Using raft as consensus algorithm for blockchain application of roaming services for mobile network

Ahmad al-Qerem a,1, *

aFaculty of Computing, Princess Sumaya University for Technology Amman 11941 Jordan
1 a.qerem@psut.edu.jo *
* corresponding author

1. Introduction

The blockchain is essentially a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified. Moreover, when information entered it can never be erased. The current digital economy uses a third trusted party to make safe transactions, but these parties can be hacked or manipulated, that is why the technology of blockchain appears.

The main vital characteristics of blockchain are distributed consensus and anonymity. There are two types of blockchain; public blockchain and private blockchain, in the public blockchain anyone can join the network and be a part of it, but in the private blockchain the members are known. Private blockchain supports private transactions and private contracts. According to this paper, private blockchain can be implemented using Raft consensus algorithm.

Because of the consensus algorithm role in maintaining the blockchain safety and efficiency, using the right consensus algorithm may affect the blockchain performance and its application. One of the most essential things in distributed systems is the consistency of their node and here comes the role of consensus algorithm to make these nodes work as a coherent group that is considered to be as a reliable large-scale system.

**ARTICLE INFO**

**ABSTRACT**

In the past few years blockchain became the new trending technology, it is expected to spread as internet in the 80s. Blockchain technology, using a consensus algorithm, is a cryptographic record that cannot be changed by repetitive sequence hashing and failure tolerance. Decentralization, persistency, anonymity, and audibility is the key characteristics of the blockchain. These characteristics encourage many industries to be interested in blockchain adoption in their IT systems. The consensus is one of the critical problems in the blockchain; these consensus algorithms play a critical role in maintaining the safety and efficiency of the blockchain. Unless transactions are pending, the Raft Consensus will not mint blocks. The raft can lead to significant storage savings, particularly if the transaction load is low because zero transactions are not minted in empty blocks. The performance of a blockchain that implements raft as a consensus algorithm has been evaluated through the simulation of roaming services for mobile network operators (MNOs) application. Performance evaluation to determine how quickly the network accepts blockchain transactions are reported. The result indicated that the proposed mechanism increase the throughput of transaction significantly.

This is an open access article under the CC-BY-SA license.

Keywords
Blockchain
Private blockchain
Raft consensus algorithm
Roaming services
Mobile network operators (MNOs) application
The raft is one of consensus algorithm that manages a replicated log [1]. The primary objective of its design is to be easy to understand. It is equivalent to Paxos in fault-tolerance and performance. The main difference is that Raft decomposes the problem into subproblems and solve them independently. In order to execute Raft in the blockchain, the node inside the network should be known and adjustable so they can exchange the messages easier, this network is known as private blockchain as Mentioned before.

Blockchain technology, which was first off coined in Bitcoin [1] by Satoshi Nakamoto in 2008, has received intensive attention recently. A blockchain is distributed database/transaction system wherever all the peers share info in an exceedingly decentralized and secure manner.

Thanks to its advantageous characteristics as decentralization and auditability, blockchain becomes a promising technology for several types of assets transfer and point-to-point (P2P) dealing. Recently, blockchain-based applications are turning out, covering various fields as well as commercial services [2]–[4], internet of Things (IoT) [5], supply chain management systems [6], and so on. Since the essence of blockchain could be a distributed system, Consensus algorithms play a vital role in maintaining the security and efficiency of blockchains.

A Consensus algorithm for distributed systems is to work out the coordination among multiple nodes, i.e., the way to return to the agreement if there are multiple nodes. Many Consensus algorithms are planned, e.g., Proof-of-Work (PoW) [1], Proof-of-Stake (PoS) [7] [8], Delegate Proof-of-Stake (DPoS) [19], sensible Byzantine Fault-Tolerant (PBFT) [10], Paxos [11] and Raft [12]. Among them, a prisoner and PoS algorithms have good support for safety, fault tolerance and quantifiability of a blockchain. Thus, prisoner and PoS are simple decisions of public blockchains, during which anyone will be part of the network, and there are not any trust relationships among the nodes. However, prisoner and PoS have an unsatisfied performance of dealing confirmation that limits its applications to those requiring high confirmation speed.

In an exceeding consortium/private blockchain network, all participants are whitelisted and finite by strictly written agreement obligations to behave “correctly,” and thus a lot of economic Consensus algorithms like PBFT and Raft are more appropriate choices. Consortium/private blockchains may be applied to several business applications.

As an example, Hyperledger [13] is developing business syndicate blockchain frameworks. Ethereum has additionally provided tools for building syndicate blockchains [14]. Raft algorithm is taken into account as a Consensus algorithm for personal blockchains [15] that is applied to a lot of unintended networks like the computer network.

What is more, many hybrid Consensus protocols are planned to enhance consensus potency while not undermining quantifiability. As an example, Zilliqa [16] planned prisoner to elect directory service (DS) committee nodes, and so the DS committee should run PBFT Consensus protocol on the sealing block. Compared with PBFT and Paxos, the Raft algorithm has high potency and ease, and it has been widely adopted within the distributed systems. Raft could be a leader-based algorithm that uses leader election as a vital half for the Consensus protocol — ledger entries in Raft-based system flow in little one direction from the leader to alternative servers.

Paxos and its improved algorithms do not take leader election as a vital a part of the Consensus protocol, which may balance load well among nodes as a result of any node may commit commands [11] [17]. However, Paxos’ design needs advanced changes to support sensible systems.

Raft achieves the identical safety performance as Paxos and is a lot of convenient in engineering implementation and understanding. Raft Consensus algorithm cannot tolerate malicious nodes and might tolerate up to fiftieth nodes of crash fault. For private blockchains, nodes are verified, members. Hence, it is a lot of vital to unraveling the crash faults than Byzantine faults for personal blockchains. The network is termed split once over 1/2 nodes are out of current leader’s management. Failure of node and communication interruption caused by packet loss are the most reasons for a network split. If a network split happens, the blockchain network with Raft Consensus algorithm will re-start a replacement leader election method. Meanwhile, the blockchain network stops acceptable new dealing, i.e., the blockchain network becomes untouchable.

The reminder of this paper will show related work -quorum- (section II), have an overview on how blockchain work (section III), overview on Raft consensus algorithm (section IV), it will also explain
how raft can be used in blockchain (section V), system fraud prevention case study explain in (section VI), the simulator and evaluation in section VII and finally the conclusion.

2. Background

Companies and researchers are recently working on private and permissioned blockchain to achieve their enterprise goal. Therefore there are many private blockchains like corda and quorum, but the one who uses a voting base consensus algorithm is quorum. JP Morgan developed quorum [6] as a private/permissioned blockchain, it uses a voting based consensus algorithm and one of its design goals is to reuse as much existing technology as possible that is why it is based on the official Go implementation of the Ethereum protocol.

Fig. 1. quorum architecture.

The modification on the go-ethereum codebase includes the blocking proposal and block validation process the figure above shows the quorum architecture. Quorum supports both public and private transaction. This will result in a segmentation of the state database, i.e. the state database is split into a private state database and a public state database [7].

- The Quorum has the following modifications to get (the Ethereum Go client):
  - It is a permissioned blockchain, unlike Ethereum.
  - The consensus is achieved with the Raft or Istanbul BFT consensus algorithms instead of using Proof-of-Work.
  - The P2P layer has been modified to only allow connections to/from permissioned nodes.
  - The state database is split into a private state database and a public state database.
  - Block validation logic has been modified to handle ‘Private Transactions’.
  - Transaction creation has been modified to allow for Transaction data to be replaced by encrypted hashes in order to preserve private data where required.
  - Quorum eliminated the concept of adding cost to a transaction using gas [8].

3. Blockchain

This section will explain blockchain by explaining how BitCoin works since it is the first application that uses the blockchain. In the Bitcoin system, each transaction is protected using a digital signature. Each transaction is broadcast to every node in the Bitcoin network and then recorded in a public ledger after verification. The blockchain is used in the Bitcoin network to agree on the order of transactions because each block in the blockchain contains the hash of the previous block as shown in the Figure 2. If two blocks from different nodes where created at the same time, the blockchain can decide which block to add, after answering a very difficult mathematical puzzle, this is known as “proof of work”, proof of work is one of the consensus algorithm, the first node who solve the problem broadcasts the new blockchain to all nodes in the network. Those nodes who solve these mathematical problems called “miner nodes.”
Ahmad al-Qerem (Using Raft as Consensus Algorithm for Blockchain Application ...)

These approaches make the blockchain safe against the fraudulent transaction because if the attacker changed one block in the blockchain, he would end up changing the rest of the blocks.

**Illustrative Example 1:** The figure 3 below shows how blockchain works, if ‘A’ wants to issue a transaction with ‘B’, the transaction is represented as a block, the block is broadcast to all nodes in the network after it has been signed by ‘A’, after the transaction has been approved by the nodes in the network, the block now can be added to the public ledger after the miner node solve the mathematical puzzle, finally the money will move from ‘A’ to ‘B’[3].

**4. Raft consensus algorithm**

This section shortly summarizes the Raft algorithm. A lot of complete and elaborated description of Raft refers to [12]. Raft could be a Consensus algorithm for managing a replicated ledger at each node. At any given time, every node is in one amongst the three states: leader, follower, or candidate. Raft algorithm divides time into terms with finite length. Terms are numbered with consecutive integers. Every term begins with Associate in the election, during which one or many candidates try to become a pacesetter. If a candidate wins the election, then it is a pacesetter for the remainder of the term. The state transition is shown in Fig. 1. All nodes begin from the follower state. If an acquaintance doesn’t hear from the leader for a particular amount of your time, then it becomes a candidate. The candidate then requests votes from alternative nodes to become a pacesetter. Alternative nodes can reply to the vote request. If the candidate gets votes from a majority of the nodes, it will become a pacesetter. This method is termed Leader Election. Explicitly, if an acquaintance receives a heartbeat at intervals the minimum election timeout of hearing from a current leader, it does not grant its vote to the candidate. This helps to increase the length of a pacesetter to stay operating and avoiding...
frequent disruptions from some isolated/removed nodes. In traditional operation of Raft, there is specifically one leader and everyone the different nodes are followers. The leader sporadically sends out heartbeats to any or all followers to take care of its authority.

All transactions throughout this term undergo the leader every dealing is supplemental as Associate in an entry within the node’s ledger. Precisely, the leader initial replicates the received new dealing with the followers. At now, the entry continues to be uncommitted and stays in an exceedingly volatile state. Once the leader receives feedbacks from a majority of followers that have written the entry, the leader notifies the followers that this entry is committed. This method is termed Ledger Replication. In the Raft algorithm, there are many timeout settings. One amongst them controls the election method. The election timeout is that the quantity of time that an acquaintance has to wait to become a candidate. The election time counter is diminished as long because the follower receives no heartbeat. The follower transfers to candidate state once the election time reaches zero. The election time counter resets to a random worth once the follower receives a heartbeat from a pacesetter. The random IThese nodes will not receive heartbeats, in order that they can make out and begin new elections — figure 1. A state transition model for the Raft algorithm election timers in Raft facilitates to cut back the chance that many followers transfer to candidates at the same time.

In summary, the Raft algorithm, as mentioned before Raft is a consensus algorithm for maintaining a replicated ledger at every node. At any given time each node in the Raft will be one of these stats: a follower, a candidate, a leader. In each term there is one leader, and the other nodes are followers, the followers receive requests from the leader and candidate, the leader handle all the client requests and if the client interacts with a follower the follower will direct him to the leader, if the follower does not receive a request from the leader he will be in the second state, candidate, where he can be a new leader if receiving a positive vote form the majority, figure 4 summarize the three stats.

Fig. 4. servers’ status in the Raft.

Mainly Raft divided into stages called terms each term has arbitrary length, they are also numbered, and they all began with an election in which candidates are electing a new leader, if one of the candidates wins the election it will become a leader.

4.1 Leader election

As shown in figure 4 all nodes start as a follower each one of them has a randomized timeout, the leader in raft sent out a heartbeat by RCP to make sure that the leader is still operating normally and prevent followers to time-out. If the followers have not received a heartbeat from the leader over their time-out period, the flowers who time-out first will become a candidate and initiates a new election. A candidate first vote for itself then it requests votes from other nodes through an RPC called Request Vote, the one that receives votes from the majority of the nodes in the network becomes the new leader for the new term.

In some cases we can have a split vote situation even if the timeout is randomized, the solution for such situation is to increment the current term and then each node will have a new time-out for the next election, the figure below shows how the term can be when we have a split vote.

Fig. 5. Terms in the Raft.
4.2 Log replication

After electing a leader, this leader will handle the client request as we mentioned before. If receiving a request from a client the leader append the request to its log and does not commit it until he makes sure that it is stored in the majority of nodes in the cluster, so an RPC called, AppendEntries, sent out to the nodes and if receiving confirmation from the majority of nodes, leader can commit the request then on the next heartbeat the nodes know that it is safety to commit it.

![Log replication diagram](image)

Figure 6. nodes log in Raft network.

Log replication

4.2 Log replication

5. Raft on blockchain

Block chain consensus algorithm can be divided into two categories; the first one is proof based consensus (proof Of Work -Pow-, Proof Of Stack -Pos-) which we will not discuss in this paper; the second one is voting based consensus, in order to use this as a consensus algorithm in blockchain the nodes should be known and adjustable unlike the proof based algorithms, there are two algorithm that can be used as a voting base consensus one of them is Raft, the main reason that they don't use paxos because it was difficult to understand. In Bitcoin network the miners are the one who add a new block of transactions to the blockchain after solving a mathematical puzzle and then broadcast it to all nodes in the network, but using the raft as consensus algorithm the leader is the one who is responsible for this process, the leader will receive a signed block from the nods in the network and then append them in his blockchain then he will send an AppendEntries RPCs with the highest index that is committed in its blockchain; to make sure that every node have the same order of blocks.

Illustrative Example 2: For example if there are 3 node ‘A’, ‘B’, ‘C’ and ‘C’ was the leader node,’A’ wants to make transaction with ‘B’, like bitcoin network the transaction will be represented as a block signed by A, this transaction will be sent to ‘C’ and ‘C’ will approve it, if it was valid, then he will add it to the blockchain, and broadcast it by AppendEntries RPC, and finally the transaction will take place with ‘B’, see figure 7.

![Illustrative Example 2 diagram](image)

In order to make sure that every node in the network is consistent, raft guarantees these additional conditions:

On each tier, there is only one leader. Entries on leader logs cannot be overwritten or manipulated after the transactions have been acknowledged by the leader. If two logs contain the same entry term and index, then these logs are identical till this index. If a node commits command in the given index, then all other nodes will commit the same command. For other problems like solving the double
spending problem, each block will have an additional section called the hash section, in this section the content of the block and a nonce will be hashed together, so the leader will not accept two blocks with the same hash, the figure below shows the new block architecture.

Fig. 8. the new block architecture.

In the following: algorithm 1 and algorithm 2 show how both follower and leader will work when blockchain uses the raft as consensus algorithm, algorithm 3 shows how leader election will take place in the network.

1. While(This.State==Leader)
2. {   RespondToTrans()
4.   {   if(search(trans.hash)) then   RejectTrans();
      Else{    AddTrans();
    3.    SendAppenEntriesRPC();//for all
4.      if(isMajority()) then   N=CommitTheTrans()
5.      Else   delete()   }
6.      RespondToHeartBeat()
7.     {   if(Last Index>follower.nextIndex)
8.      {   DecrementNextIndex()
9.        retry sending();   }   }
10.    if (N>CommitIndex& MajorityCommit[i]>=N& then    log.term==currentTerm)
11.   .   SetCommitIndex(N);   }
12.   StepDownToFollower();

Algorithm 1: the role of leader when blockchain uses raft as consensus algorithm

As mentioned before the leader is the one who is responsible for responding to transactions request and adding these transactions to the blockchain, the lines from 2 to 5 shows how leader respond to transaction request. first he uses a binary search on the existing blockchain to make sure that there is no double spending problem if the same hash found, the transaction will be rejected otherwise the transaction will be appended to leaders blockchain, and it will not be committed until the transaction is safely appended in the majority of nodes’ blockchain in the network as shown in line 4 to 5.

The rest of the lines shows that if the leader receives a heartbeat, that means that the leader fail down for a period of time and a new election has happened, so the leader will check its term index with the new term, if it is bigger than his, he will step down to follower state, and tell the new leader the number of committed index he has to keep up with the transaction that happened during the fall down time.
1. RespondToRPC
2. {   if(isVoteRequest())
3.     If(this.term< term & this.Lastindex< Lastindex) Then VotPsitive();
4.     Else VotNegative(); } //if1
5. Else if(isAppenEntries())
6.     if(isLeaderPartOfTrans()) Then ValidateTrans();
7.     Else AppendEntry(); } //elseif
8. Else if(isAcknowledgement())
9.     if(receiveAcknowledgement()) Then CommitEntere();
10. Else ResendTransAction(); } //elseif
11. IssueTransAction(sender I, ReciverId,
12.     Segnetur, Data, Hash, TermNum); }

Algorithm 2: the role of Follower when blockchain uses raft as consensus algorithm

The follower in the blockchain that uses the Raft will only respond to RPCs and issues a transaction with other nodes in the network whenever wanted. Three types of RPCs can be received by followers the lines 2 to 4 shows how the followers will respond if it was a request vote RPC, first of all, the request would be checked if it was valid or not by checking the term number and last index. While the lines from 5 to 7 present the way of handing the Append Entries request, first the transaction will be checked by the follower only if the leader was part of it otherwise the block will be added to the blockchain. The rest of the pesedou code shows when to commit the transaction.

1. If (follower not receiving heart beat)
2. {   If(electionTimeOut())
3.     {   incrementCurrentTerm();
4.     ChangeToCadidate();
5.     VoteForMyself();
6.     RequestVoteRPC(term, candidateID,
7.     LastLogIndex, LastLogTerm);
8.     While(IsCandidate())
9.     {   WaitForFeedback();
10.     Int x=ProcessFeedBack();
11.     Foreach (EntryIn_AppendEntry())
12.     {   If(this.Term< entry.term) { //if 
13.         ChangeStateToFollower(); } //if  } //foreach
14.     If(x>(n/2 +1)) 
15.         ChangeStateToLeader();
16.         Send_RPCAppendEntry(); //heartbeat
17.     }
18.     Else if(x((n/2 +1))
19.         ChangeStateToFollower(); }
20.     Else //no one won
21.     i. incrementTerm();
22.     StartNewElection(); } //while } //if2 } //end if1

Algorithm3: the election procedure when blockchain uses raft as consensus algorithm

In this part the leader election will take place only if the leader fails down, thus the follower will not be receiving a heartbeat from the leader, so the one who times out first will change its state to a candidate, increment it term, vote for itself and request votes from other nodes in the network. As long as the state was candidate it will wait for feedback, then it will check if he wins the election or not, the lines from 11 to 11 represents the way of processing the incoming request vote during the waiting time of waiting for process the feedback, if the candidate win the election his state will be changed to
a leader state and start sending heartbeats, but if no one wins the election (if the positive vote is N/2, where N is the number of node in the network) the term will be incremented and issue a new election.

There are essential characteristics to determine how effective blockchain is, like scalability, security, Block creating speed, Award and if the Nodes can join freely. These characteristics can help the enterprise to achieve their goal of using blockchain.

6. Fraud Prevention Use Case

Telecommunication business is already within the middle of doing Digital Transformation because it moves towards virtualization, AI, RPA, etc. Therefore, adding Blockchain into the equation undoubtedly offers some measurable and quantitative advantages to CSPs in their day to day current running operations further as for future readiness. In Current situation, once a User is in Roaming to a partner Network, Partner Network sends the CDRs data as associate degree offline mechanism through some information financial organization (DCH). A DCH is later accountable to send the faucet files to Home Network. A permission blockchain might be enforced between each combination of operators that have a roaming agreement. Selected nodes from each operators act as miners to verify the holiness of every dealing broadcasted on the network through a Roaming Subscriber from Home to visitant network whenever roaming agreement is enforced between the house and visitant Network as a wise contract that's triggered once a transaction containing the CDR information is broadcasted on the blockchain network through a Subscriber sort of a Voice or information packet. On every occasion a subscriber triggers an incident in a highly visiting network, the VPMN broadcasts the CDR data as an agreement to the HPMN. This information triggers the smart contract and also the terms of the agreement are dead. The HPMN will therefore mechanically calculate the asking quantity supported the services rendered and send this data back to the VPMN. This helps fast and verified authorization further as a settlement to occur in line with blockchain-based smart contract terms. CSPs may also do away with the DCH acting as the middleman, leading to any price savings. Blockchain can change fancy datasets across multiple parties, in the period with high trust and security, significantly establishing subscriber identity.

7. Simulator-based model

In this section is introduced the system model for permissioned blockchain. We tend to tend to assume part synchronous communication network. A peer-to-peer distributed system of existent nodes communicate over the network and keep a regular state update. This state is mostly a replicated arrangement shared in memory between replicas, i.e. blockchain. The blockchain may well be an assortment of transactions unionized in blocks, it will be documented as a ledger of blocks where all is cryptographically joined to the previous by implies that of its hash. A dealing represents an associate quality exchange between two parties, secured by asymmetrical cryptography, whereas blocks are primarily collections of transactions with a determined size. Throughout this case, we tend to tend to determine two class of nodes, miners, and purchasers. The purchasers are nodes responsible for submitting transactions, whereas miners are answerable for correct acceptable transactions and collect them in blocks.

Ownership has been declared as external powers by an associate that can be regarded as a formal tool of discretionary external assessment of each node at the regional level. Transactions are offered by buyers to the network. Then, once these parameters are grouped into blocks, they change the relative chain state. Each block waits for compatibility or wait status or will be rejected. Once the miners reach consensus, they visit one of the blocks. Once validated, each person maintains a list of incomplete transactions and builds the relative blocks that will be plotted as follows from the blockchain. While the block, then the combined transactions, does not immediately commit to the blockchain, initially terribly intermittent} very waiting situation. Hence, to ensure the presence of offspring on the transactional order, the consensus protocol is operated among the miners to establish the following mass that must be adhered to within the common state. We tend to define a special category of nodes, i.e leaders who are delegated to propose a ban on a specific goal in a timely manner. Figure 9 illustrates an example of a built-in blockchain network. Once the miners have completed their work, the result of the leader of the election will propose a replacement block. This can be accepted and abide by blockchain by all nodes of the network. The blockchain system will be divided into two layers. The network's miners carry out, horribly, an initial base, an associate base to collect
transactions provided by buyers and to manage blocks. This level is supported by a second layer that runs the consensus protocol. These units are linked to each other, transaction execution is granted, and the joint ledger is updated to perform the gauge, we tend to reflect on the product and the extra time of transaction and quality. Furthermore, since the results of blockchains are asynchronous [18] services, the transactions submitted to the system do not appear to be processed immediately. To measure the latency of trades, we tend to talk to the time between the submissions in addition to the commitment of the block with the contract.

![Fig. 9. throughput versus a number of transaction in the block](image1)

![Fig. 10. throughput decreases with larger number of nodes](image2)

8. Conclusion

In this research we highlight the private Blockchain with vote-based consensus algorithm instead of the public Blockchain with proof-based consensus algorithms. This paper also explains the Raft as consensus algorithm for Blockchain by applying the raft consensus algorithm to blockchain by describing a detailed implementation through three algorithms. This analysis examined the variations between a public blockchain with a permissioned blockchain with regard to performance and measurability and the way viable Hyperledger material is to use among the finance business compared to ancient centralized solutions. The results have shown that sacrificing decentralization by making
authority and trust outside of the network greatly improves performance and measurability. A permissioned blockchain doesn't need machine power to support that trust compared to public blockchains that successively eliminates the throughput problems. Hyperledger material is actually abundant quicker and scalable than each Bitcoin and Ethereum, and it will guarantee information access to permit solely the participant in every party to a group action that can see sensitive details. For Telecommunication business use cases, Hyperledger material should meet the business necessities that are not possible to satisfy with a public blockchain. The results have shown that raft consensus implementation will place up with reasonable throughput.

References


