion

In this paper, the features of Bitcoin, Ethereum, Ripple and Tendermint were compared in an inside out manner. Method to achieve high scalability and performance is required to successfully replace backbone of current IT systems with blockchains. Formal verification of algorithm guarantees of these blockchains need to be done more rigorously. This will ensure adequate security to the systems.

Acknowledgements

We thank Dr. M Sethumadhavan, Head of Department, TIFAC Core in Cyber Security, Amrita Vishwa Vidyapeetham, Ettimadai, Coimbatore and Dr. M Sindhu, Assistant Professor, TIFAC Core in Cyber Security, Amrita Vishwa Vidyapeetham, Ettimadai, Coimbatore for helping us with the idea of comparing Ripple and Tendermint in an earlier paper [30]. We also thank Catherine, MA Linguistics, for her valuable suggestions in correcting the linguistics of the paper.

References

[1] *Central vs. Distributed* https://www.securitymagazine.com/articles/76457-central-vs-distributed-1 Last accessed: 24 Oct 2019

[2] *Blockchains: Past, Present, and Future* Arvind Narayanan, 2018, 37th ACM SIGMOD-SIGACT-SIGAI Symposium, ACM

[3] *Core Banking Solutions, Comfort or Hurdle to Customer (With Special Reference to SBI)* Vishal Geete, 2011, Research Journal of Social Science and Management, Vol. 1, No. 5, p. 214, 2011, SSRN

[4] *MultiChain Private Blockchain — White Paper* Gideon Greenspan, Founder and CEO, Coin Sciences Ltd, 2015

[5] *In Search of an Understandable Consensus Algorithm (Extended Version)* Diego Ongaro and John Ousterhout, 2014, USENIX ATC'14 Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference, Philadelphia, USA

[6] *Bitcoin: A Peer-to-Peer Electronic Cash System* Satoshi Nakamoto, 2008, https://www.bitcoin.org, Last Accessed: 24 Oct 2019

[7] *Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money* Gareth Peters, Efstathios Panayi, 2015, chapter from book Banking Beyond Banks and Money: A Guide to Banking Services in the Twenty-First Century (pp.239-278)

[8] *Smart contract applications within blockchain technology: A systematic mapping study* Daniel Macrinici, Cristian Cartofeanu, Shang Gao, 2018, Telematics and informtics, Volume 35, Issue 8, December 2018, Pages 2337-2354

[9] *Napster: Home* `https://us.napster.com`, Last Accessed: 24 Oct 2019

[10] *The Bittorrent P2P File-Sharing System: Measurements and Analysis* J.A. Pouwelse, Pawel Garbacki, D. H. J. Epema, Henk J. Sips, 2005, Peer-to-Peer Systems IV: 4th International Workshop, IPTPS 2005, Ithaca, NY, USA, February 24-25, 2005. Revised Selected Papers (pp.205-216)

[11] *Google Protocol Buffers* https://developers.google.com/protocol-buffers, Last Accessed: 24 Oct 2019

[12] *bitcoinj* `https://bitcoinj.github.io`, Last Accessed: 24 Oct 2019

[13] *Hardening Stratum, the Bitcoin Pool Mining Protocol* Ruben Recabarren, Bogdan Carbunar, 2017, Proceedings on privacy enhancing technologies

[14] *Ethreum yellow paper* `https://github.com/ethereum/yellowpaper`, Last Accessed: 24 Oct 2019

[15] *Kademlia: A Peer-to-peer Information System Based on the XOR Metric* Petar Maymounkov and David Mazi Eres, 2002, IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems, Pages 53-65

[16] *Analysis of the XRP Ledger Consensus Protocol* Brad Chase, Ethan MacBrough, 2018, https://arxiv.org/abs/1802.07242, Last Accessed: 24 Oct 2019

[17] *Peer-to-Peer Architecture Case Study: Gnutella Network* Matei Ripeanu, 2002, Proceedings First International Conference on Peer-to-Peer Computing, Linkoping, Sweden

[18] *Tendermint node* https://tendermint.com/docs/spec/p2p/node.html, Last Accessed: 24 Oct 2019

[19] *Ethreum Serenity* https://docs.ethhub.io/ethereum-roadmap/ethereum-2.0/eth-2.0-phases/, Last Accessed: 24 Oct 2019

[20] *Introduction to formal languages, automata theory and computation* Kamala Krithivasan, Book, pages 267-271

[21] *Guide to Elliptic Curve Cryptography* D. Hankerson, A.J. Menezes, and S.A. Vanstone, Book, Springer, 2004

[22] *Guide to Elliptic Curve Cryptography* D. Hankerson, A.J. Menezes, and S.A. Vanstone, Book, Springer, 2004, pages 184-185

[23] *Ripple Data API v2 - XRP Ledger* `https://xrpl.org/data-api.html`, Last Accessed: 24 Oct 2019

[24] *How to Build a Highly Available System Using Consensus* Butler Lampson, 1996, Proceedings First International Conference on springer-Verlag, https://www.microsoft.com/en-us/research/publication/how-to-build-a-highly-available-system-using-consensus, Last Accessed: 24 Oct 2019

[25] *The part-time parliament* Leslie Lamport, 1998, ACM Transactions on Computer Systems 16, 2, pages 133-169, New York, USA

[26] *A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks* Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, Dong In Kim, 2018, `https://arxiv.org/pdf/1805.02707` Last accessed 28 Oct 2019

[27] *Overview of Colored Coins* Meni Rosenfeld, 2012, White paper

[28] *Blockchain Consensus - Tendermint* `https://tendermint.com/`, Last Accessed: 24 Oct 2019

[29] *Consensus in the presence of partial synchrony* Cynthia Dwork, Nancy Lynch, Larry Stockmeyer, 2002, Journal of ACM

[30] *On federated and proof of validation based consensus algorithms in Blockchain* K N Ambili, M Sindhu, M Sethumadhavan, 2017, IOP Conference Series: Materials Science and Engineering 225 (1), 012198