LONDONCOIN: THE KING OF CRYPTOCURRENCIES

LONDONCOIN SYSTEMS, LTD.
Email: info@londoncoin.biz
(ver.0.1, created on October 31st, 2021)

Abstract

Although many ideas and attempts have been made trying to increase transaction processing speed ever since a blockchain has been introduced, a speed racing war is still going on even now, since real-time payment and transactions in a real world are necessarily required for supporting actual trading in a market. Amongst them, particularly Hedera Hashgraph adopted a directed acyclic graph (DAG) that does not use a general linked block structure and it has obtained much attention by announcing its high transaction processing performance. Nevertheless, it is protected by patents and designed for a private blockchain system instead of a public one, which goes against the spirit of a blockchain technology.

Therefore, we propose an ultimate blockchain engine called a Babble Core which can be freely implemented by many open-source developers and used by blockchain communities since it is genuinely designed and targeted for a public blockchain that is even capable of working in a mobile network environment. Our insights indicate that the major properties of a global snapshot, smart gossip, leader election with extinction, a sidechain called an inter-block chain, etc. can be enforced to significantly match or outperform the current blockchain engines and to be placed in a pivotal position for achieving a feasible crypto economy.

Key words: cryptocurrency, blockchain, directed acyclic graph (DAG), consensus, trust, proof of trust, asynchronous byzantine failure tolerance (aBFT), scalability, mobility, stability, self-organizing network (SON), inter-block chain, Babble Core.

1 Introduction

Bitcoin as a forerunner cryptocurrency has been introduced into the market around early 2009. Since then, it created a new industry based on a blockchain technology and caused the production of more than 1,600 coins such as Ethereum, Litecoin, Ripple, iota, etc., which are

known as Bitcoin alternatives or alternate coins (altcoins), and it developed a market capacity worth 327 billion dollars[1] exponentially. What does it bring to the market and how does it make an impact on our society and daily lives? The fundamental concept of a blockchain may have more potential than by solely trading a cryptocurrency; so that it may change from the current Internet of information to the Internet of value (or money), or from a worldwide web to a worldwide ledger by eliminating a middle man such as a government, a bank, a big corporation, or even high-tech based traditional big companies by guaranteeing privacy, safety, transparency, and integrity rather than all of the traditional services provided by the middle man [1, 2].

Since a decade of their existence, cryptocurrencies are used by some early adopters and they are traded in a market. Even so, people can transfer the cryptocurrency to someone who lives in a foreign country and even to those who do not even own a bank account; thereby paying lower transfer fees that are not comparable to the fees of the middle man such as traditional banks, Western Union, PayPal, and more. Moreover, people can also buy commodities via a cryptocurrency or exchange a cryptocurrency to a fiat currency like the USD or Euro dollars and vice versa via the Internet or ATM equipment in certain regions. Numerous applications based on the blockchain are not limited to the cryptocurrency issuance, but they are expanding to many areas integrating with smart contracts, namely in the sector of the arts, gaming industry, music, intellectual properties, land registration certificates, agricultural products, etc.

We wish to further investigate on whether obstacles still exist or not when using cryptocurrency in the real world. At present, merchants are not willing to receive the cryptocurrency from their customers since its nominal price can fluctuate unexpectedly and it takes a considerable amount of time to receive confirmation for the transactions. In the case of Bitcoin, every 10 minutes, a block is created and confirmed, and with additional time, it takes theoretically about an hour to resolve forks [3]. If many transactions occur at the same time, then it is known that it takes more than one hour. Who will be daring enough to take Bitcoin? Current blockchain technology is still at an early stage in terms of technology, and there is currently an arms race regarding the technology. If processing throughput which is measured in transactions per second (TPS) is not resolved soon, then the future of cryptocurrency may not be promising.

[1] https://coinmarketcap.com/

As there is a contingent issue over the fall or the rise of the crypto economy, which is closely related to the transaction processing speed, therefore, we wish to propose an ultimate blockchain engine called Babble Core as an alternative to clearly resolve the issue.

2 Blockchain Platform

2.1 Blockchain

A blockchain is a purely distributed peer-to-peer system of ledgers that utilizes a software unit that consists of an algorithm, which negotiates the informational content of ordered and connected blocks of data together with cryptographic and security technologies to achieve and maintain integrity [4]. Therefore, the blockchain as a platform contains part of a database, part of a development platform, and part of a network enabler. Thus, we need many instances of it and variations thereof. As an overlay on top of the Internet, blockchains can take many forms of implementations, from which it can be working as a trust layer, an exchange medium, a secure pipe, a set of decentralized capabilities, and even more [2].

As enumerated a traditional approach based on a central server with a blockchain one, which is indicated in Table 2.1, the blockchain provides distinguished features such as fairness, resilience to denial of service (DoS), immutable records, distributed trust, etc., where the traditional approach cannot compete with it. On the other hand, transaction processing performance and finality time are still lagging far behind credit card and PayPal services because it implies that a blockchain technology is still at an infant stage.

Unlike the traditional approach, another crucial metric called finality is introduced where the finality is the guarantee that past or recent transactions will not change. In blockchain systems today, transactions are considered immutable. But, most blockchain systems only offer probabilistic transaction finality - that transactions are not immediately final but become so eventually. In the case of Bitcoin and Ethereum, consequently, it is recommended that one wait for up to 6 confirmations when transacting on the Bitcoin blockchain and 20–25 confirmations on the Ethereum blockchain to prevent a double-spend attack. In addition, finality measures are all about how long one must wait to be given a reasonable guarantee. Hence, the transaction written in the blockchain is irreversible, or in other words, it will not be orphaned.

Therefore, when it is transacting large amounts, it is better to use higher confirmation numbers, because the higher the number of confirmations, the stronger the finality of the transaction. Think about a merchant making a customer wait for an hour at a store! This is a

critical issue for a business and it can provide significant repercussions for businesses.

Table 2.1 Comparisons between the Traditional approach and Bitcoin

Features	Traditional approach (Central Server)	Bitcoin Blockchain
Fairness	No	Yes
Low computation	Yes	No
Resilience to DoS	No	Yes
No single point of failure	No	Yes
Network configuration	Static	Dynamic
Scalable	Yes	No
Immutable record for audits	No	Yes
Distributed trust	No	Yes (Power of Work)
Reliable storage and high availability	No	Yes
Preventing failures	No	Byzantine behavior
Performance (tps)	Visa: 2,000 -Daily peak: 4,000 -Peak capacity: 56,000 -Peak shopping period: 18,700 PayPal: 193	7
Finality time	7 seconds	60 mins

Since the emergence of Bitcoin, to speed up transaction processing performance and to reduce the finality time, many blockchain platforms have been devising and they have been announced. Particularly, some definitions are public or private, and permissionless or permissioned, which cause confusions for the general audience. At present, we want to categorize blockchains with some criteria such as public or private, permissionless or permissioned, or a block or a directed acyclic graph called DAG.

A. Classifications

- Public (or Permissionless) vs Private (or Permissioned)

In general, public or permissionless is interchangeably used to have the same meaning, and private or permissioned *vice versa*. In a public blockchain, any computer can freely read, write, or join the blockchain. Public chains are decentralized, meaning that no one

has an exclusive control over the network, ensuring that the data cannot be changed once validated on the blockchain. Simply meaning, anyone, no matter where he/she is located, can use a public blockchain to input his/her transactions and data if he/she is connected to the network. Public blockchain networks are expensive to run and have performance constraints resulting from fierce competitions among nodes and this narrows down the number of applications where such technologies can be practically employed. Some well-known examples of public blockchains would be Bitcoin and Ethereum, with Bitcoin being among the first Blockchain application to prove that value could be moved across the globe without third-parties like banks or remittance companies. They are open for anyone to participate at any level and provide open-source code to the public, which are maintained by their communities.

Blockchains that are private or permissioned work similarly to public blockchains but with access controls that restrict those that can join the network, meaning that it operates like a centralized database system that limits access to certain users. Private blockchains have one or multiple entities that control the network, leading to the reliance on third-parties to transact. By acting as closed ecosystems, where users are not freely able to join the network, to see the recorded history, or to issue transactions of their own, the permissioned blockchains are preferred by centralized organizations, which leverage the power of the network for their own internal business operations. Company consortiums are also likely to employ private blockchains to securely record transactions, and exchange information between one another. Private networks, unlike public networks, restrict usage to known and trusted participants. This approach brings down the cost and improves performance dramatically, while loopholes, due negligence or moral hazards in the form of relaxed security standards make these networks potential targets to DDoS attacks like the case of centralized operations. Some well-known examples would be Hyperledger and Ripple, and their code may or may not be given as open source.

On the other hand, Drescher [4] suggests some distinguished definitions unlike other ones as follows: 1) a public blockchain grants read access and the right to create new transactions to all users or nodes while a private blockchain limits read access and the right to create new transactions to a preselected group of users or nodes. 2) a permissionless blockchain grants write access to everyone. Every user of node can verify transactions and create and add new blocks to the blockchain's data structure while a permissioned blockchain grants write access only to a limited group of preselected nodes or users that

are identified as trustworthiness through an on-boarding process. As a result, only the group of nodes that have write access can verify transaction and take part in the distributed consensus procedure. It can be categorized into four classifications as shown in Table 2.2.

Table 2.2 Four Versions of Blockchains

	Reading access and Creation of Transactions
Every node	Public
Restricted node	Private

	Writing access
Every node	Permissionless
Restricted node	Permissioned

Although Drescher's definitions are useful to distinguish the public (or private) from permissionless (or permissioned) in details, it may give more confusion about some of the blockchain platforms that we are familiar with. To maintain consistency throughout this whitepaper, we will adopt the previously used definitions, not Drescher's ones.

- Block vs DAG

The data structure of a blockchain is not a straight line of blocks; instead, it is a tree-shaped data structure whose branches represent conflicting versions of the history of transaction data. The major challenge of the blockchain-algorithm is to let the nodes of the distributed system select consistently one of the branches as the authoritative chain. An alternative approach of storing transaction data is to utilize a directed acyclic graph of blocks instead of a tree-shaped data structure. One can imagine a DAG of blocks as a tree-shaped blockchain-data-structure whose branches merge later. The usage of a directed acyclic graph for storing the transaction history has far-reaching consequences on performance, clarifying ownership, and reaching consensus among peers.

2.2 Platform Model

We define a reference model of a blockchain platform (or a blockchain engine), consisting of four layers, as a motive power to drive a crypto-economy which is illustrated by Figure 2.1. From the bottom side to the top one, a peer to peer network layer covers an overlay network over peer-to-peer communications and even extends to a self-organized network over peer-to-

peer communications where mobile devices can freely form in a network and perform trading and transactions on top of a blockchain without the existence of an infrastructure such as servers and the Internet.

In particular, a trust (or consensus) layer known as a blockchain engine plays a pivotal role in managing all of the transactions and confirming them even under a Byzantine failure or a node crash, and a transaction layer provides the role of trading and arbitrage, arbitration, a mediation between services and distributed ledgers, and a service layer that provides various markets, users, applications, and services such as using a contract, cryptocurrencies, and file storage, etc. We consider this situation different compared to other blockchain platforms because traffic from smartphones will exceed PC traffic by 2021 and traffic from wireless and mobile devices will account for more than 63 percent of total IP traffic by 2021 [6].

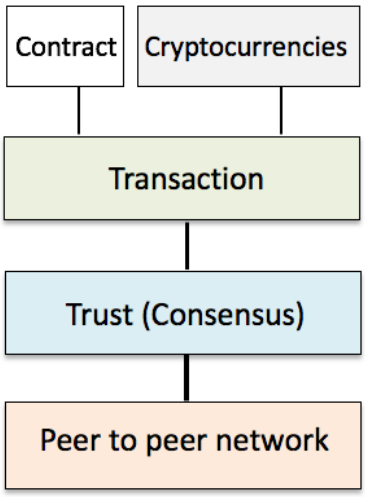


Figure 2.1 A Blockchain Platform

Even in the worst situation, where servers and the Internet do not exist, we are targeting for each wireless mobile device such as a smartphone to connect to each other and form a wireless mobile ad-hoc network or a wireless mesh network, form a blockchain and perform payment or trading via a blockchain.

2.3 Exemplary Platforms

Since Bitcoin has been introduced into a market, some exemplary cryptocurrencies and platforms are enumerated, which are shown in Table 2.3. Initially, from a public based platform such as Bitcoin, Ethereum to recent ones such as Hashgraph, Radix, etc. are covered up in terms of type, consensus protocol, chain structure, throughput and finality time.

Most of the public blockchain use probabilistic or stake-based protocols to select a certain node to propose a new block. This scales well, whereas minimal information is exchanged over a network to achieve consensus but leads to chain forks in the network and therefore longer time to finality. As you can see, private and DAG-based blockchains are shown to be faster and shorter than public and block-based ones in terms of throughput and finality time. Hashgraph seems to be a powerful platform except that it is designed to run with previously determined and fixed number of nodes, which is known to be a set of normal nodes guaranteed and controlled by a private governance body with 39 members. In addition, it is protected from patents and the code is tightly managed to avoid forking activities, unlike most of the blockchain platforms. With different blockchain platforms, the comparisons are not reasonable because each approach has its own purpose and there are assumptions to be used and to be designed.

Table 2.3 Exemplary cryptocurrencies and platforms

	Type	Consensus	Chain structure	Throughput (tps)/ finality time (sec.)
Bitcoin	Public	PoW	Block	7/3,600
Litecoin	Public	PoW	Block	56/1,800
Ethereum	Public	PoW	Block	15/360
	Private	PoS		100/7
Ripple	Private	Practical BFT	Block	1,500/3-4
Tendermint	Private	Practical BFT	Block	10,000/1
Algorand	Public	Pure PoS	Block	875/?
iota	Public	tangle	DAG	1,500/120
EOS	Public	Delegated PoS	Block	3,097/2
Radix	Public	Asynchronous BFT	DAG	24,000/?
Hashgraph	Private	Asynchronous BFT	DAG	150,000/3

Currently, there is an arms race, which is reminiscent of the Cold War era, for developing a powerful, robust, and secure blockchain platform.

To win the race, a significant amount of scalability must be obtained. The scalability is defined as the capability of a system, network, or process to handle growing amounts of work, or its potential to be enlarged, in order to accommodate that growth [7]. For example, suppose that a cryptocurrency or assets are designed to cover up worldwide economic transactions. To provide global financial services, we need to know our competitor's transaction performance.

In the case of Visa, it handles around 2,000 transactions per second (tps), 4,000 tps in a daily peak, and 56,000 in a peak capacity. Visa itself never achieves more than about a third of 56,000 even during peak shopping periods. PayPal processed an average transaction rate of 50-100 tps in late 2014 while the Bitcoin is designed to process about 7 tps. This implies that at least our platform that is being proposed will be set to a target comparable to Visa's or even more. In order to achieve high performance computing and real-time payment comparable to or more than Visa's transaction processing speed, several exemplary protocols are thoroughly reviewed such as iota [16], Algorand [17], PBFT [19], ripple [20], and hashgraph [18] protocols because these candidates are to some degree known to produce phenomenal results even though the simulation and testing results were set up and tested in a limited testing environment. Finally, we choose a candidate from a Hedera Hashgraph because it is designed based on a classical asynchronous byzantine fault tolerance (aBFT) problem integrated with a gossip protocol and it is protected from patents such as gossip over gossip and a virtual voting concept, but it works only on a fixed number of nodes due to a private (or permissioned) blockchain platform.

2.2 Hedera Hashgraph

Among DAG based platforms, we choose the Hedera Hashgraph as a potential candidate to reinvent a powerful and robust platform called Babble Core because it proposes to solve the classical aBFT problem by applying Ben-Or's algorithm [21] and a gossip over gossip protocol with mathematical proofs while Radix and iota platforms seem to lack a synchronization logic among nodes to solve the aBFT issue. In 2017, Hashgraph beat over HyperLedger lead by IBM in a CULedger's benchmarking test presided by Credit Unions in the Unites States and a company named Swirds possess their own 3 US patents. It is known that the technology itself is perceived to be worth 6 billion US dollars in value, even in August 2018. In addition, some distinguished experimental results about performance and scalability were released to the public, and as a result, caused many attention and investment from blockchain communities, in which it will substitute the Bitcoin and commence a new era soon.

A Hashgraph protocol provides a new platform for distributed consensus, which is described as the future of superior distributed ledger technology. Major common attributes to own as a Blockchain are distributed, transparent, consensus-based, transactional and flexible, and Hashgraph bears all these features. It uses two special techniques to achieve fast, fair and secure consensus so that it uses a DAG based data structure as shown in Figure 2.2 and a

consensus algorithm using a gossip about gossip based on Ben-Orr's algorithm that is much faster, fairer, and more secure than blockchain. Gossip about Gossip means attaching a small additional amount of information to this Gossip, which are pointed with two hashes containing the last two nodes that it originated from.

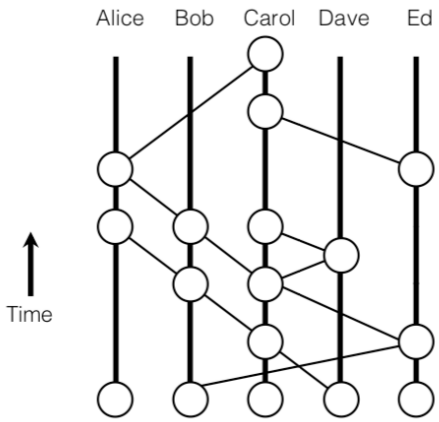


Figure 2.2 Directed Acyclic Graph

Using this information, Hashgraph can be built and regularly updated when more information is gossiped, on each node. Once the Hashgraph is ready, it is easy to know what a node would vote, since each node is aware of information that each node has and when they knew it. This data can thus be used as an input to the voting algorithm and to find out which transactions have reached consensus quickly.

The Hashgraph's consensus protocol as depicted in Figure 2.3 overcomes these inherent shortcomings as it requires neither Proof of Work nor a Leader. Moreover, it promises to deliver low cost and good performance with no single point of failure. This eliminates the requirement for massive computation and unsustainable energy consumption like those of Bitcoin and Ethereum.

```
run two loops in parallel:
    loop
        sync all known events to a random member
    end loop
    loop
        receive a sync
        create a new event
        call divideRounds
        call decideFame
        call findOrder
    end loop
```

Figure 2.3 Consensus protocol

- Fairness

In the blockchain world, a node playing a role as a miner can choose the order for which transactions occur in a block, it can delay orders by placing them in future blocks, even stop them entirely from entering the system. Consensus time stamping available with Hashgraph offers a solution to this problem. It prevents an individual from affecting the consensus order of transactions by denying any sort of manipulation of the order of the transactions.

- Asynchronous Byzantine Fault Tolerant

Unlike the other systems, Hashgraph is proven to be a fully asynchronous Byzantine Fault Tolerance. This means it makes no assumptions about how fast the messages are passed over the Internet unlike Bitcoin. This capability makes it resilient against distributed denial of service (DDoS) attacks, botnets, and firewalls. On the other hand, in Bitcoin, there is never a moment in time when you know that you have a consensus.

- 100% Efficient

No mined block ever becomes stale, whereas in the blockchain, transactions are put into containers (blocks) that form a single, long chain. If two mining nodes create two blocks at the same time, all the nodes in the network will eventually select one and discard the other, resulting in a waste of efforts. In Hashgraph, every container is used, and none are discarded.

Thus, although Hashgraph appears to be superior to ordinary block-based blockchains, it should be remembered that it is designed for a private blockchain system. We are targeting to maintain the pure spirits of Satoshi Nakamoto [5], thereby excluding private (or permissioned) blockchains and liberating some restrictions from patents and allowing all the communities to freely use it.

2.4 Reinventing a DAG-based Blockchain

To provide real-time payment and transactions even under a wireless mobile environment, we redesigned a DAG-based consensus protocol named as Babble Core [24] by adding some innovative ideas and functionality. We added some functionality by reinforcing gossiping (or

rumor spreading) with push and (conditional) pull features to accelerate information dissemination and determination, and to promptly reach consensus with the aid of a snapshot and an election with extinction to escape from Swirlds' patents and to encourage open source development and deployment. Babble Core's performance will outperform that of Visa's, which can target 100,000 transactions per second within 7 seconds and serving a global coverage.

In table 2.4, we propose some notable ideas to be reflected and implemented with DAGs. The following section will deal with these ideas in more detail.

Table 2.4 Some issues to be reflected

Issue	Details	Remarks
Avoiding infringement from patents	3 US patents [20] - US Patent: US 9,390,154 B1 (Jul. 12, 2016) (Methods and apparatus for a distributed database within a network) - US Patent: US 9,529,923 B1 (Dec. 27, 2016) - US patent: US 9,646,029 B1 (May 9, 2017) * Provisional application No. 62/211,411 filed on Aug. 28, 2015 1 PCT [21] - PCT/US2016/049067 (2016-08-26) - WO2017040313A1 (2017-03-09)	-Stake/Round -Gossip
Mobile	Enabling to process transactions in a blockchain among wireless mobile devices (smartphone)	
Sidechain	Proposing an inter-block chain at a node as a side chain to provide interconnection to other blockchain (reminiscent of an IP in Internet)	
Information dissemination	Smart gossip (push and conditional pull)	
Global snapshot	Sharing asynchronous common knowledge among nodes with Global snapshot and voting	
Leader election with extinction	- Electing a temporary leader - Reducing overhead during election	
Network topology	Applicable to dynamic and mobile networks, arbitrary networks where the total number of nodes are not known	
On-boarding	Proof of Trust (Public or permissionless)	

3 Babble Core

Proof of Trust

Don't employ anyone who is suspicious. If someone is employed, do not suspect him[2] (Korean proverb)

[2] 의인막용 용인물의 (疑人莫用 用人勿疑), 명심보감 성심편 (明心寶鑑 省心篇)

3.1 Partly distributed control

A centralized network and a peer-to-peer (or fully distributed) based network are shown in Figure 3.1 for comparison purposes. The centralized network consists of a central node and the rest of the nodes, which are linked to the central one. On the other hand, a pure peer-to-peer network as an overlay on top of the Internet has distinguished features such as self-organization and swarming where there is no centralized node, and in which it mediates and relays as a middle man. All the nodes in a (pure or full) peer-to-peer network are working in an autonomous way and all the nodes are equally considered. In terms of the points of failure and maintenance, the centralized network is easier to maintain as there is only a single point of failure, while the distributed one is the most difficult one to maintain. When it comes to fault tolerance and stability, the centralized one is highly unstable since in the case where the center node is down, it results in the whole network not functioning. However, the distributed network is very stable, and a single node failure does not have any effect on the operations. Regarding scalability, the centralized one has a very low scalability and the distributed one has a high scalability.

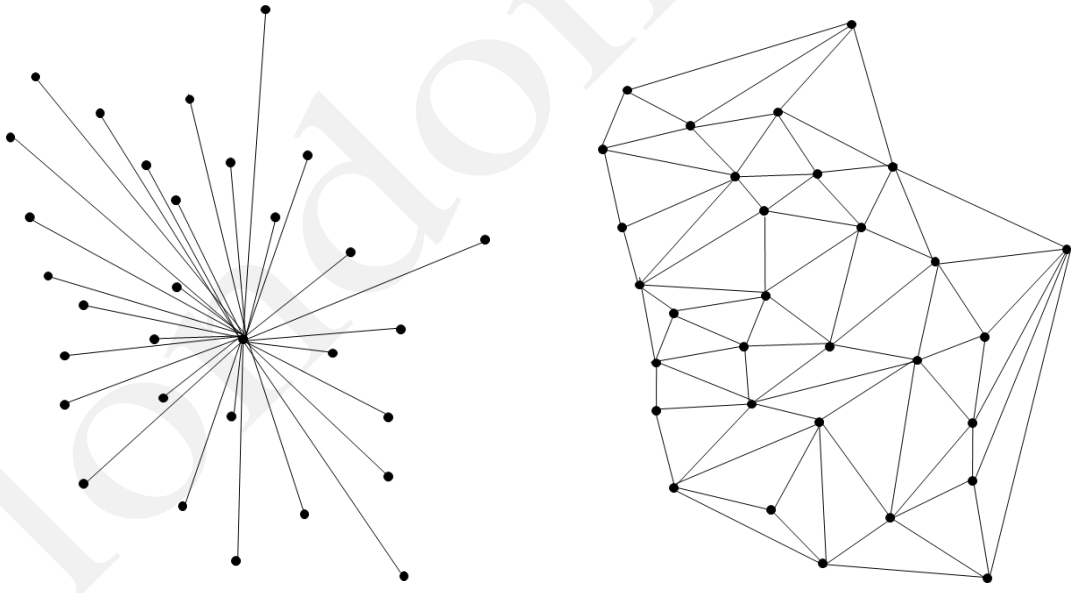


Figure 3.1: A centralized network and a peer-to-peer (fully distributed) one

In general, each node in a pure peer-to-peer network usually has overly excessive freedom and it runs its own control by communicating adjacent nodes and exchanging information such as control, data, and states, thereby resulting in too many traffic flows in the whole network and a late convergence in the whole network. To reduce the excessive

messages and controls over the network, running the pure peer-to-peer network is not a clever idea. Instead, a small network consisting of a part or a subset of a node set called a partly distributed (clustered or team or hybrid) network, which is obtained by partitioning the whole network, can be a better choice due to the actual experiences obtained from a mobile ad-hoc or a mesh network, wherein a node communicates to the other nodes without using any aids or relays via any special central node.

The partly distributed network consists of a set of clusters (or regions, groups, teams, shards), from which we schematically represent the partly distributed network in Figure 3.2. The dotted polygonal line represents an elected set of nodes, precisely a leader or a cluster head in each partitioned network. The leader node is the elected node depending on the satisfaction of certain conditions in each partitioned network. When the node is rejected in the election, then we call the node as a non-leader (plain) node, and the non-leader node can communicate with other nodes in its internal networks via its leader node working, which in turn works as a network gateway. And the non-leader node can also communicate with other nodes located at other partitioned networks via its leader node functioning as a network gateway. If possible, a set of the elected nodes can be formed as a clique (or complete graph) to rapidly exchange messages amongst nodes where each pair of nodes is directly connected by an edge with one diameter.

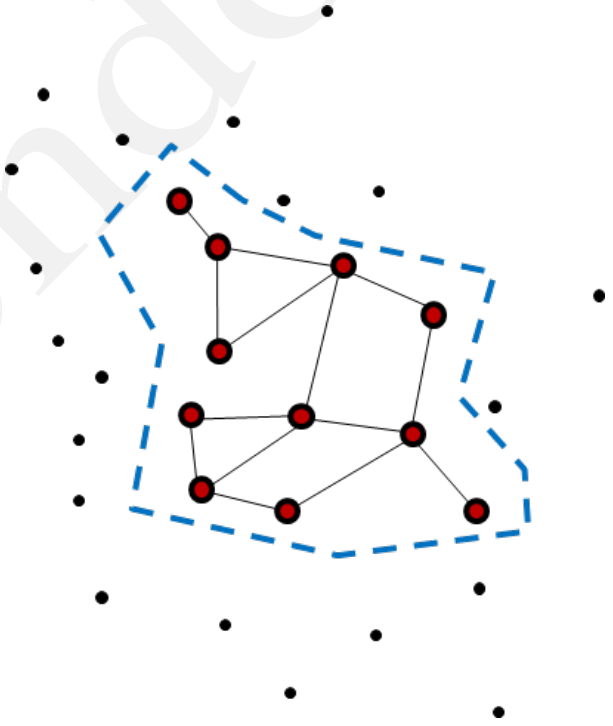


Figure 3.2: Partly distributed (clustered or hybrid) network

3.2 Deciding the Good, the Bad and the Ugly[3]

Unlike other cryptocurrencies, deciding good (honest) nodes or bad (dishonest) ones while forming the partly distributed network topology is a core concept of the Babble Core protocol. A decision process consists of four steps: Dissemination, Global snapshot, Election by extinction, and Consensus. The dissemination stage consists of the sharing amongst all nodes to have consistent information by using a gossip (or rumor) message. The global snapshot stage contains constructing explicitly a system-wide (or network-wide) global configuration featuring local states (snapshot states) of each process and channel states. By using the distributed snapshot, we can find the total number of elected nodes, the global timestamp, the trust level of the network, and more in a snapshot instant. The election by extinction stage is a process to choose an arbitrary node from the whole node set and all the nodes can be legitimately chosen depending on the satisfaction of qualifications. Within a finite time, a part or all the nodes that started the election process can finally be determined as one elected node so that the election process is faster than the normal election one and a certain node is not required to play the role as a designated node in advance. Finally, the consensus stage is a process where all the nodes reach an agreement. A node showing faults will be excluded from the chosen node set or it cannot be selected to belong to the chosen node set.

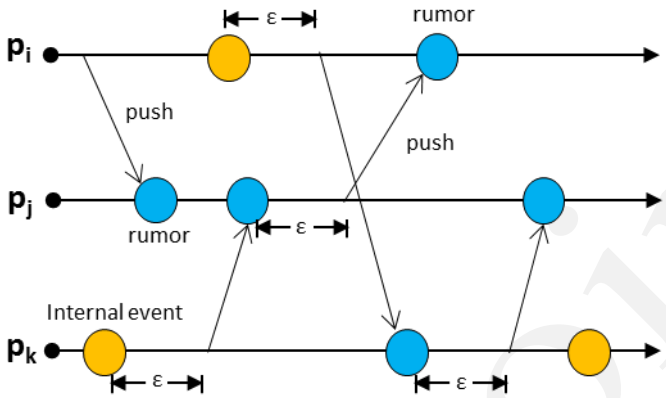
A. Dissemination

When a transaction has occurred, then the information of the transaction must be shared amongst all the nodes in a system. We are using a gossip (or a rumor) and a gossip about gossip (a rumor about rumor) to share the information because gossiping (or rumor spreading) is an efficient way of spreading in comparison with broadcasting and flooding [15]. We extend to an information dissemination protocol by using a push and (conditional) pull gossip message to reach fast convergence for all the nodes to have the same consistent information.

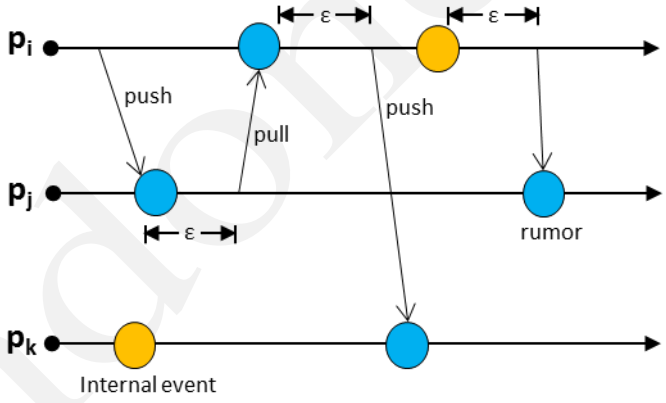
As shown in Figure 3.3(a), when a transaction known as an internal event (or a self-event) is created at a node P_i, and then the P_i randomly chooses adjacent node P_j and sends its local information to P_j within a time interval ε. At node P_j, when the gossip message receives, then it combines its own local information with the information from a remote

[3] We quoted the title of a famous 1966 Spaghetti Western film directed by Sergio Leone and starring Clint Eastwood, Lee Van Cleef, and Eli Wallach in their respective title roles since it matches with our metaphor.

node P_i. In Figure 3.3(b), P_i sends to a randomly chosen node P_j. If P_j knows that P_i have the same consistent information, then P_j does not respond to its gossip message anymore until the next gossip message has arrived. Otherwise, within an interval ε, node P_j packs its unshared information into a gossip message and sends it back to the sending node P_i.



(a) Push



(b) Push and conditional Pull

Figure 3.3: Gossiping (or rumor spreading)

The dissemination via gossiping autonomously proceeds without control from a certain node called the leader, and all the nodes voluntarily take participation in the dissemination process. There is no distinction between control and gossip messages. By piggybacking a gossip message with a special tag «*m*, *tag*», where *m* is a gossip message and a *tag* a special tagged information, our dissemination process can be extended to estimate a network size using aggregation computations and to detect termination of rounds for Byzantine Fault Tolerance (BFT). When it reaches to detect the termination of rounds, it can finally perform self-voting and confirm all the transactions just prior to an

instant of the termination detection.

B. Global snapshot

The notion of a cut underlies the construction of global snapshot algorithms [11, 12, 13, 14]. As shown in Figure 3.4, a cut divides the events of a system into those occurring before the cut and those occurring after the cut. Messages then travel between the "past" and the "future", as defined by the cut. A consistent cut is one in which no messages from the future travel into the past. Otherwise, we consider the cut inconsistent. To obtain a global snapshot, local snapshots are gathered from individual processes "along the cut". For the global snapshot to be meaningful, it is necessary that the protocol satisfies a consistent cut.

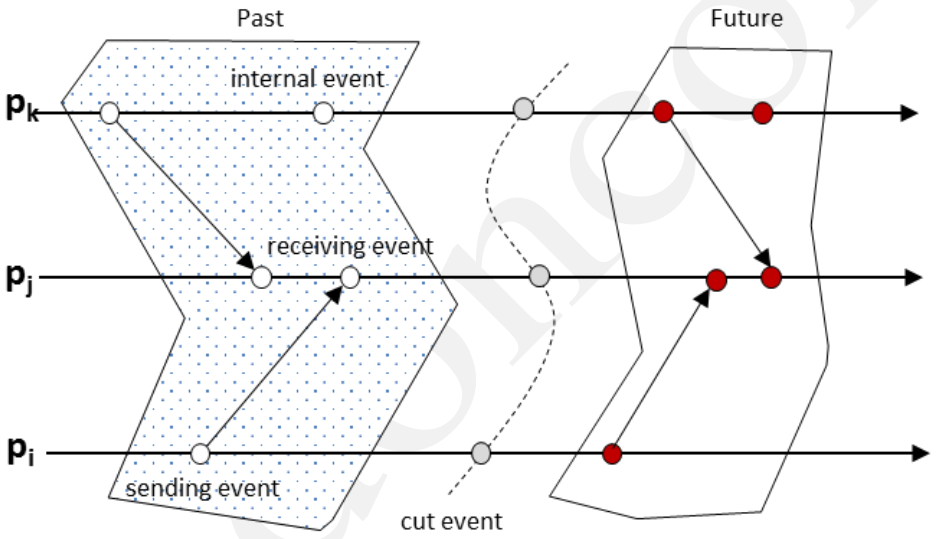


Figure 3.4: A cut message

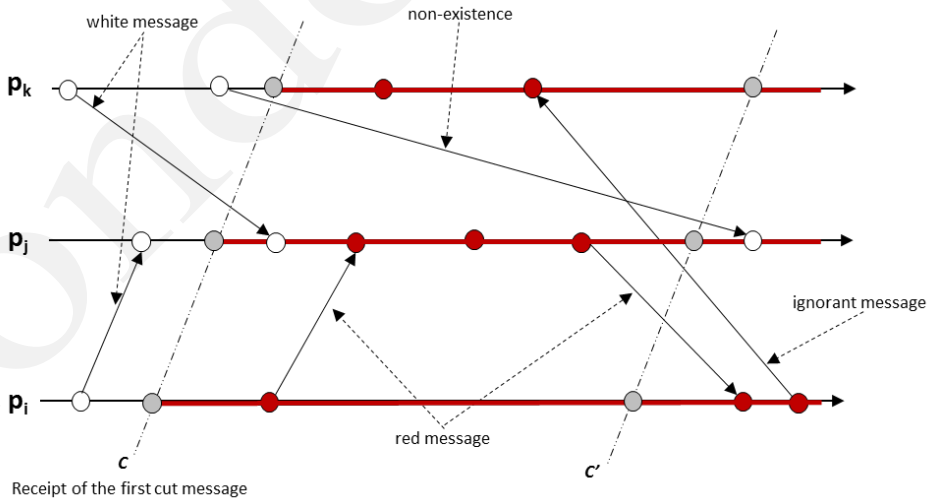
Lai and Yang [13] developed an elegant algorithm for obtaining such a cut. Their algorithm applies to non-FIFO systems, and only invokes the piggybacking of status information in one bit onto all messages. The algorithm is as follows: (1) every process is initially white and turns red when taking a local snapshot, (2) every message sent by a white (red) process is colored white (red), and (3) every process takes a local snapshot before a red message is received.

Ensuring that a local snapshot is taken before a red message is received at a process is accomplished by examining the color of the messages before processing them. If a message is red, the local snapshot is taken prior to processing the message. One way of implementing the algorithm is to circulate a control message, which colors each of the

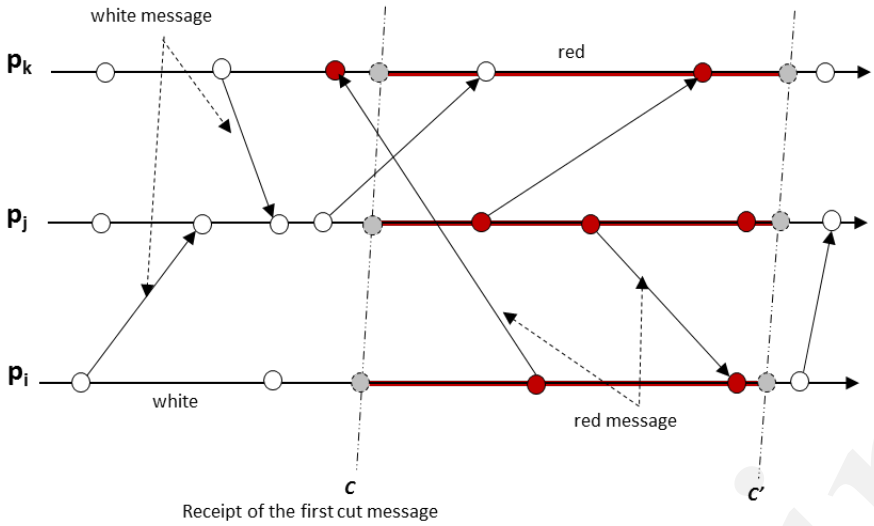
visited processes in red, *i.e.*, upon receipt of a control message, a process colors itself red as illustrated in Figure 3.5.

At the same time, the local state of the process can be appended to the control message (or sent directly to the process initiating the algorithms). However, it is possible that the white messages are in transit while the local snapshots are being collected. Consequently, it is necessary to record the states of the channels (or links). A way of doing this is for red processes to send copies of these messages to the initiator and to use termination detection algorithms to determine when they have all arrived. When the cut message is traversing all the nodes in an elected node set and gathers local snapshot and messages in transit, all the nodes have their own local and global information. Each node always maintains and announces its local information called a trust level for a node proof, which contains information like a node's computing power, history information, etc., when it performs its local snapshot. By a global snapshot, all the nodes share the total number of elected nodes and the current global time since a recent global snapshot.

When a node wants to participate in the election, it may be accepted or rejected according to its trust level. When a set of elected nodes needs to decide its actions, then based on all the gathered local and global snapshot information, nodes can vote for some decisions based on the collected information.



(a) An inconsistent cut message



(b) A consistent cut message

Figure 3.5: Global snapshot

C. Election by extinction

Initially, all the nodes are qualified to become a chosen node within a finite time. During the extinction, only a node can be chosen as a final leader. If a node or a link shows some failures, *i.e.*, the node itself and nodes connecting to the link are not working properly, then the nodes automatically lose their qualifications to be chosen. Once a node belongs to the set of the elected nodes, it will keep its status until it does not meet its trust qualifications during the run. Unelected nodes do not participate in a consensus work because in the case of Babble Core, we use a proof of trust mechanism. The total number of chosen nodes N in a cluster is fixed in advance or can be varied upon the number of faulty nodes or the required number of transactions. Initially, a low bound number of nodes is chosen for preventing faulty nodes from joining, and during the run, the number of chosen nodes is increasing till it reaches N. The chosen nodes are shared and compensated by a minimal amount of a transaction fee instead of issuing new coins like a Bitcoin, thereby preventing excessive competition for mining and resulting in saving computing power and electric energy, while it provides a proof of the transactions' correctness.

D. Consensus

There exist five types of failure modes as follows:

- Link failure: A link is said to be a failure if it remains inactive and the network

gets disconnected.

- Initially-dead process: A process is called initially dead if it does not execute a single step of its local program.
- Crash: A process is said to crash if it executes its local program correctly up to a certain moment and does not execute any step thereafter.
- Byzantine behavior: A process is said to be Byzantine if it executes steps that are arbitrary steps and, not in accordance with its local program. The Byzantine process sends messages with an arbitrary content.
- Timing error: There is an additional failure in synchronous distributed systems, where a process executes correct steps but at the wrong time due to a slow or fast clock of the process.

Fischer et al.'s impossibility result shows that under some conditions, which include the nodes acting in a deterministic manner, they proved that consensus is impossible with even a single faulty process. The impossibility result was proven for a specific model [9]. First, it introduces the idea of incentives, which is novel for a distributed consensus protocol. This is only possible in Bitcoin because it is a currency and therefore has a natural mechanism to incentivize participants to act honestly. Thus, Bitcoin does not quite solve the distributed consensus problems in a general sense, but it solves it in the specific context of a currency system. Second, Bitcoin embraces the notion of randomness. Also, it does away with the notion of a specific starting point and ending point for consensus. Instead, consensus takes place over a long time, about an hour in the practical system.

Let f be the maximum number of faulty nodes. An initially-dead process is no longer a problem because it cannot be chosen and there are no initially-dead nodes in the elected node set. In the case of a link failure, it is impossible to reach consensus even in the synchronous case, and even if one only wants to tolerate a single link failure. Fortunately, the node with a link failure or unreliable link cannot be chosen as an elected node or it can be purged from the elected node set [6, 8].

As proven by Fisher et al. [9], there are no asynchronous, deterministic 1-crash robust consensus protocols known as the impossibility of consensus. To determine crash failures, at least f+1 rounds of message broadcast (or flooding) and f+1 number of elected nodes are required while in the Byzantine failure, f+1 rounds of message

broadcast and $3f+1$ number of elected nodes are also needed. To determine a minimal number of nodes to be chosen as an elected node initially, at least $3f+1$ nodes are required. The number of elected nodes can be explicitly indicated in an expression (1) where N is the total number of nodes in an elected node pool.

$$3f+1 \leq \text{number of elected nodes} \leq N \quad (1)$$

After broadcasting (flooding) messages in $f+1$ number of rounds, every non-faulty node knows about all the values of all other participating nodes, thereby deciding the same value even under the occurrence of crash and Byzantine failures at nodes.

$$f+1 \leq \text{number of rounds of cut message} \quad (2)$$

3.3 Protocol

A. Configuration

We describe configurations in our cryptocurrency network as follows:

- (Elected) Node: Once a node is elected, then the node runs entire functions requiring for the cryptocurrency processing.
- Unelected node: Although it can run all the functions, it is not chosen as an elected node due to the limited number of available nodes at that instant. Once an elected node is to be proven that it is faulty or that it has crashed, then an unelected node can challenge to become a node again.
- Distributed ledger: The ledger containing all the transactions are recorded in a repository of the elected nodes where all the transaction records are stored in a chronological sequence and opened to all the users and unelected nodes. The whole distributed ledger at each elected node can be stored by compression for efficiency.

B. Procedures

We explain some of the details about 4 major procedures.

- **Dissemination**
  - A push and (conditional) pull gossip message is sent to the randomly chosen node.

- By piggybacking, network size and termination detection of rounds can be determined for fast confirmations on transactions.
- For disseminating information, it is not required to elect a special node as a leader.

➢ **Global snapshot**

- Cut messages periodically are traversed from the message initiator to the rest of the elected nodes. When the cut message visits each node, it records its local state and visits adjacent nodes till it returns to its initiator.
- During the cut message transversal, all the local states are collected and shared amongst the whole nodes. We are set to a one second time lapse for each global snapshot. During the interval, the global snapshot and consensus agreement are performing to obtain common information and detect node failure.

➢ **Election by extinction**

- By introducing a concept of extinction, the number of required messages to determine the elected nodes can be reduced.
- Initially, some nodes with a high trust level as a proof of trust are only accepted and then they become elected nodes until it reaches the total number of nodes N in an elected node pool. All the works needed for cryptocurrency are processed among elected nodes.
- Some malicious nodes can be screened by the election process.
- By reducing unnecessary nodes and overly excessive freedom, performance will be enhanced.
- It can be extensible to both of permissioned (public) and permissionless (private) blockchain systems.
- There are incentives for elected nodes, which can provide confirmation services for proof of trust. When a cryptocurrency is transferred to other cryptocurrencies or fiat currency ones, part of a transaction fee will be shared for the elected nodes.

➢ **Consensus**

- During the global snapshot, node failure can be detected via the elected nodes by circulating rounds of cut messages.

- Some internal and external threats can be eliminated by consensus.
- Consensus gives a low bound of minimal number of nodes and number of rounds of message gossiping (or rumor spreading) for making a decision given a condition of *f* faulty nodes.
- Due to global snapshot, a node causing a timing error can be eliminated.

C. Correctness proof

We proved the correctness of the protocol by establishing the safety and the liveness of the protocol. Safety corresponds to the protocol producing an estimate, which is less than (or equal to) the exact global time *GT*. Liveness corresponds to the protocol producing monotonically increasing estimates. We first establish the safety property. Let *GT(t)* be the exact *GT* at time *t* and *GT(t)* be the approximate *GT* as computed by our protocol at time *t.*

THEOREM 4.1 (SAFETY) Let *t* be the instant at which *GT(t)* is computed. The $GT(t) \leq GT(t)$.

PROOF. *GT(t)* is computed by the initiator $\Leftrightarrow$ *count* = 0. *count* = 0 $\Leftrightarrow$ there are no white messages in transit. Hence, we need only concern ourselves with the timestamps of red messages in transit when computing the *GT*, *i.e., GT(t) = min*{timestamp of all nodes at time t, timestamps of red messages in transit at time *t*}. From the protocol,

$$GT(t) = min\{min(lt), min(ts)\}$$

where *min(lt)* = minimum of the nodes' timestamps for all the nodes, *i.e*., *lt* = each local node's timestamp, and *min(ts)* = minimum timestamps of all the red message since each node became red. The $min(ts) \leq$ timestamps of all red messages in transit at *t* since the red messages in transit at time *t* form a subset of all the red messages sent since each node became red. Furthermore, at time *t,* no *lt* can be less than the minimum timestamp of the red messages in transit at time *t*. (These are the only messages which can roll back a node since the cut message has visited all the nodes except the initiator prior to time *t*.) Hence, we conclude that $GT(t) \leq GT(t)$. ■

We now establish the liveness of the protocol.

THEOREM 4.2 (LIVENESS) *if* $t_1 < t_2$, then $GT(t_1) \leq GT(t_2)$.

PROOF. After the computation of $GT(t_1)$, it is possible for one of the nodes to be rolled back by a red message, but not by a white message (the white messages have all arrived). However, the minimum timestamp of the red messages in transit is included in the definition of *GT(t)*

and by virtue of this definition, the $GT(t_2)$ cannot decrease subsequent to the computation of $GT(t_1)$, the theorem follows. ■

THEOREM 4.3 Node coloring and choosing a leader in the course of node election can be achieved within a finite time.

PROOF. Suppose that the channels in the network have a finite transmission time, that transmission is fault-free, and that a node takes finite time δ to be colored. If all the nodes begin to color at the same instant, the time for coloring will be δ. Otherwise, if the nodes are colored sequentially, in the worst case, it takes $N\delta+\varepsilon$, where ε is the time for the cut event to traverse the network and N total number of nodes participating in the snapshot. Therefore, choosing a leader requires time $\leq N\delta+\varepsilon$. ■

3.4 Comparisons

Table 3.1 compares features of Hedera Hashgraph with our Babble Core. The Babble Core is providing preventions of all the failures because it is designed to be working even under a public blockchain and a mobile network, thereby guaranteeing to run well even under all kinds of network configurations and security threats.

Table 3.1 Comparison of Hashgraph and Babble Core

Features	Hedera Hashgraph	**Babble Core**
Fairness	No (master shard, proof of stake)	Yes (no master shard and no proof of stake)
Info. Dissemination	Push	Push and (conditional) pull
Network configuration	Static and fixed	Dynamic and adaptive (self-organizing)
Convergence detection for dissemination	No (normal round and coin round)	Yes (Snapshot)
Detection for immutable transactions	No	Yes (Snapshot)
System launching	Synchronous launch (all the nodes are starting at the same time)	Asynchronous launch
Participation	Permissioned/Pre-admission	Public
Load balancing	Yes (Static)	Yes (Dynamic)
Preventing failures (Link failure, Crash, Byzantine behavior,	Byzantine failure	All the failures

Timing error)		
Action for security threats	Passive	Active
On-boarding	Requiring one day	Real-time
Supporting arbitrary networks (total number of nodes is not known)	No	Yes

4 Conclusions

In this paper, we introduced an ultimate blockchain engine, Babble Core to accomplish a true crypto-economy, by particularly realizing scalability as a steamroller, which can provide dominant features into the crypto-economy.

First, our babble core is designed to enable and to support real-time payments and transactions, which is more powerful than VISA's current processing rate by focusing on the scalability aspect. To realize the goal, we are using the following Babble Core's open protocol: 1) extending to a permissionless and public blockchain system only from a permissioned and private one, by using a partly distributed control which can decide between good nodes and bad ones based on the cooperation amongst nodes in the whole network; 2) unlike other cryptocurrencies, only the chosen nodes satisfying a certain trust level can maintain its number of active nodes by adding or purging nodes in the set. Basically, if some nodes with powerful computing resources may collude with other nodes, then they may be not elected or purged from the elected active node list; 3) rapidly reaching information dissemination by changing from a push-based gossip to a push and (conditional) pull based gossiping (or rumor spreading); and 4) promptly deciding to reach a consensus by a global snapshot.

Secondly, we are not encouraging competitive mining from all the participating nodes. In a PoW-based consensus, only one node can have additional Bitcoins as eventual winning compensation by enormously consuming electric energy and computing resources. On the other hand, only the chosen nodes that are proven to be trustworthy can participate in the proof of trust, and all the nodes share a transaction fee for transfer payment and a node fee for connecting nodes to access a blockchain as incentives.

Finally, we expect to make a debut for realizing crypto-economy soon and hope that our Babble Core can resolve many uncertainties and concerns in the crypto-economy.

Accompanying with our Babble Core blockchain platform, we also delivered a wallet called ABM [25] into a market, which can easily send and receive cryptocurrencies between cryptocurrencies' owners using a phone number unlike ordinary ones.

References

1. Tapscott, D. and Tapscott, A., "BLOCKCHAIN REVOLUTION: How the Technology behind Bitcoin is Changing Money, Business, and the World", Penguin Random House LLC, 2016.

2. Mougayar, W., "The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology", John Wiley & Sons, 2016.

3. Narayanan, A. et al., "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", Princeton University Press, 2016.

4. Drescher, D., "Blockchain Basics: A Non-Technical Introduction in 25 Steps", APress, 2017.

5. Nakamoto, Satoshi, "*Bitcoin: A Peer-to-Peer Electronic Cash System*", https://bitcoin.org/bitcoin.pdf, 2008.

6. CISCO, "*Visual Networking Index: Forecast and Methodology, 2016-2021*", June 2017.

7. Bondi, A., "*Characteristics of scalability and their impact on performance",* Proceedings of the second international workshop on Software and performance, WOSP '00, page 195, 2000.

8. Lamport, L., Shostak, R. and Pease, M., "*The Byzantine Generals Problems*", ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, pages 382-401, 1982.

9. Fischer, M., Lynch, N. and Paterson, M., "*Impossibility of Distributed Consensus with One Faulty*", Journal of the ACM, Vol. 32, No. 2, pages 374-382, 1985.

10. Attiya, C., Dolev, D., and Gill, J., "*Asynchronous Byzantine Agreement*", In Proceedings, the 3rd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, pages 119-133, 1984.

11. Lamport, L., "*Time, clocks, and the ordering of events in a distributed system*", Communications of the ACM, vol. 21, pages 558-565, 1978.

12. Chandy, M. and Lamport, L., "*Distributed Snapshots: Determining Global States of Distributed Systems*", ACM Trans. on Computer Systems, Vol. 3, No. 1, pages 63-75, 1985.

13. Lai, T. and Yang, T., "*On Distributed Snapshots*", Information Processing Letters, Vol. 25, No. 1, pages 153-158, 1987.

14. Mattern, F., "*Efficient Algorithms for Distributed Snapshots and Global Virtual Time Approximation*", Journal of Parallel and Distributed Computing, Vol. 18, pages 423-434, 1993.

15. Demers, A. et al., "*Epidemic Algorithms for Replicated Database Maintenance*". Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC), pages 1-12, 1987.

16. Popov, Serguei, "*The Tangle*", https://iota.org/IOTA_Whitepaper.pdf, 2017.

17. Micali, Sylvio, "*ALGORAND The Efficient Public Ledger*", CSAIL, MIT, https://arxiv.org/pdf/1607.01341v7.pdf, 2016.

18. Baird, Leemon et al., "*Hedera: A Governing Council and Public Hashgraph Network*", https://s3.amazonaws.com/hedera-hashgraph/hh-whitepaper-v1.0-180313.pdf, Mar. 2018.

19. Castro, M. and Liskov, B., "*Practical Byzantine Fault Tolerance*", Proceedings of the Third Symposium on Operating Systems Design and Implementation, Feb. 1999.

20. Schwartz, D. et al., "*The Ripple Protocol Consensus Algorithm*", Ripple Labs Inc., https://ripple.com/files/ripple_consensus_whitepaper.pdf, 2014.

21. Ben-Or, M. "*Another advantage of free choice (extended abstract): Completely asynchronous agreement protocols*", In Proceedings of the second annual ACM symposium on Principles of distributed computing, pages 27–30, ACM, 1983.

22. https://www.swirlds.com/ip/

23. https://patents.google.com/patent/WO2017040313A1/en

24. Babble Core, https://github.com/mosaicnetworks/babble

25. ABM Wallet App, https://play.google.com/store/apps/details?id=com.sikoba.wallet&hl=en&gl=US