Introduction to blockchain technology with Bitcoin protocol

Babu Pillai[1,2] and Zhé Hóu[2]

[1] Southern Cross University, Australia
[2] Griffith University, Australia
pillai.babu@outlook.com

Abstract. The engineering behind the technology that powers Bitcoin, known as Blockchain, has gained attention as a potential software solution for various industrial applications. The interest in this technology is growing due to its potential to revolutionize digital transactions. Although Blockchain technology has evolved greatly in the past decade, the development of applications beyond Bitcoin has not yet kept pace. In this chapter, we will examine the basic design principles of Blockchain technology using Bitcoin's architecture as a foundation, and understand the rationale behind its design limitation of scalability.

Keywords: distributed ledger technology · blockchain · blockchain scalability trilemma

1 Overview of distributed ledger technology

DLT (Distributed Ledger Technology) is a rapidly developing concept that deals with the process and technologies allowing a network of nodes to reach an agreement on the status of a distributed ledger [20]. In contrast to traditional databases, there's no central storage or administration in a distributed ledger, which is a set of synchronized databases among a network of nodes. DLT operates based on the principle of distributed consensus, where each node verifies, processes, and confirms every transaction [5]. The transactions are then recorded in the distributed ledger and synchronized across the network once a consensus is reached. The challenge lies in ensuring that all nodes maintain the same record in their ledgers, despite potentially *selfish*, *faulty*, or *malicious* nodes in the network.

Definition 1 (Distributed system) *A distributed system is a collection of independent entities that communicate through a communication medium to achieve a common goal and appear as a single coherent system to its users.*

The entities in DLT are referred to as nodes. They can either be hardware devices or software processes. In practice, they are independent nodes but program to achieve a common goal by communicating through a communication medium.

Reaching consensus in a distributed system can be challenging due to communication issues and byzantine network partitions. These can result in different nodes having differing views of the data, making agreement on what the correct data is difficult. The presence of malicious nodes in the network can further complicate matters as they may attempt to disrupt the consensus process or deceive other nodes for their own benefit. Designing a consensus algorithm that is both efficient and secure is a significant challenge. Despite these difficulties, DLT systems offer various potential benefits.

1.1 Advantages and limitations of DLT systems

A distinct characteristic of the DLT system is its redundancy and resilience which eliminates the potential for a single point of failure through the use of several nodes/validators that can validate, provide the same service, or both. DLT systems can also be faster than single-computer systems i.e scaling through to improve the performance of systems by expanding or increasing hardware resources. A fundamental problem in DLT systems is to make nodes agree on a given data value known as the *consensus problem.* Traditional DLTs rely on Byzantine Fault Tolerance (BFT) protocols to reach consensus. Typically, the BFT protocol is designed to tolerate 51% of malicious nodes[3] and works under weak synchronous network assumption. However, in digital space, it fails to solve the *double-spending* problem [8, 14]. Improved versions of BFT, Practical Byzantine Fault Tolerance (PBFT) [7] and its derivatives [1, 16] focus on providing a practical Byzantine state machine by a leader, or group of entities to address the double-spending problem. While there are several implementations in existence the most prominent implementation is blockchain technology. It is able to securely maintain state changes in a permission-less environment. Several principles are combined in blockchain technology to enable value exchange in a DLT system and solve the *double spending* and *consensus* problems.

The limitations of the DLT system are expressed using *CAP theorem* [12]. The theorem defines three properties i) *Consistency*: property ensures all nodes have the identical copy of data; ii) *Availability*: property ensuring that the system is accessible all the time; and iii) *Partition tolerance*: property ensures the system continues to operate even if there are some node/link failures.

Distributed systems are prone to network failures that generally cause partition; therefore, the choice is between consistency and availability in the presence of partition. However, when choosing consistency over availability, the system returns an error if the information is not up-to-date. In contrast, when choosing availability over consistency, the system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up-to-date due to network partitioning. In the absence of partitioning, the properties of availability and consistency can both be guaranteed.

1.2 Blockchain-based DLT systems

Blockchain-based systems have distinct characteristics that set them apart from other DLT systems. Unlike traditional databases, they bundle transactions into

[3] nodes failing or propagating incorrect information to other peers

blocks that are cryptographically linked to form a chain of data. The incentive mechanism encourages nodes to agree on the same chain as new, valid blocks are added. The blockchain feature in DLT is a novel approach to improving business processes. Through decentralization among consumers, businesses, and governments, it promotes trust and securely stores transaction data in a reliable manner.

The first generation blockchain, often called Blockchain 1.0, known as Bitcoin [18] is a peer-to-peer electronic cryptocurrency system. Bitcoin proved the possibility of trust-less peer-to-peer transactions on a public network without a trusted third party [28]. However, it is limited to cryptocurrency-based applications. The second generation blockchain, often called Blockchain 2.0, developed as a programmable transaction platform using smart contracts [17]. This capability opened up new opportunities such as a contract being deployed on the blockchain and autonomously triggering transactions when key supply milestones are met. This combination of security and immutability provides an integrated level of integrity to the contract process.

2 Introduction to blockchain technology

The concept of blockchain technology was introduced in October 2008 as part of a proposal for *Bitcoin* which is a decentralised cryptocurrency system. A paper titled *Bitcoin: A Peer-to-Peer Electronic Cash System*, written under the alias of Satoshi Nakamoto was published and later used as a reference for the Bitcoin implementation [18]. The paper elaborates on the concept of a virtual currency system that operates without a central authority to issue or transfer the currency. Unlike the traditional system, it operates on a decentralised network of computers, which are purely driven by software protocols. The data is cryptographically protected and replicated among this network of computers. Each new update will go through a verification process before it reaches a consensus.

Bitcoin was the first application built using blockchain technology and introduced a digital currency called *Bitcoin*. This digital currency is stored in a wallet software and controlled by private and public keys. There are two types of participants in the Bitcoin application: users who transact (send and receive) with the digital currency, and miners who process transactions through a process called mining. Mining involves verifying each transaction and solving a mathematical puzzle to create a block.

At the core of the Bitcoin system is a decentralized network of peer-to-peer nodes that validate transactions directly between users, without intermediaries like banks. Every transaction is verified by the network of nodes and then grouped into blocks and recorded in a publicly distributed ledger.

The main aim of Bitcoin was to create a value transfer system that isn't controlled by a trusted third party, such as a bank. This system focuses on a transparent ledger of transactions managed and controlled by the participants. Every time a Bitcoin owner makes a payment, the transaction is broadcast over the network and recorded in the blockchain, ensuring immutability.

2.1 The architecture of a blockchain-based system

The architecture of blockchain is diverse and can be explained from various perspectives. The original term blockchain refers to a public, decentralised, permission-less system. The architecture of such blockchains can be described as a distributed append-only database, which functions as a single system on a public network of linked computers. Such systems are capable of performing transactions between non-trusted parties without going through a trusted intermediary.

In a blockchain-based system, each record of transactions is maintained across several computers that are linked in a peer-to-peer network. Unlike other data structures, in blockchain, the data is segmented into blocks, which are cryptographically interconnected in chronological order. These blocks are an append-only data structure, meaning that the data added to the block are never removed or updated. Instead, the update basically adds a new transaction in the block. For example, If *Bob* owns 10 bitcoins that he received from a single transaction, in Block # n, this is recorded on the ledger. If *Bob* transfers 5 bitcoins to *Alice*, that will be a new transaction. In the new transaction, in Block # n + 1 Bob needs to use the entire output (10 bitcoins) of the previous transaction as an input. Apparently, as shown in Figure 3, Bob specifies two outputs: One refers to the transfer of 5 bitcoins to *Alice*, and the other refers to 5 bitcoins back into his account. Similarly, if *Alice* wants to transfer her 5 bitcoins, she repeats the same procedure. However, she can only spend the output of the previous transaction.

In a blockchain, *accounts* are like the placeholder of stateful objects that have a corresponding address. The blockchain system will keep track of the *state* of the tokens. Public and private key pairs control these accounts. Bitcoin's *state* is represented by its global collection of unspent transaction outputs (UTXO). The transfer of value in bitcoin is actioned through transactions.

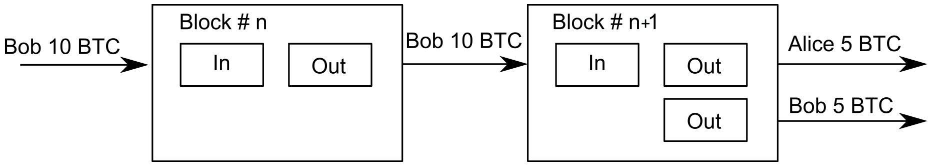

Fig. 1. Simplified structure of a block

The data recorded on the blockchain consists of a transaction history that starts with the genesis block. Each subsequent block contains all transactions that have taken place in the network and reference an earlier block in the chain. This creates a comprehensive and continuous record of all transactions.

When a transaction is made, it is broadcast to the entire network and made publicly available. Other computers on the network will verify the transaction and if it is deemed valid, it will be included in the next block. The new block will then be published to the network and all other nodes will also validate the transactions it contains. Once a majority of nodes agree on the validity of the new block, it becomes part of the main blockchain. As the blockchain grows, the data recorded on it becomes more secure and permanent.

2.2 Blockchain value transfer protocol

In blockchain technology-based systems, transactions are the only atomic event allowed by the underlying protocol to update the state of the system [3]. The application features the user to update the state by extending through a new block. A blockchain records all successful transactions in its state. Each previous transaction can be reviewed at any time but cannot be updated without a collective effort. A new transaction entry must refer to its previous transaction that everybody has agreed to in which the user now transfers itself to the recipient of a previous transfer [11, 19]. Figure 2 shows the main concept of a Bitcoin transaction where each input relates to a previous output and each output waits as an Unspent Transaction Output (UTXO) until a later input spends it. The genesis block and the block mining reward (Coinbase transaction) are the only exceptions to this process.

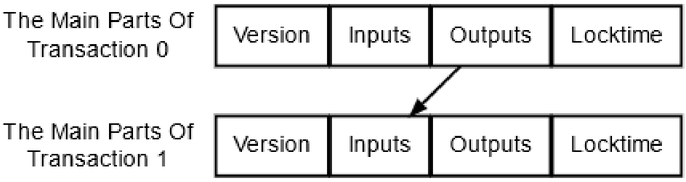
The Main Parts Of Transaction 0
Version
Inputs
Outputs
Locktime
The Main Parts Of Transaction 1
Version
Inputs
Outputs
Locktime

Fig. 2. Bitcoin transaction

It is also important to understand how the transaction process works. For example, when Bob wanted to send bitcoins to Alice, they both corresponded to their private and public key. Blockchain address is mathematically derived from the public key. This address will be used as the designation address in the transaction process, and the private key is used to unlock the account. To send bitcoin, Bob must refer to the block where the transaction corresponds with his public key. By signing that transaction with Bob's private key, Bob verifies the ownership and authorises the transfer to Alice's public address. In each transaction process, the sum of input transaction units must add up to the amount of output. If the amount Bob is sending to Alice is more than the input value, Bob must create a new transaction corresponding to the remainder back to his account. In blockchain this is known as a change account otherwise the difference will be automatedly given to the miner.

Blockchain operates under the rigid assumption of decentralised consensus. Mining is a process of extending the chain by proposing new blocks through consensus. Nodes confirm transactions by mining blocks of transactions through a defined way[4] that is a function of the transactions with the history so far. More precisely, each participant maintains its own local history of block records and each block b consists of a tuple $(H(b-1), m, n)$

- where $H(b-1)$ is a pointer to the previous block in the chain,
- m is the hash of transactions contained in the block,
- n is the solution to the problem.

In the blockchain protocol, each node tries to extend its chain by performing the mining process to add new blocks of transactions. in Bitcoin, mining, which involves verifying and incorporating new blocks into the blockchain, is performed using *proof of work* protocol. Miners employ special hardware to solve proof of work mathematical puzzles, and they receive cryptocurrency as a reward for their efforts.

The typical permission-less proof-of-work (PoW) protocol transaction execution describes in the Bitcoin white paper [18] works as follows: From the pool of transactions, validator nodes called *miner*s try to assemble a block by complaining transactions with an unknown hash puzzle. The first miner to solve the puzzle is rewarded with a certain number of cryptocurrency tokens. The only way to find this puzzle is to guess and it is a computationally hard task. At the same time if a miner receives new blocks from another miner. First, that miner verifies the transaction and the puzzle's validity. If valid, extend his chain with that block and tries to solve the new puzzle with another set of transactions. The scheme elegantly justifies that transactions are validated by those who are willing to do by putting some competitive effort into an incentive. This competition enables the property of security within the ecosystem. An incentive mechanism [23] attracts miners to support the network. Specifically, the miner who successfully creates a new block is granted a block reward. Generally, blockchain can provide two incentives for miners: block mining rewards and transaction fees [6].

A block mining reward transaction is a special transaction in which a miner includes a newly mined block. This transaction credits the miner with the value of the mining reward. Once his block is accepted by the network that reward value can be used. As per the protocol, from that point onwards the block mining reward transaction acts as the root reference of the value created by that transaction [15]. It is created from thin air, there is no previous reference to it. Even though it does not have any previous reference, the system and its users will fully accept the value derived from it.

The ability to transfer value from one user to another in a decentralized manner on a blockchain is made possible through a verification process of past transactions carried out by nodes.

2.3 Blockchain system key components

The blockchain is made up of a chain of blocks, with each block containing a unique cryptographic hash of the previous block, a timestamp, and transaction

[4] a series of specialized math problems or state a token

information. This information is recorded and added to the blockchain as new blocks, resulting in an ever-growing chain of data.

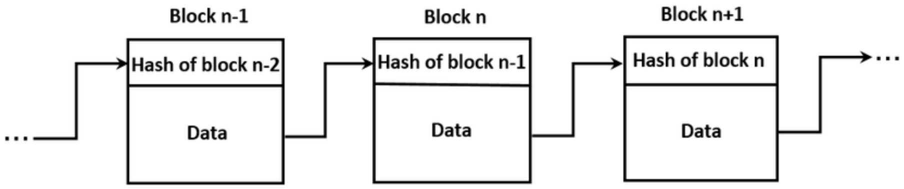

Fig. 3. Blockchain data structure

Blocks In blockchain technology, the database is segmented and compiles records of transaction data into systems called *blocks*. Each block is cryptographically connected with the other; thus, it forms a chain of blocks that hold the entire transaction history. Blocks are identified by the cryptographical hash of their contents called *block headers*, which are made up of their metadata[5], a reference of the previous block, and a root hash of all the transactions contained in that block (refer Figure 4). A block header is a critical piece of information of any block, which represents every single bit of information about that block.

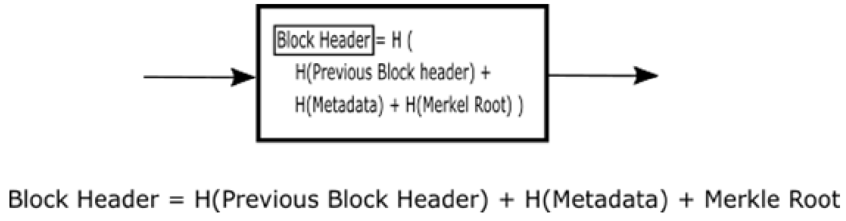
Block Header = H (
H(Previous Block header) +
H(Metadata) + H(Merkel Root))
Block Header = H(Previous Block Header) + H(Metadata) + Merkle Root

Fig. 4. Basic internal components of a block

Blockchain is a system of recording transactions in a decentralized manner. Transactions are grouped into blocks with a fixed size, and each block has a cryptographic hash of the previous block, a timestamp, and transaction data. These blocks are linked and replicated on all network participants' systems, creating a distributed ledger that maintains an identical copy of the transaction history. The ledger is an append-only data structure that records all updates

[5] information about the block

made to the initial states and preserves the exact order in which they were stored.

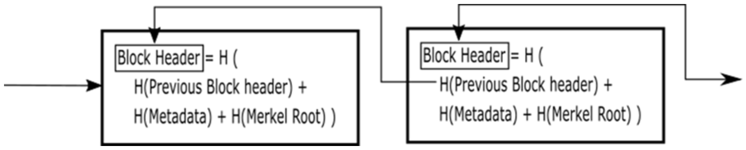
Block Header = H (
H(Previous Block header) +
H(Metadata) + H(Merkel Root))
Block Header = H (
H(Previous Block header) +
H(Metadata) + H(Merkel Root))

Fig. 5. Block connecting structure

Hash functions Hash functions are used in blockchain to check the integrity of the contents in the block. The primary identifier of a block is the hash value of its block header. As shown in Table 1, the block header consists of three main components: a) the metadata of the block which includes the difficulty level, timestamp and nonce; b) the hash of the previews block header; c) a transaction ID (Merkle root hash of the transactions). The exception of this will be the first block on that chain which is called the genesis block (the beginning block).

Table 1. Block header

Metadata (difficulty level, timestamp and nonce)
Hash of the previous block
Transaction ID (Merkle root of all the transactions included in that block)

Each block has a reference to the previous block identity. This forms a chain of blocks and establishes the link between blocks which is cryptography secured by their hash values. A hash function is a mathematical function that generates a unique fixed fingerprint for an arbitrary size of data [10].

The hash of a block can be used to verify the integrity of the block's data. If the data within the block changes, the hash of the block will change, but if the data remains unchanged, the hash will stay the same.

A hash function is a mathematical tool that takes in a numerical input of data and outputs a digital fingerprint. The digital fingerprint represents the input data and is used to confirm the data's accuracy. The ideal properties of a hash function include:

- Collision resistance: - it is not computationally feasible to find two inputs that have the same hash output where x and y are two different sets of data and $x \neq y$ and $H(x) = H(y)$.
- One-way function: - For any given hash value h, it is computationally infeasible to find y such that $H(y) = h$. Therefore, it is practically infeasible of generating the input message given a hash value.

– pre-image resistance - meaning that given a value h, it is not feasible except with negligible probability to find a value x such that $H(x) = h$.

The hash function is used in a blockchain to create a digital fingerprint of each block of data. The Merkle tree is a data structure used in computer science, particularly in the blockchain field. It is used in Bitcoin to store transaction information in a format that allows for efficient and secure verification of the entire transaction history.

The Merkle tree structure The Merkle tree hash technique is a way of hashing a number of pieces of data together like a tree structure (refer Figure 6). To build the transaction tree, two transactions are concatenated and hashed. The result is again concatenated and hashed until there is only one hash left. Merkle trees are a fundamental component of blockchain technology. Each block contains a Merkle root summary of all the transactions included in that block.

The Merkle tree structure provides an efficient method for verifying large sets of data. When a data block is added to the blockchain, its hash is calculated and incorporated into the Merkle tree. The hash of the entire tree is then updated and added to the blockchain. To check the authenticity of a data block, a user only needs to obtain the block and its corresponding Merkle tree, then compare the calculated hash of the data block with the hash stored in the Merkle tree. If the two hashes match, the user can trust that the data block has not been altered.

The Merkle root hash for a transaction is created using a Merkle tree hash function that encompasses the hash of all transactions in the block. Any modification to a single bit in these transactions will result in a different hash value. Hence, the transaction ID serves as cryptographic evidence of each transaction in a block, which is then included in creating the block header. Therefore, after a block is approved and accepted, changing just one bit will result in a different hash value.

The network The blockchain operates within a decentralized network, managed by a network of nodes that are responsible for the creation, verification, and validation of transactions. The network incentivizes nodes to self-govern the system by adhering to the protocol. As a technology of distributed databases, each node in the network holds a copy of the database and any updates must be shared across the network. As a result, blockchain is a decentralized data structure that maintains a continuously growing list of records, shared among all participants in the distributed network [22].

Consensus The consensus mechanism is a critical aspect of the blockchain technology that ensures the nodes in the network agree on the global order and state of transactions. This is a security measure to avoid double-spending and secure the system. Unlike traditional databases, blockchain transactions are never altered or deleted. Instead, they are recorded and kept as a permanent record, with

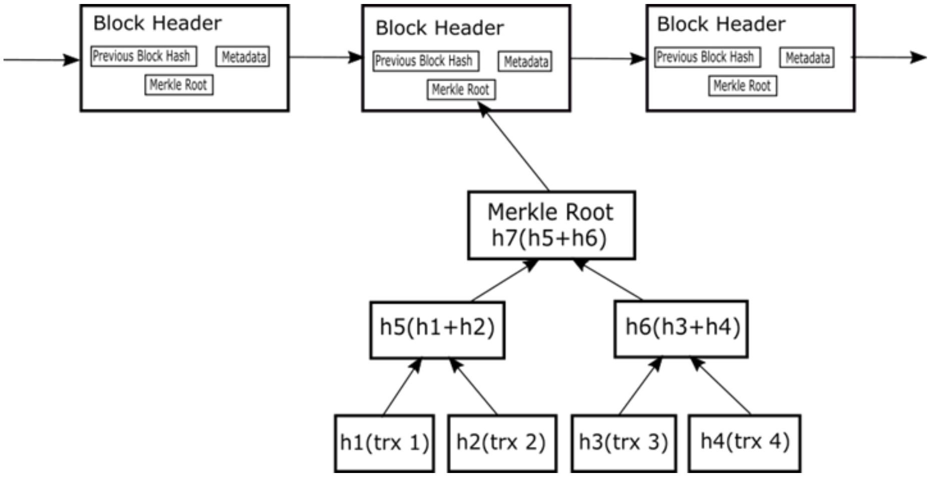

Fig. 6. Blockchain with consists of a continuous sequence of blocks

new transactions added to the next block. Transactions are validated, grouped into blocks, and added to the chain in the order they are received. Each block is time-stamped and has a unique cryptographic hash that is linked to the previous block, forming a chain of blocks. This makes it practically impossible to modify the data once it is recorded on the blockchain. Different consensus protocols exist, with Bitcoin using a public, permissionless Proof-of-Work protocol. Private blockchains, on the other hand, operate on a permissioned network where the identity of nodes is known and trusted, so they do not have to perform computational work to validate transactions.

Mining Mining is a competition to solve a puzzle, where the winner is the miner who can broadcast the new block to the whole network first. The sender's node does not need to rely on the other nodes on the network, as long as it uses multiple nodes to ensure the transaction is spread. Conversely, the nodes on the network do not need to trust the sender, as transactions are signed and can be verified by anyone. Mining nodes perform the process according to the consensus protocol, with Bitcoin using a Proof of Work (PoW) mechanism. This involves finding a number that, when hashed with the block's data, produces a result starting with a certain number of zeros. This requires investing in computing power to solve a difficult mathematical problem. The level of difficulty, which increases with the number of zeros required, is determined by the system configuration as part of the consensus mechanism.

Mining difficulty Making mining computationally expensive, each node participating in the mining process has to invest in computation power to find the right random number (called a nonce) as part of the process to form the next block. This process involves finding the block header, which will be a root hash value of all the components in that block with a random number. The network requires the block header to start with the number of leading zeroes that the network defined. This process is called proof of work. The only way to find this random number is to perform brute force to try the node that solves this problem to get to all the blocks of the existing chain. It is computationally hard to perform this operation but easy to verify and any other node can easily verify the result.

There are several types of consensus mechanisms, such as proof of work (PoW), proof of stake (PoS), delegated proof of stake (DPOS), proof of activity (PoA), proof of authority (PoA), proof of burn (PoB), proof of space (PoSpace), proof of elapsed time (PoET), proof of history (PoH), proof of importance (PoI), that can be used to validate transactions in a blockchain network [26]. Each consensus technique has a unique set of benefits and drawbacks. While new algorithms are constantly emerging, PoW and PoS are the most commonly used algorithms.

PoW In the PoW network, to create a new block, a random miner must be chosen for the job. Network miners compete against each other using computational power to solve highly complex mathematical problems for the right to create a new block in the network; hence it is named *proof of work* [31]. Additionally, the victorious miner receives a *block reward*, a fixed sum of digital money more commonly known as cryptocurrencies. Although it has been the first ever designed consensus method, it is still used today (2022) by currencies such as Bitcoin, Dogecoin, Monero, and Litecoin. Nevertheless, this method has several scaling difficulties with high running expenses since producing new blocks demands a significant amount of computer power and energy [24].

PoS Unlike PoW, PoS is n eco-friendly alternative that utilises staking procedures. Users must ”stake” their cryptocurrencies to be eligible to be selected as a miner on the network [31]. The more coins are staked more chances the user has to be chosen for each round. Even though this consensus mechanism is a more sustainable alternative to PoW, still the system is more favourable for those who stake the highest number of tokens, hence more likely to centralised validate power which goes against the blockchain principles [26].

Transactions in the Bitcoin blockchain are secure because the network employs sophisticated security algorithms, and the transaction ledger is distributed across a network of unrelated computers. To compromise the security of the Bitcoin blockchain, a hacker would need more computing power than half the nodes in the Bitcoin blockchain. Due to the size of the network participants in Bitcoin blockchain, this is very much impossible for attacks to take over the network because of resource required for it.

3 Technical limitation of blockchain technology

The two major challenges of blockchain technology are scalability and integration with other systems. Despite being a decentralized system, blockchain is not able to handle the same level of transactions as centralized systems due to the need for every node in the network to process and verify each transaction. This can slow down the system and consume a lot of energy. Additionally, blockchain has difficulty integrating with other systems and technologies, making it challenging to integrate it into existing systems.

3.1 Scalability

In the context of blockchain, scalability is defined as the ability of a blockchain system to sustain performance while growing and expanding[13] and that is the number one issue the blockchain system is facing now. In many ways, scalability is identified as a major technical limitation for the wide adaptation of this technology [30]. There have been several proposals to solve this problem. However, in practice, different blockchains have limitations on performance and scalability. This has been suggested as the most important limitation for a public blockchain, and many efforts have been undertaken to address the scalability issues. Moreover, in a blockchain system, scalability is not determined by a single property.

The design of permissionless blockchain systems can only process a fraction of transactions in comparison to centralised services like VISA and PayPal [2]. This will not change no matter what the network power or mining power is. A mining node can only include a relative number of transactions per block due to its design restriction. Therefore, with the current architecture (until 2022) based on *Bitcoin* or *Ethereum*, a blockchain system with a given protocol can only perform a certain number of transactions.

The design of the *Number of transactions per second* is an important part of the blockchain system [23]. Generally, there are two parameters that determine the number of transactions per second. Primarily, the *block size* determines the number of transactions that can be included in a block, and the *latency*, sets the interval for the block creation time as part of the security mechanism on the blockchain design.

Block size A simple solution to the scalability challenge in the blockchain is to increase the block size so that each block can store more data. The main benefit of increasing the block size limit is that it allows the Bitcoin network to process more transactions per second. This is important because the more transactions that can be processed per second, the more useful Bitcoin becomes as a payment network. However, there are some trade-offs to increasing the block size limit. For example, larger blocks mean that each individual node in the network needs to process more data. This could lead to fewer nodes being able to participate in the network, which could centralize power and make the network less secure.

As of 2022, the protocol that runs by Bitcoin has a fixed block size limit, this will limit the number of transactions that can be included in that block. The block size limit is set in regard to network propagation time. Whereas in Ethereum, the block size limit is based on the block gas limit, each transaction costs a certain amount of *gas*, and each operation is assigned a fixed amount of gas. In effect, the gas limit of a block is determined by how many transactions will fit in a block based on the gas limit specified by each transaction in the block.

Latency Blockchain protocol has a latency or network propagation time parameter added to its protocol. Let us examine what is it and how to define the numbers. Latency, the set interval for block creation time is the time for the network to agree on standardising the block, which belongs to the security mechanism of the blockchain design. The bitcoin blockchain is designed to take 10 minutes to confirm a transaction. Each new block announcement that is broadcasted to the network has to propagate to the entire network [2]. This is implemented as a solution for the double-spending problem, where one can spend the money twice before the first request reaches all the nodes. So, imposing a delay will make sure that all nodes have been updated on the status before moving on to the next transaction.

In a blockchain system, the nodes are spread across the network around the world. To perform effectively, all nodes should update the information as it is available. However, there may be network delay and latency in the network itself. As a result, there is a probability of two nodes forming a block around the same time. In fact, both are technically valid blocks and may accept as the next block by the network. However, eventually, the network that comes up with the next block will be accepted as the longest chain by the network. This will result in rejecting the other block created even though the block contains all valid transactions. Discarding a block will not only be a waste of effort but will also create confusion and conflicts among the networks resulting in forking, at least for a couple of blocks. This can be avoided by imposing a block propagation time [22].

Definition 2 (Latency) *Network latency is defined as the time required for network communication. Bitcoin protocol imposes a delay so that the network takes an average time of 10 minutes for a block to confirm.*

In order to understand how the delay is created and imposed, it's important to consider two factors. One factor is the network propagation time and the other is the core of the mining process. The mining process involves miners solving a computational problem to form a valid block, which also acts as a security system to prove the authenticity of miners by putting down their computational power. However, this is only necessary in a permissionless network, where the nodes' identities are unknown. Meanwhile, in a permissioned network, where nodes' identities are known and trusted, this step can be omitted. Despite this, the network propagation time delay is still necessary to ensure that the information spreads across the network with multiple miners validating. To maintain its core properties, a blockchain system should limit its scalability. Currently, blockchain has limited scalability. In a distributed system, there is no ideal consensus protocol and a trade-off must be made between consistency, availability, and partition fault tolerance (CAP) [12]. This is similar to the CAP theorem, which states that a distributed computer system can only provide two out of the three guarantees of consistency, availability, and partition tolerance at once.

CAP theorem. In a distributed system, network failures can often result in partitions, leading to a trade-off between consistency and availability. If consistency is prioritized, the system will return an error if the information is not current, but if availability is prioritized, the system will still process the request and attempt to provide the most recent version of the information available, even if it may not be completely up-to-date due to network partitioning. If there is no partitioning, both consistency and availability can be ensured.

3.2 Scalability trilemma

The trilemma in the context of blockchain states that the system can only balance the properties of decentralisation, security and scalability. This is a design constraint that blockchains may not be able to solve without a trade-off.

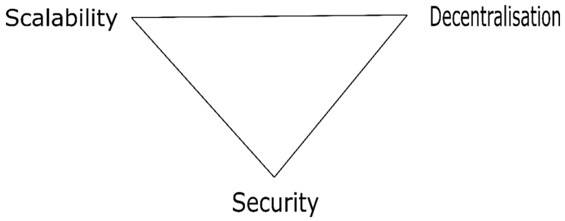

Fig. 7. Blockchain scalability trilemma

- Scalability refers to the number of transaction processing capabilities.
- Decentralisation, means that every single node on the network processes every transaction and maintains a copy of the entire state.
- Security is achieved by the decentralisation consensus mechanism.

In the case of blockchain, this trilemma in Figure 7 states that decentralised blockchain networks can only provide two of three benefits at any given time with respect to decentralization, security, and scalability. Consistency, every node holds a current valid set of blocks. Availability with the distributed nature every node has a copy of data. Partition tolerance, when a network partition occurs, the system continues to operate until the partition is resolved, instead of becoming unavailable or failing entirely.

Technically, a distributed system consists of a network of independent nodes that are connected together [5]. These nodes can only communicate through messages i.e. there is no physical or shared memory between the nodes. As illustrated in Figure 8 a valuable property to maintain is the balance between the *event computation time* and the *message transmission time*. The *event computation time* $t(n1)$ and $t(n2)$ is the time it takes for a node to perform meaningful processes and the *message transmission time* $t(b - c)$ is the time takes to communicate messages between the nodes on the network. Leslie Lamport[6] defines a distributed system as distributed if the message transmutation time is not negligible to the time between events in a single process. CAP theorem has proved that a distributed system cannot achieve consistency, availability or partition tolerance simultaneously [12]. Generally distributed systems present the problem of fault tolerance which leads to choosing the balance between consistency over availability.

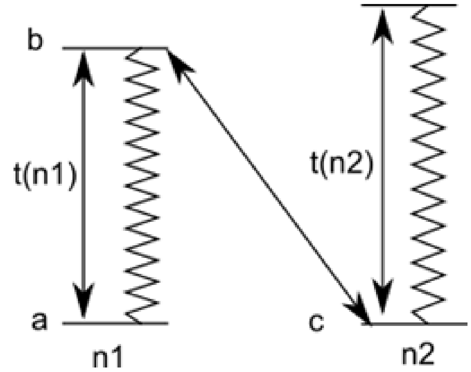

Fig. 8. Distributed system message between nodes

The design of a system depends on certain parameters, which determine its performance based on the number of results it produces. In a decentralized blockchain system, every node is responsible for verifying and processing every transaction, as well as keeping a copy of the entire system state. This design enhances the system's fault tolerance, but it also limits its scalability. For this reason, the scalability of blockchain systems must be balanced with the need to maintain their core security features. To gain a deeper understanding of the performance and limitations of blockchain technology, it is important to examine factors such as throughput and latency in relation to scalability.

In the case of blockchain, this trilemma in Figure 7 states that a decentralised blockchain network can only provide two of the three benefits at any given time

[6] http://www.lamport.org

with respect to decentralisation, security, and scalability. An improvement in scalability to increase performance will impact either decentralisation or security.

From another angle, to address scalability and improve efficiency, a decentralised network can adopt parallel computing. In a typical blockchain system, every node has to store an up-to-date state of every transaction and perform the same operation in order to participate in the network. More nodes verifying the transactions will be good for the system's security but, limit the scalability to a point where a blockchain system cannot process more transactions than a single node. Logically a network with thousands of nodes will have higher throughput than a single node, but a blockchain network is not capable of it. In permissioned blockchain, the design permits configuring a network of nodes to scale independently of each other. The nodes need to have permission to operate, thus it removes the computation task, or even parallel computing can be possible whereas in a public blockchain with proof-of-work like a consensus, a scalability limit is built into it. A protocol with a shorter block time latency needs more confirmation for the same level of security as a protocol with a longer block time [9].

3.3 Classification of scalability approaches

Blockchain scalability approaches can be classified into three main categories:

- On-chain solutions: This refers to increasing the capacity of a blockchain network by making changes to its underlying protocol.
- Off-chain solutions: This refers to moving transactions and computations off the blockchain and into secondary networks or layer 2 solutions, while maintaining the main blockchain's security and trust.
- Interoperable blockchain networks: A solution to improve the scalability of blockchain networks by making them interoperable, allowing transactions to occur across multiple networks and reducing the strain on any one network.

Each approach to scalability has its own trade-offs in terms of security, decentralisation, and speed, so it is important to carefully evaluate the specific needs and requirements of a given use case before choosing the most appropriate scalability solution. Currently, researchers are still exploring the technology and identifying the benefits of this approach.

On-chain On-chain approaches aim to enhance blockchain scalability by adjusting the internal parameters of a blockchain. This can be achieved by improving the network latency, specifically, by optimizing the transaction or block size.

Off-chain A recent development with blockchain scalability is the *Roll-up* solutions. Rollup solutions refer to the execution of transactions outside the main network, so-called layer 2 networks, while anchoring the transaction data proof on the main chain referred to as layer 1. This will allow the transaction to have the security features of layer 1 even though the executing operations outside of the network on layer 2. There are a number of roll-up solutions proposed to help scale a blockchain application by handling transactions off layer 1 while taking advantage of the robust decentralised security model of layer 1. Rollups are deployed using a set of smart contracts on Layer 1 that are responsible for processing deposits, withdrawals, and verifying proofs. When it comes to verifying proofs, this brings the main distinction between different types of rollups such as *Optimistic rollups* use fraud proofs and *ZK rollups* use validity proofs.

Interoperability In information systems, interoperability is generally understood as the ability of two or more systems to communicate and exchange information [25]. The process related to exchange is the use of the exchanged data to carry out an operation, referred to as interoperation [29]. Interoperability can also be characterised as a relationship of the exchange and cooperative use of data. Interoperability occurs when two systems successfully use the exchanged data despite differences in language, interface, and execution platform [27]. Recently, interoperability has gained different definitions within the context of blockchain. Increasingly common usage refers to the *interaction* and *exchange* of data between networks of blockchains. This opens up various possibilities for cross-blockchain transactions, for example, value transfer in the form of asset or payment versus payment and payment versus delivery schemes or information exchange [4].

Generally, integration involves middleware mechanisms to transport the data and make it accessible to the other network. In the case of blockchain-based systems, the most significant obstacles to interoperability are the consensus of each chain and how data moves from one chain to another. Cross-blockchain technology has become a topic of discussion to address the integration issues of blockchain networks[21].

4 Conclusion

Irrespective of differing views about cryptocurrency, business applications triggered significant interest in blockchain technology. In future, blockchain technology will transform into various business applications to provide its added benefits of security, transparency and accountability. The ability to create new business models makes it an exciting and promising technology worth exploring. However, as the business application evolves to develop new and innovative applications through the development of different variations and implementations of blockchain architecture. Modifications to the core design compromise the principles and fundamentally alter the nature of the blockchain and potentially undermine its stability and trustworthiness. Therefore, it is essential to approach any modifications cautiously and consider the potential consequences thoroughly. This chapter provides a brief overview of the fundamentals of DLT, Blockchain and its design limitations.

5 Acknowledgements

The authors would like to acknowledge the valuable comments and suggestions by Prof. Vallipuram Muthukkumarasamy and Dr. Kamanashis Biswas on the content given in this chapter.

Bibliography

[1] Alysson Bessani, João Sousa, and Eduardo EP Alchieri. State machine replication for the masses with bft-smart. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 355–362. IEEE, 2014.

[2] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE symposium on security and privacy*, pages 104–121. IEEE, 2015.

[3] Daniel Burkhardt, Maximilian Werling, and Heiner Lasi. Distributed ledger. In *2018 IEEE international conference on engineering, technology and innovation (ICE/ITMC)*, pages 1–9. IEEE, 2018.

[4] Vitalik Buterin. Chain interoperability. *R3 Research Paper*, 2016.

[5] Christian Cachin, Rachid Guerraoui, and Luís Rodrigues. *Introduction to reliable and secure distributed programming.* Springer Science & Business Media, 2011.

[6] Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167, 2016.

[7] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186, 1999.

[8] Usman W Chohan. The double spending problem and cryptocurrencies. *Available at SSRN 3090174*, 2017.

[9] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18. Springer, 2015.

[10] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1085–1100, 2017.

[11] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[12] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, 33(2):51–59, 2002.

[13] Garrick Hileman and Michel Rauchs. 2017 global blockchain benchmarking study. *Available at SSRN 3040224*, 2017.

[14] Jaap-Henk Hoepman. Distributed double spending prevention. In *International Workshop on Security Protocols*, pages 152–165. Springer, 2007.

[15] Nicolas Houy. The bitcoin mining game. *Available at SSRN 2407834*, 2014.

[16] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. Zyzzyva: speculative byzantine fault tolerance. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 45–58, 2007.

[17] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, 2020.

[18] Satoshi Nakamoto. A peer-to-peer electronic cash system. Available at: `https://bitcoin.org/bitcoin.pdf` (Date last accessed 28-Feburary-2020), 2008.

[19] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[20] Harish Natarajan, Solvej Krause, and Helen Gradstein. Distributed ledger technology and blockchain. 2017.

[21] Babu Pillai, Kamanashis Biswas, Zhé Hóu, and Vallipuram Muthukkumarasamy. The burn-to-claim cross-blockchain asset transfer protocol. In *2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 119–124. IEEE, 2020.

[22] Yonatan Sompolinsky and Aviv Zohar. Accelerating bitcoin's transaction processing. fast money grows on trees, not chains. *Cryptology ePrint Archive*, 2013.

[23] Yonatan Sompolinsky and Aviv Zohar. Bitcoin's underlying incentives. *Communications of the ACM*, 61(3):46–53, 2018.

[24] Florian Tschorsch and Björn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123, 2016.

[25] FB Vernadat. Interoperable enterprise systems: architectures and methods. *IFAC Proceedings Volumes*, 39(3):13–20, 2006.

[26] Yuhao Wang, Shaobin Cai, Changlong Lin, Zuxi Chen, Tian Wang, Zhenguo Gao, and Changli Zhou. Study of blockchains's consensus mechanism based on credit. *IEEE Access*, 7:10224–10231, 2019.

[27] Peter Wegner. Interoperability. *ACM Computing Surveys (CSUR)*, 28(1):285–287, 1996.

[28] Kevin Werbach. Summary: Blockchain, the rise of trustless trust? 2019.

[29] Lawrence E Whitman, Danny Santanu, and Hervé Panetto. An enterprise model of interoperability. *IFAC Proceedings Volumes*, 39(3):609–614, 2006.

[30] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, and Kari Smolander. Where is current research on blockchain technology?—a systematic review. *PloS one*, 11(10):e0163477, 2016.

[31] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. IEEE, 2017.