On Federated and Proof Of Validation Based Consensus Algorithms In Blockchain

IOP Conf. Series: Materials Science and Engineering **225** (2017) 012198 doi:10.1088/1757-899X/225/1/012198

On Federated and Proof Of Validation Based Consensus Algorithms In Blockchain

Ambili K. N., Sindhu M., M. Sethumadhavan
TIFAC CORE in Cyber Security
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University, India
ambilikn@gmail.com, m_sindhu@cb.amrita.edu, m_sethu@cb.amrita.edu

Abstract: Almost all real world activities have been digitized and there are various client server architecture based systems in place to handle them. These are all based on trust on third parties. There is an active attempt to successfully implement blockchain based systems which ensures that the IT systems are immutable, double spending is avoided and cryptographic strength is provided to them. A successful implementation of blockchain as backbone of existing information technology systems is bound to eliminate various types of fraud and ensure quicker delivery of the item on trade. To adapt IT systems to blockchain architecture, an efficient consensus algorithm need to be designed. Blockchain based on proof of work first came up as the backbone of cryptocurrency. After this, several other methods with variety of interesting features have come up. In this paper, we conduct a survey on existing attempts to achieve consensus in block chain. A federated consensus method and a proof of validation method are being compared.

Keywords: Blockchain, Consensus algorithm, Crypto currency, Ripple, Tendermint.

1. INTRODUCTION

Most of the existing IT applications run on centralized architecture. They rely on one or more intermediaries to successfully conduct business on a global scale. Distributed systems involving variations of conventional consensus algorithms are being used by various service providers to satisfy specific needs. To reduce dependency on third parties and remove associated issues like double spending, distributed ledger is considered to be a good option. Databases have characteristics which enable features like core banking. But there are also problems like forgery of transaction, reversal of transaction and censorship of transaction [16]. These can be minimized with cryptographically strong distributed ledgers. Forgery then becomes impossible. Of the other two, priority can be set based on use cases. The strength of these distributed ledger increases manifold when strong cryptographic primitives are added as in blockchain. The distributed database in blockchain makes use of consensus algorithms to agree on a common value on a peer to peer network. With blockchains, consensus is on computation. The data structure involved in a blockchain is a chain of blocks which grows in the forward direction only. Each block is linked strongly using cryptographic techniques with the previous one and contains data of all transactions within a period of time. Data integrity is maintained using

Merkle tree.

2. BLOCKCHAIN TECHNOLOGIES

Blockchain is technically the back end database that maintains a distributed system openly [17]. Blockchain based on proof of work first came up as the backbone of cryptocurrency [1]. 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. However, blockchains are not suitable for transactions requiring promptness. For example, on a coffee vending machine, customer will not be willing to wait for long to get his cup of coffee once he pays while inclusion of transaction on blockchain may take time since it is a distributed system. 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 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 permission less [6]. 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, permission less 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 [5]. There is an active attempt to adopt blockchain based architecture in various domains [7, 12]. There are many use cases in financial domain like for remittance, settlement, transaction of securities, claims etc. Ripple [18] is one of the sample blockchains implemented in this field. Another use case is for cloud funding and investment to artists. Since there are no third parties involved, the artists and business entities are bound to obtain larger shares from collected funds. Contributors can receive dividends under a smart contract in an ongoing trial implementation called Swarm [21]. To ensure that artists get their share of payment for each copy of their work, a file format called dotblockchain has been proposed by dot blockchain music project [30, 34]. GyftBlock [20] tries to provide an exchange service of gift cards using blockchain and can also control users and monitor how the service is used. Cloud service can also be provided on blockchain. Storj [28] tries to redefine cloud storage using end to end encryption on blockchain. Messaging services and Social Networking Services (SNS) are tried out in Getgems [22]. It provides SNS separately from blockchain. Virtual currency, grant GEMZ tokens to users when they browse advertisements etc. are sample services provided on blockchain. Ownership and transfer of assets including land registration can be managed on blockchain. Factom[13] has commenced the provision of such a service. The scope of blockchain as a mechanism to manage the authentication of various items like works of art, digital contents etc are wide. Ascribe [32] provides a service to manage copyright of works of art on a blockchain. Rights to use shared cars or other goods in sharing economy can

be managed using blockchains. La'Zooz [23] intends to provide such a service. At present, it presents a ride sharing application like Uber. Traceability of commodities can be realized by registering all histories of processing from raw material to final products by replacing EDI with blockchains. Everledger [24] tries to provide such a system using blockchain by utilizing 42 different measurable parameters of diamond. Blockchains can be used for delivering content on internet. Streamium [25] provides a service to support content delivery. A new service has emerged to have participants vote on prediction of various matters and share rewards depending on voting results. This is called the prediction market. Augur [26] provides a decentralized prediction market platform. BitHealth [27] aims to achieve decentralized medical services by enabling users to safely check their own health records from anywhere in the world while keeping data available to limited parties. ADEPT is an attempt by IBM and Samsung to provide blockchain service in IoT.

Various organizations have planned to try out blockchain platforms. Myanmar has decided to establish a blockchain based stack exchange. Scotland is looking at their own blockchain driven stock exchange. MIT Media Labs has rolled out Blockcerts - an open infrastructure for academic credentials on the blockchain. Webjet has begun testing blockchain for hotel bookings. IBM is working with SBI securities to test blockchain technology for bond trading platform [30]. Attempts are on to utilize the cryptographic strength behind blockchain and scalable applications. BigchainDB [35] is an attempt in this direction.

3. CONSENSUS ALGORITHMS

Consensus algorithms [3, 17] 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. These may be users who are given rights to collectively perform transitions through an algorithm. It should be securely de-centralized. The result is that no single actor or group of actors can take up majority of the set. Paxos algorithm 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, PBFT and derivatives and N2N [15]. Each of these basically tries to solve Byzantine generals' problem [9]. This 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 [11].

There are different ways though which consensus is trying to be achieved. Under proof of work [10,14], 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 [29] and transferred over the networks running on proof of work based blockchains. A successfully running example is the bitcoin [33]. 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 [16]. 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 [14] 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 permission less or discretionary systems may not help always. If domain of finance is considered, the validators cannot be pseudonymous or anonymous and 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 [16]. 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 validators 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.

3.1 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 stats 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 80 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. Multiple Ripple ledgers can communicate using Interledger protocol.

3.2 *Tendermint Protocol*

Tendermint [8, 19] 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 [2] 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.

4. Comparison Of Ripple And Tendermint

The significant difference between Ripple and Tendermint is on the basic method they used to achieve consensus. Ripple uses federated consensus while Tendermint uses proof of validation and stake. Ripple achieves Byzantine fault tolerance of twenty percent while Tendermint is developed as one-third Byzantine fault tolerant. 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”. 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. Tendermint uses locking mechanism and evidence transaction to achieve accountability. The network split detection algorithm prevents forks. Commit vote in Tendermint has highest significance. It can invalidate Prevote and Precommit of previous rounds and hence prevent fork. 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. Power to achieve consensus is intrinsic to the blockchain system in Ripple and Tendermint and hence are permissioned systems. 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. Ripple focuses on blockchain solutions for financial domain and is a part of inter ledger protocol as well. Tendermint has several sub protocols and aims to provide application development platform through Cosmos. Comparison of the approaches to achieve consensus in Ripple and Tendermint is given in Table I.

Table I.	
Ripple	**Tendermint**
Federated consensus	Proof of validation
BFT - 20%	BFT – 33%
Multiple rounds for a block to be confirmed	Three votes per round for a block to be confirmed
Yes or No vote	Prevote, precommit and commit vote
Accountability - flagging for removal	Accountability – locking mechanism and evidence transaction
Servers and UNL	Validators
Member of Interledger	App development platform through Cosmos
Network split detection algorithm to prevent fork	High priority to commit vote of last round
Malicious nodes identified flagged and removed from network	Malicious nodes penalized and suffer lowering of their account value
Applied in financial domain	All kinds of applications can be developed in any language. It is also suitable for light weight applications like IoT

5. CONCLUSION AND FUTURE WORK

The proof of work method of consensus allows pseudonymous and anonymous participant nodes. The mining process in proof of work consumes lot of electricity. Consensus algorithm comparable against the proof of work mechanism suitable for use in private networks with known validators is yet to be designed. In this paper, the features of Ripple and Tendermint were compared. Method to achieve high scalability and performance is required to successfully replace backbone of current IT systems with blockchains. Performance optimization for Tendermint needs to be done when running with systems on a large scale. Formal verification of algorithm guarantees of ripple and Tendermint is yet to be done. Lot more of applications based on Tendermint sub protocols need to be built. Also, capacity of system needs to be increased in both these consensus algorithms.

REFERENCES

[1] Andreas Antonopoulos, Mastering bitcoin, 2014
[2] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer, "Consensus in the presence of partial synchrony", Journal of the ACM (JACM) 35.2 ,1988, pp. 288-323.

[3] Diego Ongaro and John Ousterhout, "In Search of an Understandable Consensus Algorithm (Extended Version)", USENIX Annual Technical Conference (USENIX ATC 14), 2014.
[4] David Schwartz, Noah Youngs and Arthur Britto , ”The Ripple protocol consensus algorithm” ,white paper, Ripple Labs,2014.
[5] Erik Hillborn and Tobias Tillstrom, "Applications of smart-contracts and smart-property utilizing blockchains", Feb 2016.
[6] Gareth.W.Peters and Efstathios Panayi, "Understanding Modern Banking Ledgers through Blockchain Technologies", 18 Nov 2015.
[7] Jacob Stenum Czepluch, Nikolaj Zangenberg Lollike and Simon Oliver Malone, “Use of block chain technology in different application domains”, 20 May 2015.
[8] Jae Kwon, “Tendermint : Consensus without mining”, white paper, 2014.
[9] Lamport, Shostak and Marshall Peace, “The Byzantine Generals Problem”, 1982
[10] Marko Vukolic, “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication”, Proc. IFIP WG 11.4 Workshop on Open Research Problems in Network Security (iNetSec 2015), 2015.
[11] Nomura Research Institute, “Survey of block chain technologies and services”, 2015.
[12] Paul Snow, Brian Deery, Jack Lu, David Johnson, Peter Kirby –Factom whitepaper- 17 Nov 2014.
[13] Satoshi Nakamoto, “Bitcoin: A peer to peer electronic cash system”, 2008.
[14] Sigrid Seibold, George Samman, “Consensus - immutable agreement for the internet of value”, KPMG, 2016.
[15] Tim Swanson, “Consensus as a service: a brief report on the emergence of permissioned distributed ledger systems”, April 2016.
[16] William Mougyar, “The Business Blockchain: Promise, Practice, and Application of the Next
Internet Technology”, 2016.
[17] https://ripple.com/
[18] https://tendermint.com
[19] https://block.gyft.com/
[20] http://swarm-gateways.net/bzz:/swarm/
[21] https://gem.co/
[22] www.the-blockchain.com/docs/LaZooz
[23] www.everledger.io/
[24] https://streamium.io/s/blockchain-uni
[25] https://www.augur.net/
[26] https://angel.co/bithealth
[27] https://storj.io/
[28] https://coloredcoins.org/
[29] https://www.the-blockchain.com/
[30] https://blockchain.info/charts
[31] https://www.ascribe.io
[32] https://bitcoin.org
[33] http://dotblockchainmusic.com/
[34] https://www.bigchaindb.com/

Ambili K. N., received her B. Tech in Computer Science from Calicut University and currently pursuing her M. Tech. in Cyber Security at Amrita University. Her current research interests include: Blockchain technology and Cryptography.

Sindhu M., received her PhD (Cryptography) from Amrita Vishwa Vidyapeetham (University). Currently, she is working as an Assistant Professor in TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham University, Coimbatore. Her current research interests include: Crytography and sequence analysis .

M. Sethumadhavan received his PhD (Number Theory) from Calicut Regional Engineering College. Currently, he is working as a Professor in the Department of
Mathematics and Computer Science, Amrita Vishwa Vidyapeetham University, Coimbatore. His current research interests include: Cryptography and Boolean functions .