A Brief Analysis of Blockchain Algorithms and Its Challenges

Chapter · January 2021

DOI: 10.4018/978-1-7998-5351-0.ch002

2 authors, including:

Rajalakshmi Krishnamurthi
Jaypee Institute of Information Technology
77 PUBLICATIONS **236** CITATIONS

SEE PROFILE

Chapter 4
A Brief Analysis of Blockchain Algorithms and Its Challenges

Rajalakshmi Krishnamurthi
Jaypee Institute of Information Technology, India

Tuhina Shree
Jaypee Institute of Information Technology, India

ABSTRACT

Blockchain is the world's most trusted service. It serves as a ledger that allows transaction to take place in a decentralized manner. There are so many applications based on blockchain technology, including those covering numerous fields like financial services, non-financial services, internet of things (IoT), and so on. Blockchain combines a distributed database and decentralized ledger without the need of verification by central authority. This chapter surveys the different consensus algorithms, blockchain challenges, and their scope. There are still many challenges of this technology, such as scalability and security problems, waiting to be overcome. The consensus algorithms of blockchain are proof of work (POW), proof of stake (POS), ripple protocol consensus algorithm (RPCA), delegated proof of stake (dPOS), stellar consensus protocol (SCP), and proof of importance (POI). This chapter discusses the core concept of blockchain and some mining techniques, consensus problems, and consensus algorithms and comparison algorithms on the basis of performance.

INTRODUCTION

Blockchain is one of the most important services. It is a database which contains information about all the transaction ever executed in the past and works on bitcoin protocol. It combines a distributed database and decentralized ledger and there is no need of verification by a central authority. In blockchain, the completed blocks are recorded and added to the blockchain in chronological order so that market participants can keep track of digital currency transaction without central record keeping. Each time the block is completed, the new block is generated and completed blocks goes into the blockchain as a permanent database. Each block contains a hash of the previous block. The blockchain has all the information about user address and their balances from the genesis block to the most recent block. The

DOI: 10.4018/978-1-5225-9257-0.ch004

first block is called as genesis block in blockchain. The blockchain was designed so that the transactions cannot be deleted. The blocks are added using cryptography so that data can be distributed but not copied. The continuous growth of blockchain can be considered as a problem to some, such as creating issue of storage and synchronization. Blockchain works on bitcoin protocol. *So now what is bitcoin?*

Bitcoin is digital currency released as open source software (Singh, 2016) and was first invented by a researcher 'Nakamoto' in 2008. It is a digital token that can be stored in a digital wallet and is designed to work as a currency. It is often called as a cryptocurrency because encryption techniques are used to secure transactions and controls the creation of additional units. It is a decentralized cryptocurrency produced by all the participating nodes in the system at a defined rate. The chain of bitcoin created over period and linked to each other called block chain. Bitcoin, which gave birth to the concept of blockchain and Ethereum. Ethereum, is an open source, public, blockchain based distributed computing platform and operating system featuring smart contract functioning. Through blockchain, bitcoin is solved the double-spending problem which is the risk, particularly when digital currency is exchanged, that a person could concurrently send a single unit of currency to 2 different sources. So, the bitcoin become unique because it solved the double-spending problem through blockchain.

The main objective of this work is overview and compares different consensus algorithms. There are so many algorithms which are currently using for blockchain technology. So, for the comparison, we have taken some commonly used algorithms like Proof of Work (POW), Proof of Stack (POS), Proof of Importance (POI), delegated Proof of Stake (dPOS), Practical byzantine fault tolerance and Ripple Transaction Protocol. Then we will compare those algorithms based on properties of blockchain and how they are fit for the blockchain technology. This work focuses on steps of the algorithm, scalability and method of algorithm and security risks present within the algorithm. We will discuss about consensus problem which includes The Byzantine Generals Problem, Byzantine Fault Tolerance (BFT) and Delegated Byzantine Fault Tolerance (dBFT).

Key contributions of this chapter are:

- First understand the core concept of blockchain.
- Secondly analyze the architecture of blockchain and some mining techniques.
- Then discuss about consensus problems and consensus algorithms including steps of algorithm and scalability of algorithm.
- Then analyze and compare the algorithms on the basis of performance and security risk present in algorithm.
- Finally conclude with limitations of blockchain.

THEORETICAL BACKGROUND

In this section, our focus is on core concept of blockchain, architecture of blockchain and some mining techniques used in blockchain.

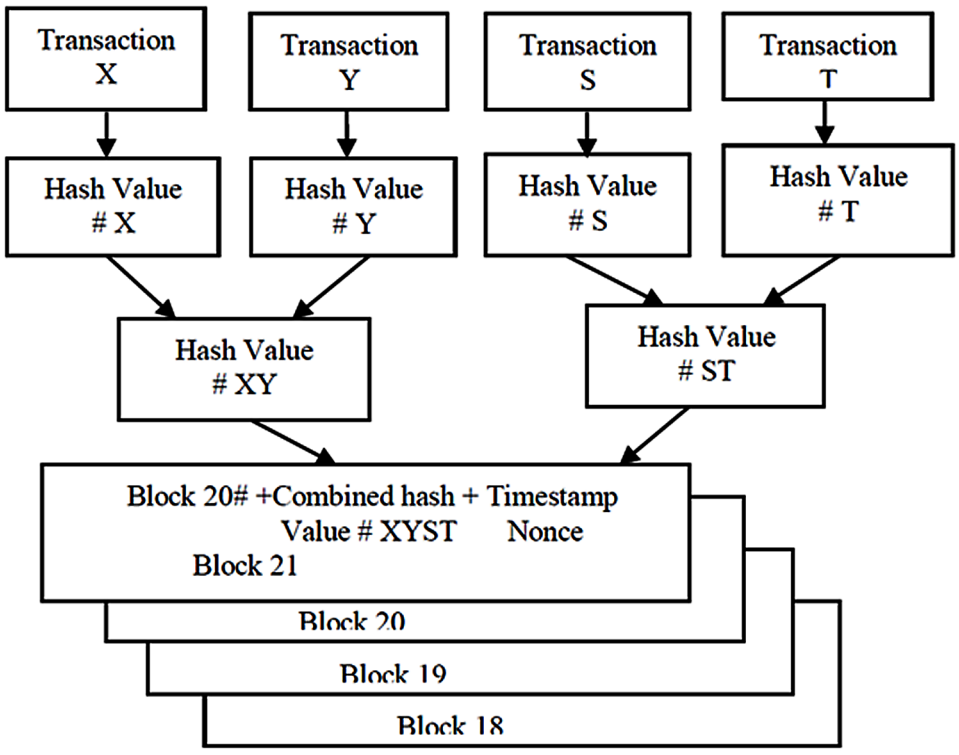
Core Concept of Blockchain

Satoshi Nakamoto gave birth to blockchain technology. He is the inventor of the cryptocurrency Bitcoin which is published in 2008. The core concept of this technology is that it is a public, shared and

tamperproof ledger that allows people to share information in a trustworthy ledger, where any sorts of immaterial information of value can be stored. The blockchain technology is efficient, can increase transparency. It reduces risks when less assets are tied up during transactions and reduce expenditure. The main idea of the blockchain technology is that it is accessible for everyone but cannot be controlled by any user alone. The participants together enhance and continue the blockchain by complying strict rules and general agreement that means the participants agree on how the chain will be updated. This agreement is called 'The consensus mechanism' (Laura, 2017).

In this technology, a peer-to-peer network is used. In peer to peer network, every node is equally privileged. Nodes can come and go as they placed in the network. Through mining, new blocks can be added by specialized nodes, that nodes are known as miners. In this currency transaction technology, multiple miners supervise that everything is in order and that the person who wants to make transaction actually has the money to spend and verify the transaction. If the transaction is valid, the miners confirm the change. Hereafter the only completed blocks are added to the chain and become blockchain. It is added in the chronological order with all the information. An identifying code has assigned to every transaction known as Hash value, which contains all the original information of the transaction. The hash value of the transaction is bundled together in a block, are combined in a system called as 'The Merkle Tree' (Laura, 2017) shows in Figure 1. The header of a new block contains this combined hash value additional with some more information such as the hash value of previous block and a timestamp. The timestamp proves that the data existed at the time being.

Figure 1. The Merkle Tree

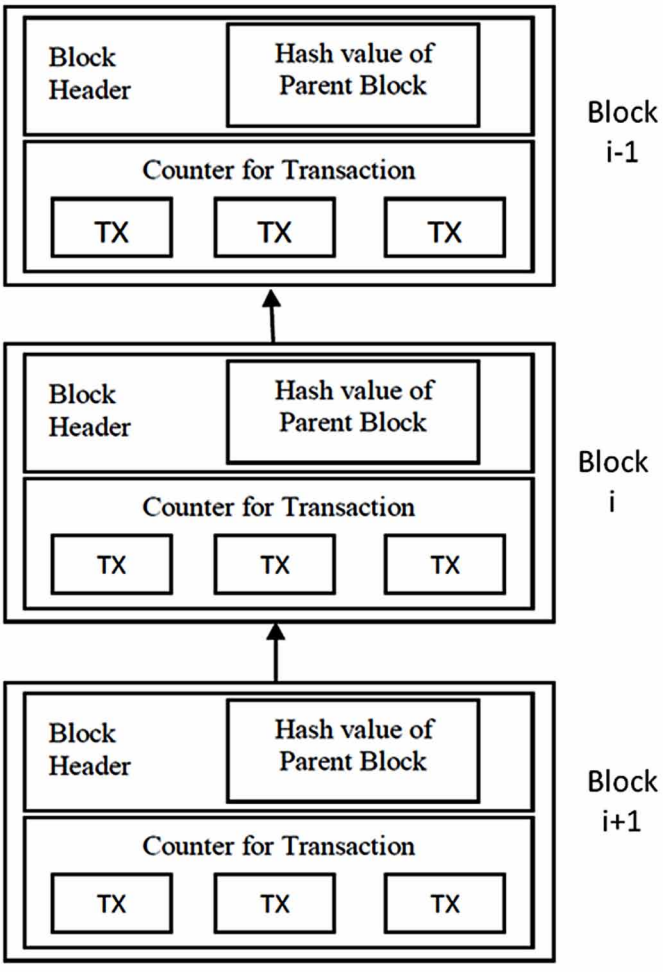


Architecture of Blockchain

A blockchain consist of sequence of blocks. A blockchain database is distributed, shared, fault tolerance and append only database that maintains the record in blocks (Salman et al., 2016). Blocks cannot be deleted or altered though it is accessible by all blockchain users. Blockchain database consist sequence of blocks. Each block has a hash value of its previous block and contains several verified transactions. Also, each block includes a timestamp and a random number (nonce) for cryptographic operations. Timestamp indicates creation of time of block. A block has only one parent block. The first block in the blockchain is known as genesis block which has no parent block and its hash value is entirely zeros. Figure 2 shows the architecture of blockchain (Zheng, 2017).

- **Block**: A block consists of the block header and block body (Zheng, 2017). The block header includes: -
  - **Blockversion:** It indicates which set of block validation rules to follow.
  - **Merkle Tree Root Hash:** The hash value of all the transactions in block.

Figure 2. Architecture of Blockchain



- **Timestamp:** Current time as seconds in universal time.
- **nBits:** It is a target threshold of a valid block hash.
- **Nonce:** It is a 4-byte field, which starts with 0 and increases for every hash calculation.
- **Parent Block Hash:** 256-bit hash value that point to previous block.

The block body is composed of a transaction counter and transaction. Figure 3 depicts the basic structure of a block. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transaction.

- **Digital Signature**: Each user of the blockchain owns a pair of private key and public key. Private Key is used to sign the transaction and that shall be kept in confidentiality. There are 2 phases: - signing phase and verification phase. Figure 4 depicts the block network, block, databases and transactions. e.g.: For instance, a user John wants to send another user Max a message.
  - In the signing phase John encrypts his data with his private key and sends Max the encrypted result and original data.
  - In verification phase, Max validates the value with John's public key. In this way Max could easily check if the data has been tampered or not.

Mining Techniques

Mining is the process of creating blocks, that blocks will be permanently attached to the database of the blockchain. In some of the blockchain applications, the miners who creates the first valid block for blockchain is rewarded, like in bitcoin. This reward is given by the system and is generally in term of money for financial applications (Salman, 2016). Mining is one of the critical concepts in the blockchain technology. It allows nodes to create blocks which will be validated by others as well (Kaushik, 2017). If the new block is valid, it will be attached to the database. Nodes that try to create the new blocks are called "mining nodes". The mining nodes race to validate the transactions and create a new block as fast as they can to win the reward.

Figure 3. Basic structure of a block

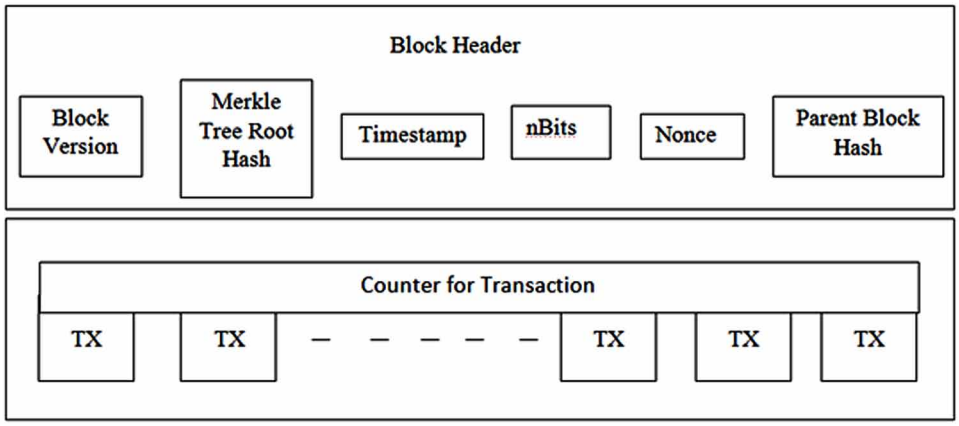
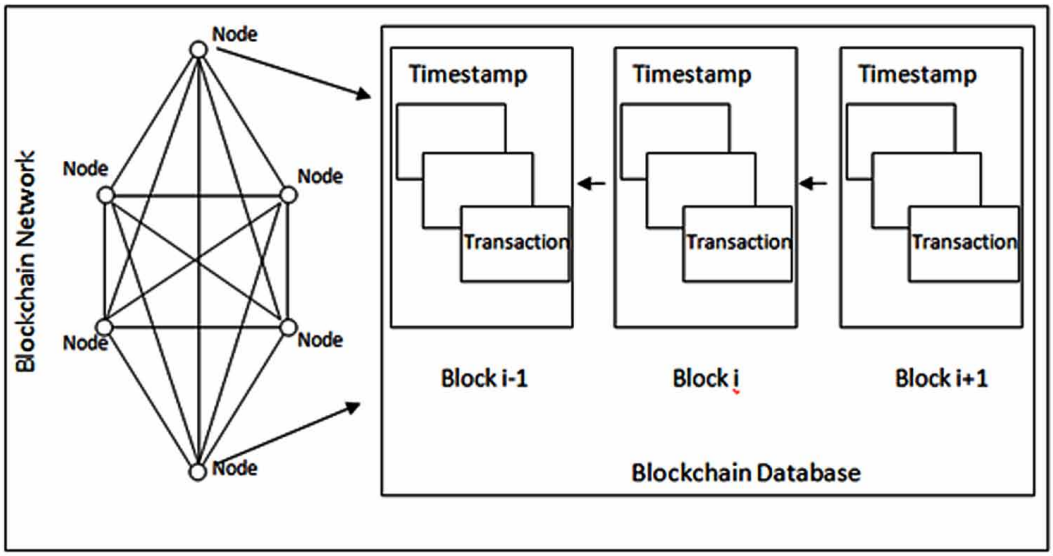


Figure 4. Blockchain network, database, blocks and transaction



There are so many approaches exist to decide which miner wins as follows:

- **Proof of Work (POW)**: POW is the mining technique used in Bitcoin and is currently used by many other blockchain technologies. In this technique, mining nodes need to solve the hard-mathematical puzzle. Once the nodes validate the transaction and solves the puzzle, the block is added to the blockchain network. Other mining nodes validate the block to make sure the submitted block is not false. Once all the miners agree that the block is legit, that block will be added to the blockchain and submitter mining node is rewarded. The agreement here is based on a majority consensus. Thus, it is difficult to fake unless the attackers compromise more than 50 percent of mining nodes.
- **Proof of Stake (POS):** Unlike POW, in POS mining nodes are not required to solve a computational mathematical puzzle. In this miner or next block creator is chosen in a random way. Here if the nodes wealth or stake is high that node has the chance being chosen to create the new block. In other words, more money a node has, higher its chance to mine the block. POS does not reward the miner in native version, but extended versions of POS awards and punish the miners based on their performance. In this method selection is totally based on richest account, this may result that a single account is handling all the creations. Hence it is unfair to all other miners and may lead to unfair distribution or even centralization. Thus, a randomized node selection and a coin age-based selection process have been proposed. In coin age-based method, the users that have unspent coins and not created any block from past 30 days are considered for mining.
- **Proof of Space:** Proof of space is similar to proof of work except that the puzzle requires a lot of space. In this technique, the mining needs need to have a high storage capability instead of having a high computational capability. Several theoretical and practical implementations of POSPACE have been released however the challenge is high requirement of memory space.

- **Proof of Importance:** This mining technique calculates the significance of an individual node based on the transaction amount and the balance of that node. It assigns a priority with a hash calculation to more significant nodes. Then the node with the highest priority is chosen for the next block creation.
- **Measure of Trust:** This is another way to perform mining. It uses the dynamic trust measurements and selects the node with the highest trust level as the block initiator. The trust worthiness is based on the node's behaviors; therefore, good behaving nodes that follow the protocols are rewarded. The trustworthiness is approximated by the history of good and bad actions that the node has taken so far. If specific node plans to increase its trustworthiness for several interactions in order to attack the network later. The MoT approach could be subject to malicious attacks.
- **Minimum Block Hash:** In this approach, a miner is chosen randomly and not based on its resources. The system selects the miners based on a generated minimum has value across the entire network. The next miner thus is selected randomly and the probability of selecting the same miner is low. This approach was implemented on a modified Bitcoin network and it was shown to offer energy saving for mining.

THE CONSENSUS PROBLEM

The Byzantine Generals Problem

The Byzantine Generals Problem (BGP) is a problem related to communication failure like how can a node ("general") in a system be certain that the information they are receiving are valid (Bach, 2018). This problem includes an imaginary general who makes a decision to attack or retreat and try to communicate with lieutenants. They are traitors including General. Traitors cannot be relied upon the communicate order. They may alter message in attempt to follow the process. Here the generals are known as processes. Source process is the General who initiate the order and the orders that send to the other processes are messages. Here faulty processes represented by traitor generals and lieutenants and correct processes are represented by loyal generals and lieutenants. This is BGP which is applicable to every distributed network. It is more complex in bitcoin network as there is no true "general" or server. All participant nodes need to agree upon every message that is transmitted to the nodes. If the group of nodes is corrupt or the message which they transmit is corrupt, then the network should not be affected by it and should resist this 'attack'. The network should entirely agree upon every message transmitted in the network. This agreement is called consensus.

Byzantine Fault Tolerance

A Byzantine Fault is a faulty operation/algorithm that occurs in a distributed system. These faults can be classified as Omission failure and Execution failure. A failure of not being present is called Omission failure such as failing to respond to a request or not receiving a request. A failure due to sending incorrect or inconsistent data, responding to a request incorrectly is known as Execution fault. Byzantine fault tolerance can guarantee the safety and liveliness of a system given that no more than [(n-1)/3] replicas are faulty over the systems lifetime (Bach, 2018). When n is the total number of replicas within a sys-

tem. So Byzantine Fault Tolerance can handle up to 33% of faulty nodes. Up to 3f+1 replica to reduce to 2f+1 required replica in order to provide safety and liveliness in a system where f is total number of faulty replicas contained within the system (Bach, 2018).

Delegated Byzantine Fault Tolerance (dBFT)

Delegated Byzantine Fault Tolerance (dBFT) is a variant of standard BFT. There is a simple analogy to explain how dBFT works. There is a country called Neo. Every citizen in this country has right to vote to select the leader known as delegate. The delegate makes laws for the country. If the citizen not agreed with how a delegate voted on a law, then they can vote for different delegate next time. Citizen tells the delegate what makes them happiest. Delegates must follow and keep track of the demand of the citizen and document it on the ledger. A speaker is randomly assigned from the group of delegates when it is time to pass a law. Then speaker process the law. Speaker calculates how the law affects on the Happiness Number of countries in the speaker's proposed law. Then speaker hands out the proposed law to the delegates. Then delegates decide if the speaker's calculation matches to their own calculation, they confer with other delegates to verify the calculated Happiness Number is valid. If 66% of the delegates agree that the calculation is valid then law passes and is finalized. If less than 66% of delegates are agreed, then the new speaker is randomly selected for the process. Likewise, in blockchain, delegates represent Bookkeeping nodes. Bookkeeping nodes verify each transaction. Citizen represent ordinary node which does not take part in validation. So, in the blockchain, if 66% of the Bookkeeper agrees that the transaction is valid then it is permanently attached to the blockchain.

THE CONSENSUS ALGORITHM

There are currently over 1500 active crypto currencies. Here is some high-profile consensus algorithm as follows:

Proof of Work (PoW)

Proof of Work algorithm is most widely used algorithm. This algorithm is used by crypto currencies such as bitcoin and ethereum, each one with its own differences (Sankar, 2017). PoW algorithm is used to confirm transaction and produce new block in the blockchain. Using Proof of Work, miners compete against each other to complete the transaction on the network and get rewarded. In this algorithm, main work is to solve the mathematical puzzle. Now what is mathematical puzzle? It is an issue that requires a lot of computational power to solve. Miners solve the puzzle then confirm the transaction and form the new block.

When other nodes confirm that the transaction is valid then only block is added to the chain permanently. The problem should not be too complex to solve, if it is like that, block generation takes a lot of time. But in other scenario if problem is too easy, it is prone to DOS attack and spam. The solution needs to be easily checked by other nodes otherwise not all nodes are capable of analyzing that calculation is correct. Thus, it will have to trust other nodes and it violates the one of the important features of

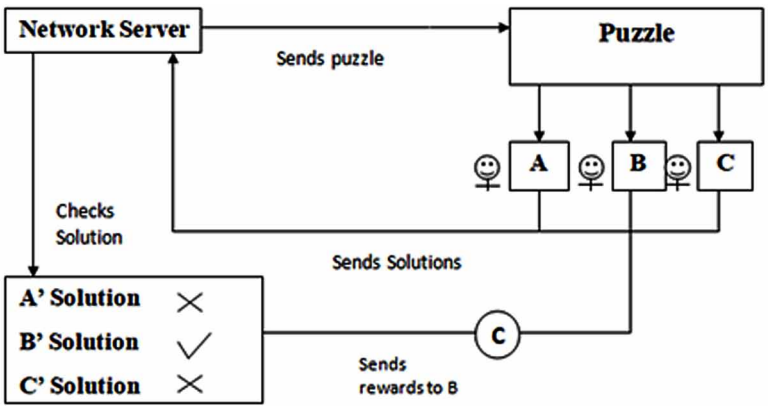
blockchain - transparency. Complexity of puzzle is depending on the number of users, the current power and the network load. The hash value of each block contains the previous block's hash value which increases the security. The genesis block is an exception as it has no parent block, so its hash value is completely zeros (Mingxiao, 2017).

Bitcoin is the foundation of this kind of consensus. The puzzle called as Hashcash. The Proof of Work algorithm allows changing the complexity of a puzzle based on the total power of the network. The average time of the formation of any block is approx 10 minutes. The main disadvantages of this algorithm are huge expenditure, uselessness of computations and 51 percent attack (Tosh, 2017). Figure 5 depicts the working of the proof of work algorithm.

Proof of Stake (PoS)

Proof of Stake has been mentioned in the first bitcoin project, but it was not used in bitcoin because of its robustness and other reasons (Mingxiao, 2017). It is different from Proof of Work algorithm in which hashing algorithm is used to validate the transaction. Proof of Stake is most commonly used as the replacement of PoW in PeerCoin. Traditionally in this algorithm, selection of miners was based on the account balance i.e higher the balance in account, higher the chance to become the miner. So, the richest person has probability to become permanent miner as it has high balance in his account and that leads to be unfair to other persons. So, this process leads to the centralization that is why several other processes of selection has been devised. In peer coin crypto currencies, proof of Stake combines the randomization with the concept of "coin age". The formula is proofhash< coin age *target (Mingxiao, 2017). Coins that have been unspent for at least 30 days, being competing for the next block. To signing the next block, the older and largest set of coins have greater possibility. Then once the Stake of coin has been selected to sign a block, they must start with zero 'coin age' and then wait at least 30 more days before signing another block. So, this process secures the network and produce new coin without much computational power. Proof of Stake is more efficient than Proof of Work which mainly relies on energy use (Chalaemwongwan, 2018).

Figure 5. Working of proof of work (PoW) algorithm



Delegated Proof of Stake

Satoshi Nakamoto hoped that all the participants can use the CPU to mine in initial design stage. So, the hashing power could match the nodes and each node has the opportunity to participate in blockchain. Then finally, the machines that are specially designed for mining are invented. In a delegated proof of stake (dPOS) system, stakeholders vote to elect any number of witnesses to generate blocks (Bach, 2018). During each maintenance interval, the roster of witnesses is shuffled. Each witness has a turn to produce a block at the fixed schedule of 1 block per n number of seconds. Witnesses are rewarded for each block produced. The witness may be voted out in future election when they fail to produce a block after being elected. The N number of witnesses creates new blocks as assigned and then need to ensure some fixed online time. BitShare is an example of dPoS system. The blockchain using DPoS is more efficient and power saving than PoS and PoW.

Proof of Importance

POI uses a method that clusters through transaction graph analysis, utilizing the transaction quantities and the balances of individual nodes as indicators, determining the importance of each node and designating the priority using hash computations to more significant nodes (Tasatanattakool, 2018; Chalaemwongwan, 2018). Proof of Importance was first introduced by NEM. PoI uses a mechanism that determines which network participants (nodes) are eligible to add a block to the blockchain and it is a process that is known as "harvesting" by NEM. Nodes are able to collect the transaction fees in exchange for harvesting a block. Those accounts who has higher importance score will have a higher probability of being chosen to harvest a new block. Proof of importance uses an underlying cryptocurrency called XEM. Each account has a XEM balance within the NEM network. The balance is split into two parts: vested and unvested. When an account receives XEM, the new XEM is added to this account's unvested balance. An account's unvested balance of one tenth is moved into the vested part every 1440 blocks (Bach, 2018). In addition, when an account sends XEM, XEM is taken from both vested and unvested balances in order to retain the same vested and unvested ratio. An account must hold at least 10,000 vested XEM to be eligible for an "importance calculation".

Practical Byzantine Fault Tolerance

Byzantine fault tolerance can be a good method to solve the transmission errors in distributed systems (Wang, 2019). But previously Byzantine system requires exponential operations. In 1999, the PBFT system was proposed and the algorithm complexity was reduced to a polynomial level, this improved efficiency (Mingxiao, 2017).

Ripple Transaction Protocol

Ripple is based around the XRP ledger which is a shared public ledger, that was a consensus process that allows for payments, exchanges and remittance in a distributed process. Ripple payment protocol was first developed in 2004 by Ryan Fugger. The ripple transaction protocol (RTP) was developing by OpenCoin based on Ryan Fugger's concept. The Ripple transaction Protocol enables the direct transfer

of money between two parties. Ripple enables security, is fast. It was structure of distributed public ledger of the kind that utilizes consensus procedures which permit payments, exchanges and remittances.

COMPARATIVE ANALYSIS OF ALGORITHMS

Different consensus algorithms of blockchain have different strengths and drawbacks. Table 1 shows comparative analysis of these algorithms for some essential properties of blockchain.

1. **Energy Saving:** In Proof of Work, mining nodes need to solve the mathematical puzzle continuously which leads to consume high computational power. Therefore, the amount of energy is immense (Ogiela, 2018). But in PoS and DPoS, miners work decreases as the search place is produced to be restricted. Regarding PBFT and Ripple, there is no mining in terms of consensus Strategy. Therefore, it saves energy.
2. **Data Model:** A data model is a transaction that focuses on assets. All systems require specific configurations, with several organizations being able to spin up a network to exchange assets with each other (Dinh, 2018). These organizations are known as ledger owners. Ripple issue their own token assests and provide their ledger as a method of exchange.
3. **Application:** Some ledger support running general, user-defined computations. Ethereum and its derivatives, namely Hyperchain, Quorum, Monax, Parity and Definitely let users write arbitrary business logic executed on top of the ledger (Chalaemwongwan, 2018).
4. **D.Examples: -**Bitcoin and ethereum uses Proof of Work algorithm. Peercoin only focus on Proof of Stake in other hand ripple which has account- based data model uses Ripple Transaction Protocol.

Table 1. Comparative analysis of blockchain algorithms for a set of blockchain properties

Parameters/ Algorithms	PoW	PoS	PoI	DPOS	PBFT	Ripple
Developer	Markus Jakobsson and Ari juels	Peercoin	NEM	Danial Larimer	Castro and liskov	Jed McCaleb and Chris larsen
Year	1999	-	2015	2014	1999	2012
Node identity	Public	Public	Public, Private	Public	Private	Public
Computational Power	High	Comparative low	low	low	low	low
Energy Efficiency	No	Partial	yes	Partial	Yes	yes
Data model	Transaction -based	Account - based	Transaction-Based, Account-Based	Transaction-based, Account-Based	Key- value	Account-based
Language	C++, Golang, Solidity, LLL	Michaleson	Java	No scripting	GoLang, java	Java, Go, c++
Applications	Crypto-currency, General application	Michaleson Application	Blockchain Platform	Decentralized Exchange	General Application	Digital Assets, payment
Examples	Bitcoin, Litecoin, Ethereum, ZCash	Peercoin, Tezos, Tendermint	XEM	Bitshares	Hyperledger	Ripple

IMPLEMENTATION

In the implementation section we have taken blockchain structure somewhat similar to the actual blockchain which is used by bitcoin. This implementation section shows the basic structure of blockchain in which we need to create wallet for account holders and make transaction between them. These successful transactions are needed to be verify and then group together to create blocks. This shows that the blocks are nothing but transactions data which is made by users. These blocks are added to the blockchain after verification. Stages of execution of bitcoin-like Blockchain are as follows:

Hash Function and Mining

In actual scenario, Bitcoin uses two rounds of SHA256 hash function. In this example, we will use one SHA256 hash function (Ishan, 2018). Hash function will change a string of arbitrary length into a fixed-length string of 64 hexadecimal characters. Now the process of mining is, when given an arbitrary string x, then find a nonce such that hash(x+nonce) produces a hash string with the number of leading ones. In our example here, we'll "mine" a nonce so that the hash of our message ("hello bitcoin") when concatenated with our nonce will have at least 2 leading ones. Figure 6 depicts the snippet of finding nonce.

After this we are defining two functions: one to hash a string and one to mine a nonce for a given string. Figure 7 showing that the number of iterations required for a difficulty of 3 is much larger than for a difficulty of 1.

Creating a Wallet and Doing Transaction

In bitcoin a wallet is a private and public key pair. The public key is used to receive transactions and the private key is used to spend money. Wallet is more complicated in real scenario. It is a set of multiple private/ public key pairs and an address is not directly the public key. This ensures better privacy and security in Bitcoin.

Figure 6. Finding nonce and return a string

```
In [5]: import hashlib
        import random
        import string
        import json
        import binascii
        import numpy as np


        import logging

In [6]: def sha256(message):
            return hashlib.sha256(message.encode('ascii')).hexdigest()

In [7]: message = 'hello bitcoin'
        for nonce in range(1000):
            digest = sha256(message + str(nonce))
            if digest.startswith('11'):
                print('Found nonce = %d' % nonce)
                break
        print(sha256(message + str(nonce)))

        Found nonce = 32
        112c38d2fdb6ddaf32f371a390307ccc779cd92443b42c4b5c58fa548f63ed83
```

Figure 7. Showing number of iterations required for difficulty of 3

```
In [8]: def dumb_hash(message):
            """
            Returns an hexadecimal hash
            """
            return sha256(message)


        def mine(message, difficulty=1):
            """
            Given an input string, will return a nonce such that
            hash(string + nonce) starts with `difficulty` ones

            Returns: (nonce, niters)
                nonce: The found nonce
                niters: The number of iterations required to find the nonce
            """
            assert difficulty >= 1, "Difficulty of 0 is not possible"
            i = 0
            prefix = '1' * difficulty
            while True:
                nonce = str(i)
                digest = dumb_hash(message + nonce)
                if digest.startswith(prefix):
                    return nonce, i
                i += 1

In [9]: nonce, niters = mine('42', difficulty=1)
        print('Took %d iterations' % niters)

        nonce, niters = mine('42', difficulty=3)
        print('Took %d iterations' % niters)

Took 23 iterations
Took 2272 iterations
```

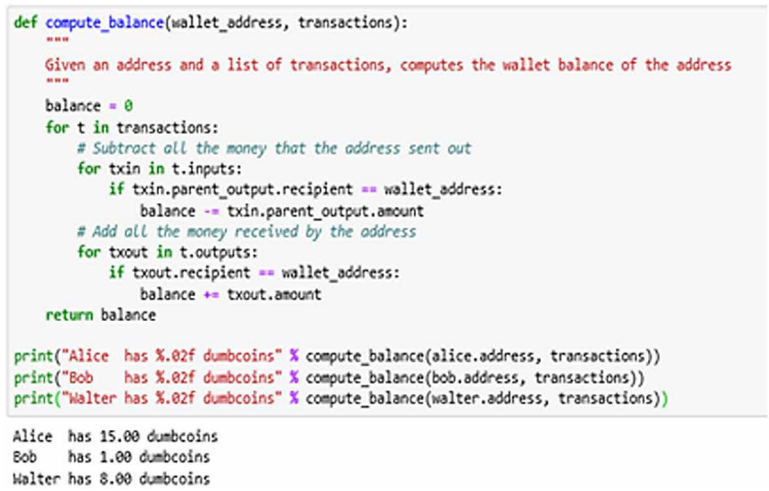
In our example, we are using single key-pair and use the public key as the address. After creating the Wallet, we will then transfer money from one account to another.

Figure 8 is showing the wallet's balance.

Putting Transactions in Block

When we will create wallet and do some transaction between them, we need to group the transactions into the block. Then miners need to mine the block. Mining a block generally consist of two parts: Verifying the transactions in the block and Finding a nonce such that the block's hash starts with a number of 0. Mining generates money by the convention that the first transaction in a block is a Genesis block which does not have parent hash value. Transaction that gives 25 coins to whatever addresses the miner chose.

Figure 8. Code snippet of showing balance of wallet



```
def compute_balance(wallet_address, transactions):
    """
    Given an address and a list of transactions, computes the wallet balance of the address
    """
    balance = 0
    for t in transactions:
        # Subtract all the money that the address sent out
        for txin in t.inputs:
            if txin.parent_output.recipient == wallet_address:
                balance -= txin.parent_output.amount
        # Add all the money received by the address
        for txout in t.outputs:
            if txout.recipient == wallet_address:
                balance += txout.amount
    return balance

print("Alice  has %.02f dumbcoins" % compute_balance(alice.address, transactions))
print("Bob    has %.02f dumbcoins" % compute_balance(bob.address, transactions))
print("Walter has %.02f dumbcoins" % compute_balance(walter.address, transactions))

Alice  has 15.00 dumbcoins
Bob    has 1.00 dumbcoins
Walter has 8.00 dumbcoins
```

The miner can add transactions to redirect the fees from the transactions in the block to whatever address it chooses. Figure 9 shows creating blocks such that first block is genesis block, second is block1 and third is block2 so on. Figure 10 shows that verification of blocks in terms of true false.

CONCLUSION

Blockchain with its key characteristics, has shown its potential to reshaping traditional industry. In this paper, we first introduce what is blockchain and bitcoin and how bitcoin gave birth to blockchain. Then the core concept and architecture of blockchain is explained. We have mentioned some mining techniques used by blockchain. Mining technique plays essential role in Blockchain. Then discuss consensus algorithm of blockchain regarding theoretical aspect of algorithm and its advantages and disadvantages.

Figure 9. Code snippet of putting transaction in block

```
alice = Wallet()
bob = Wallet()
walter = Wallet()

genesis_block = GenesisBlock(miner_address=alice.address)
print("genesis_block : " + genesis_block.hash + " with fee=" + str(genesis_block.fee()))

t1 = genesis_block.transactions[0]
t2 = Transaction(
    alice,
    [TransactionInput(t1, 0)],
    [TransactionOutput(bob.address, 5.0), TransactionOutput(alice.address, 15.0), TransactionOutput(walter.address, 5.0)]
)
t3 = Transaction(
    walter,
    [TransactionInput(t2, 2)],
    [TransactionOutput(bob.address, 5.0)])

t4 = Transaction(
    bob,
    [TransactionInput(t2, 0), TransactionInput(t3, 0)],
    [TransactionOutput(walter.address, 8.0), TransactionOutput(bob.address, 1.0)]
)

block1 = Block([t2], ancestor=genesis_block, miner_address=walter.address)
print("block1        : " + block1.hash + " with fee=" + str(block1.fee()))

block2 = Block([t3, t4], ancestor=block1, miner_address=walter.address)
print("block2        : " + block2.hash + " with fee=" + str(block2.fee()))
```

```
genesis_block : 1162dce8ffec3acf13ce61109f121922eee8cceeea4784aa9d90dc6ec0e0fa92 with fee=0
block1        : 112c9894c0ebf0e33709d73e5d294a0b39f63034e0827f2c95d8c490e79bf50d with fee=0.0
block2        : 11e37778d7935029bd94d3aa188a8a6cc540a35227415abef8b8cd1d7bc2c81a with fee=1.0
```

Figure 10. Verification

```
block2 = Block([t3, t4], ancestor=block1, miner_address=walter.address)
print("block2        : " + block2.hash + " with fee=" + str(block2.fee()))
```

```
genesis_block : 1162dce8ffec3acf13ce61109f121922eee8cceeea4784aa9d90dc6ec0e0fa92 with fee=0
block1        : 112c9894c0ebf0e33709d73e5d294a0b39f63034e0827f2c95d8c490e79bf50d with fee=0.0
block2        : 11e37778d7935029bd94d3aa188a8a6cc540a35227415abef8b8cd1d7bc2c81a with fee=1.0
```

```
In [20]: verify_block(block1, genesis_block)
         verify_block(block2, genesis_block)
```

```
Out[20]: True
```

We analyzed and compare those algorithms on the basis of some essential properties of blockchain. In this paper, we focused on public blockchain which is used by bitcoin and the widely used algorithms like Proof of Work and Prof of Stake. Proof of Work still has some limitations such that 50% attack, huge expenditure and uselessness of computational power in other hand Proof of Stake is more energy efficient as it uses less computational power. There are so many algorithms proposed but still blockchain has some limitations.

FUTURE RESEARCH DIRECTION

Blockchain is chain of blocks which contains the information of transactions. In this paper we have discussed about some mining techniques and architecture of blockchain. We have also discussed use for algorithms and its limitations. Nowadays Blockchain is growing faster but it still has many limitations such as redundancy, complexity, energy and resource consumption, security flaws etc. One of the limitations of blockchain is storage issue. The future research direction would be solution for storage issue of blockchain. What will happen if blockchain is combined with cloud technology? How it will affect the storage and security issue of blockchain.

REFERENCES

Bach, L. M., Mihaljevic, B., & Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1545-1550. 10.23919/MIPRO.2018.8400278

Chalaemwongwan, N., & Kurutach, W. (2018). State of the art and challenges facing consensus protocols on blockchain. *International Conference on Information Networking (ICOIN)*, 957-962. 10.1109/ICOIN.2018.8343266

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering*, *30*(7), 1366–1385. doi:10.1109/TKDE.2017.2781227

Ishan, P. B., & Rai, G. (2018). Analysis of Cryptographic Hash in Blockchain for Bitcoin Mining Process. *International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 105-110. 10.1109/ICACCE.2018.8441688

Jutila, L. (2017). *The blockchain technology and its applications in the financial sector*. Available at: https://aaltodoc.aalto.fi/bitstream/handle/123456789/27209/bachelor_Jutila_Laura_2017.pdf;jsessionid=EB73ECF52889104CB772C6FA3B968EF7?sequence=1

Kaushik, A., Choudhary, A., Ektare, C., Thomas, D., & Akram, S. (2017). Blockchain — Literature survey. *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2145-2148.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017). A review on consensus algorithm of blockchain. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2567-2572. 10.1109/SMC.2017.8123011

Ogiela, M. R., & Majcher, M. (2018). Security of Distributed Ledger Solutions Based on Blockchain Technologies. *IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA),* 1089-1095. 10.1109/AINA.2018.00156

Salman, T., Zolanvari, M., Erbad, A., Jain, R., & Samaka, M. (2016). Security Services Using Blockchains: A State-of-the-Art Survey. *IEEE Communications Surveys and Tutorials*.

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1-5. 10.1109/ICACCS.2017.8014672

Singh, S., & Singh, N. (2016). Blockchain: Future of financial and cyber security. *2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 463-467. 10.1109/IC3I.2016.7918009

Tasatanattakool, P., & Techapanupreeda, C. (2018). Blockchain: Challenges and applications. *International Conference on Information Networking (ICOIN)*, 473-475.

Tosh, D. K., Shetty, S., Liang, X., Kamhoua, C., & Njilla, L. (2017). Consensus protocols for blockchain-based data provenance: Challenges and opportunities. *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 469-474.

Wang, Y., Cai, S., Lin, C., Chen, Z., Wang, T., Gao, Z., & Zhou, C. (2019). Study of Blockchains's Consensus Mechanism Based on Credit. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 10224–10231. doi:10.1109/ACCESS.2019.2891065

Yuan, Y., & Wang, F. (2018). Blockchain and Cryptocurrencies: Model, Techniques, and Applications. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, *48*(9), 1421–1428. doi:10.1109/TSMC.2018.2854904

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *IEEE International Congress on Big Data (BigData Congress)*, 557-564.

KEY TERMS AND DEFINITIONS

Bitcoin: Is a cryptocurrency based on blockchain technology that enables it to function as a medium of exchange without involving the intermediary, such as a bank.

Decentralized Ledger: Is ledgers or system of records for business economic activities and interest that are dispersed instead of reliant on and housed within one third-party system, such as a financial institution.

Double Spending Problem: Which is the risk, particularly when digital currency is exchanged, that a person could concurrently send a single unit to two different sources.

Ethereum: Is an open source, public, blockchain-based distributed computing platform and operating system featuring smart contract functioning.

Hash Function: It takes a set of digital data and delivers a numeric piece of data with a fixed range. If you deliver a same exact data to a hash function, it will deliver the same exact numeric piece of data every time. If the data input varies even by one variable, the hash function output will change.

Nounce: Is a number chosen at random used once for a specific purpose and then discarded.

Peer-to-Peer Network: Is a computer network based on nodes (e.g., computers that are maintaining the network worldwide). It is a decentralized network where nodes share information with each other without anyone controlling the network.