Toward Achieving Unanimity for Implicit Closings in a Trustless System

Mitsuyoshi Imamura[1(✉)] and Kazumasa Omote[1,2]

[1] University of Tsukuba, Tennodai 1-1-1, Tsukuba 305-8573, Japan
s1730148@s.tsukuba.ac.jp, omote@risk.tsukuba.ac.jp
[2] National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

Abstract. Blockchain, which has a decentralized management structure, is a technology that challenges conventional wisdom about the availability and durability of an unstable structure, because the network system is managed by volunteers, as opposed to cloud- and network-service providers that leverage centralized management structures. The most popular services based on blockchain (e.g., Bitcoin and Ethereum) have structures that make it challenging to close or discontinue services unless all users agree, no matter if they are honest or malicious. Remarkably, this structure shows stable availability and durability, even now. Considering that unpopular services are eventually terminated, there are more than 2,000 service projects forked from Bitcoin and Ethereum, and it is not realistic to expect all of them to operate in the same manner. When users abandon services like these because of low interest, the blockchain, which depends on volunteers to maintain the system, is affected by reduced availability and durability until it is finally closed. However, unlike centralized management organizations, for service closings, both the indicator and closing mechanism are unclear, because management depends on user dynamism. Therefore, we investigate the mechanism of public blockchain closing by focusing on three decentralized roles of blockchain users. Then, we discuss the closing implication of blockchain-based services using the empirical analysis of 200 different systems.

Keywords: Blockchain · Cryptocurrency · Consensus of closing blockchain

1 Introduction

More than 10 years ago, Satoshi Nakamoto first published the now well-known blockchain whitepaper [20] as a new paradigm of a trust structure that differed from the previous centralized trust structures. The technological challenge of managing a system used by an unspecified number of non-trusted users has been commonly encountered over the past few years. Every year, the self-sovereignty of

X. Yuan et al. (Eds.): QShine 2021, LNICST 402, pp. 75–93, 2021.
https://doi.org/10.1007/978-3-030-91424-0_5

end-users over their devices and data has accelerated the adherence to decentralized structures, because distrust has grown around centralized ones. Although previous studies [1] have reported that blockchains are potentially excellent technology, as the number of users increase, participants face greater uncertainty of performance, because the operation reflects the harmony of all users. With centralized structures, keep in mind that a planned-performance design is curated.

Typically, cloud- and network-service providers use centralized structures to control performance and availability via the direct management of resources, and they systematically abandon the durability of infrequently used services. In contrast, the public blockchain has an unstable structure, because performance, availability, and durability depend on user behavior. From their decentralized natures, leading blockchain-based services (e.g., Bitcoin and Ethereum) experience project forking caused by disagreements within the community. This cannot be prevented unless all users agree to closing. Hence, various projects persevere without losing the original data.

In particular, we focus on the fixing of the blockchain at Ethereum that triggered "The DAO attack," during which, vast amounts of assets were stolen on July 20, 2016, owing to a vulnerability in the program [9]. This incident is a notable example of how difficult it can be to close a service without the full agreement of all users of the blockchain. Looking back on the situation at the time, although 80% of users managing Ethereum nodes used a ledger that allowed modifications, another group continued to use the original ledger that did not apply it. Currently, each exchange lists Ethereum Classic based on their unmodified ledger, and the community declared independence from the modified version on the official website, and the developer was also informed that the project would continue to be developed individually on August 2016. Therefore, Ethereum Classic remains robustly available and persistent, despite Ethereum being a majority opinion project.

Because there are more than 2,000 blockchain projects derived from Bitcoin, it is not realistic to expect all projects to continue running in the same manner, despite being difficult to close or modify. This is rational, given the physical resources and costs shouldered by users to maintain the blockchain. Unlike a planned closing promulgated by a centralized management organization, the mechanisms leading to decentralized closings remain unclear, because management depends on user dynamism.

The conclusion that networks lacking interest and dynamism are closed is intuitively easy to understand. However, researching the validity of this conclusion is outside of the scope of this study and irrelevant to our focus of understanding the process of this closure, including the signs of closure and the impact of closure on the network. Our intended scope is beneficial because of two reasons.

First, understanding the closure process will help in designing blockchains that have strict closure mechanisms. Although it may seem strange to incorporate an erasure mechanism into a blockchain that is data-robust, we have encountered several situations where closure is necessary. In a hard fork, which requires the approval of a large number of stakeholders, a mechanism of fork loca-

tion is necessary to avoid the confusion caused by value dichotomization. Public blockchains must completely close, owing to security risks, such as the misuse of blockchain availability [2,6] and the poisoning of ledgers [16,19]. Additionally, a general data protection regulation (GDPR) that includes the right to be forgotten [22] should be achieved, even under a decentralized system having no party responsible for protecting users from data embedded in the ledger. This means that the issue occurs after the service is used. Hence, consolidating the majority agreement for the ledger is very time consuming. Thus, a substantial amount of time must pass before everyone loses interest, leaving zero users for consensus. However, in previous blockchain communities, this situation has indeed been left to the passage of time, owing to the decentralized responsibility. Reducing this time not only minimizes the loss of service caused by community disruption, but it also reduces the ongoing risk present in the ledger. It should be noted that although implementing a mechanism for global consensus may seem difficult, this type of closure mechanism already exists. For example, forking occurs daily and is commonplace owing to competition in mining. However, the reason why forked blocks do not remain in the network is that a clear closure mechanism exists between the distributed nodes maintaining the ledger, which keeps only the longest blocks. In summary, we aim to extend this mechanism via this study.

The second benefit is that understanding the closure process provides a benchmark against which continuity can be measured. The paper deals primarily with public-cryptography-focused blockchains, but we envision business, commercial, and industrial compliance as well. Future works can expand the research scope of permitted blockchains. For example, researchers should seek to determine whether or not, if one of the main partners drops, the continuity of the blockchain will be affected. In other words, how do user critiques manifest in the system when there is no price to be linked to cryptocurrencies or tokens. Additionally, we assume that understanding the closure process helps define the user policy of blockchains for the industrial use case if metrics were to exist that could be used to finally close the service forms between users prior to releasing the service. As is evident in public blockchains [9], it is very difficult to achieve consensus after a service has been launched. Therefore, postponing the issue presents risk to all participants by wasting time and money owing to confusion.

In this paper, closing public blockchains has been discussed and the setups required to do so on a trustless network have been summarized. Our proposal entails a using a method that applies a time constraint on the validity of the user's traceability and consensus. It clarifies the rational framework under which an unspecified number of users agreed to terminate blockchain ledger, rather than it being modified or deleted by some strong authority, such as consortium or private blockchain.

We begin by investigating the mechanisms of service closure by focusing on the three decentralized user roles in the blockchain (i.e., sender, miner, and recorder). Then, we discuss network dynamism and the consensus models behind the blockchain system by leveraging the empirical analysis of 200 different systems. The main contributions to our research are as follows:

– We propose a new framework for blockchain-closing consensus.
– We analyze the transactions and block trends of near-closure blockchains.
– We visualize the propagation flow toward consensus of blockchain closure.
– We discuss the implementation of the modeled consensus-closing mechanism.

The remainder of this paper is organized as follows. Section 2 presents the blockchain mechanism, focusing on the distributed user roles to understand the consensus required for closing. Section 3 introduces previous research related to closing blockchains. In Sect. 4, we propose a framework for reconsidering this consensus. Section 5 analyzes real networks to validate the new framework. Section 6 presents new implementation ideas based on the framework and empirical analysis. Finally, we present our conclusions in Sect. 7.

2 Blockchain Mechanism

This section introduces the necessary blockchain components and operational mechanisms necessary to form a unanimity for closing. We emphasize the cycle of roles in the blockchain network to demonstrate the user motivation and rational behavior mechanisms for system continuity. Furthermore, we explain the common cryptocurrency-type blockchain that we target in our empirical analysis.

2.1 Blockchain Overview

The key components of a blockchain are the transaction and the block. In this section, we explain the common components of public blockchains based on Bitcoin. A transaction consists of ledger data and inputs of the transferred value. It is the smallest data-structure unit in the blockchain. The block also contains the block version, the height, Merkle tree root hash value, and timestamp in addition to the verified transactions and a hash value calculated from the previous block. Among all blocks, a "genesis block" is a pointer that defines the initial position in the list structure that connects future blocks. The blocks linked by the hash are continuous, characterizing the blockchain's unchanging data structure, and this characteristic structure contributes to its tamper-resistance feature. Distributed nodes that replicate all blocks support the network's worldwide data consistency, and consistent storage is provided by users connected to the network. Transaction creation and block storage are thus repeated across the network.

According to previous research [14], the flow that the original paper [20] describes as a step on the system, structured as a user's roles, is represented by the cyclic structure shown in Fig. 1. Blockchain users have three roles: the sender generates transactions; the miner verifies the transactions and stores it in a block; and the recorder saves the latest blocks. Note that users may have duplicate roles. In the cycle presented in Fig. 1, the function of the three roles as follows:

– The sender obtains the block required for signature from the recorder, creates new transactions, and sends these to the miner.

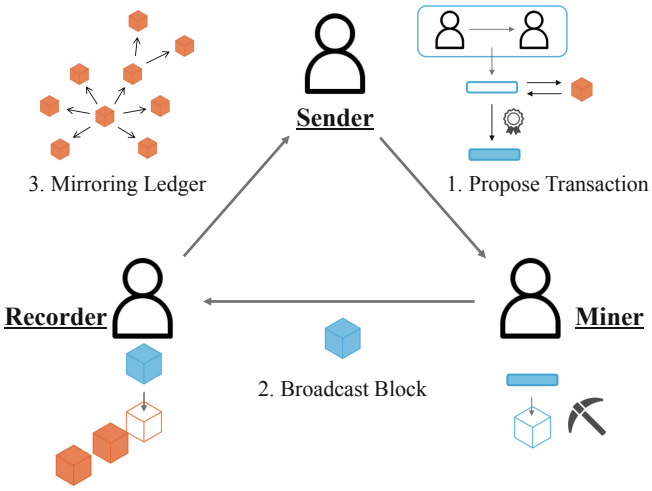

Fig. 1. Cycle of user roles in blockchain.

- The miner collects the new transactions to store in a block and locates a nonce of proof-of-work (PoW) for its block to store the transaction. When the miner finds a nonce, it broadcasts the block back to the recorder.
- The recorder accepts the block only if all transactions therein are valid and not already spent. It then mirrors the latest block status to other nodes on the network. The recorder's mirroring path duplicates the ledger and creates a propagation path that is used in transaction passing by the sender and the miner on the blockchain network.

We focus on the driving force of the cycle in which the blockchain network operates. For a blockchain to exist, miner and recorder roles are required to validate and store transactions. The miner receives a fee from the sender with a reward for the verification work, thereby providing the miner with motivation to play the role. However, the recorder can only use the correct data and will never receive any money for broadcasting the latest blocks to the network. Therefore, the recorder tries to work at a low cost to maintain storage via the same behavior that the miner uses to work at a low cost to receive more rewards. If recorders give up managing the ledger due to maintenance costs, the availability and durability of the network is reduced. Then, the sender regards the service as unlikely to survive. Consequently, we can understand the non-programmable background mechanism by which the users on the blockchain network tend to behave rationally.

2.2 Cryptocurrency

Widely known as the original cryptocurrency, Bitcoin started on January 9, 2009, when Satoshi Nakamoto, who is an anonymous developer, announced the

concept in an email[1] to the developer community, sharing it on SOURCEFORGE[2]. A well-known feature of Bitcoin is its limitation on the participation of malicious users based on the requirement to solve the Byzantine consensus problem using a PoW process, referred to as HashCash [5]. This protocol provides a defense mechanism that prevents Sybil attacks [4] and a lottery mechanism that selects a random leader at each round via competition among users. For this reason, the PoW is known as the "Nakamoto Consensus." It is also an aspect of financial services, because the numbers generated mathematically and cryptographically inside the blockchain have a quantitative meaning. Using this concept as a baseline, several developers have proposed new cryptocurrencies adapted to new challenges and functional expansions.

Cryptocurrencies that were forked or inspired by Bitcoin are classified into two main types. The first is such as Litecoin, which only inherits the currency implications of the blockchain and improves Bitcoin throughput. Because we know that the Nakamoto Consensus requires low-transaction throughput, high latency, and energy inefficiency, PeerCoin was the first to implement an alternative consensus algorithm (Proof-of-Stake). Other cryptocurrencies include Dash, Monero, and Zcash, which provide more anonymity in transactions.

Another type of cryptocurrency is the implementation of roles and functions, instead of just monetary implications. Ethereum is a leading cryptocurrency that implements smart contracts and has generic functions. Furthermore, Storj provides cloud-storage capabilities, and Namecoin has the equivalent of a domain name service.

3 Related Work

Although there are mechanisms for closing the consortium blockchain because of resource limitations [8], we could not find any direct research related to the requirements of closing public blockchains. However, we located secondary research that included the context of blockchain closing.

Bartoletti et al. [7] measured project mortality based on source-code updates from 120 blockchain-related social-good projects on GitHub. The projects were declared "abandoned" by a third party if their websites were down or if old content was not updated. Fantazzini et al. [10] estimated credit risk in financial-asset portfolios for cryptocurrencies. In their model design, they considered "coin death." If the value drops below one cent with no trading volume and no nodes on the network, no active community, no listing from all exchanges, and no thresholds for price peaks, it dies. However, the death is not permanent. A coin can revive many times. Guo et al. [12] investigated whether wealth distribution was an important factor in determining whether a cryptocurrency survives and gains popularity using a power-law model for Bitcoin and Auroracoin. They showed that some features leading to the death of many coins included complexity of the design of the block-reward scheme, disappearance of the developer, and malware

[1] https://www.mail-archive.com/cryptography@metzdowd.com/msg10142.html.
[2] sourceforge.net.

attacks set up by keyloggers and wallet thieves. They also showed that cryptocurrencies were unique and not clones or forks of other cryptocurrencies. The names were not duplicated, and the timing of the listing on the exchange was not bad.

Secondary research shows a common trend of using external information and financial data rather than referring to the system's running status. DeadCoins[3] and Coinopsies[4] are the leading external sources for identifying and locating dead coins. Both media published their lists with evidence, including screenshots and links submitted by users. These studies do not necessarily imply closings in the blockchain, however, we found that they provide empirical evidence of closing consensus in the community.

Studies on the blockchain network availability suggest the state of the system when discussing closing, and research that targets user mechanisms in the system is helpful in understanding the triggers and dynamics of stopping use. In addition, although our study assumes that users do not voluntarily use the blockchain based on the perspective that it is not available, malicious attacks on availability help our discussion as they create a similar situation.

Imamura et al. [14] reported that user behavior was based on economic rationality, owing to storage costs and network maintenance. This means that maintaining a system that is almost dead is irrational because the rational user behavior was positive. Motlagh et al. [17] used continuous-time Markov chains to cover four states (i.e., sleep, getting headers, waking up, and operationally modeling the churning process of nodes). Referring to their research, understanding the potential churn process is an essential milestone in considering the unanimity required to close a blockchain.

Heilman et al. [13] reported an Eclipse attack that isolated nodes by blocking peer-to-peer (P2P) network connections. Apostolaki et al. [3] reported a border-gateway-protocol hijacking attack that similarly caused a partition of the P2P network, leading to the isolation of a specific attack target. Tran et al. [21] reported an extended Erebus attack that partitioned the network without routing manipulations of the larger network. These studies did not express closings, but they reported situations that were more-or-less equivalent in that the volunteer nodes were no longer connected, and the nodes could not refer to the original ledger. The discovery of new blocks and nodes is an empirical factor related to closing a blockchain. The difference between these attacks and the closing blockchain is that they are not temporarily unreferenced, but they are instead permanently unreferenced.

In terms of new-block propagation, a block-delay attack represents a similar situation. Gervais et al. [11] proposed a method to reduce availability without splitting the network by exploiting the application's protocol to delay the propagation of blocks to the attack-target nodes. Walck et al. [23] reported a high-probability block-delay attack by hiding malicious nodes near the network in which the attack-target nodes were built. Based on these reports, even if a

[3] https://deadcoins.com/.
[4] https://www.coinopsy.com/dead-coins/.

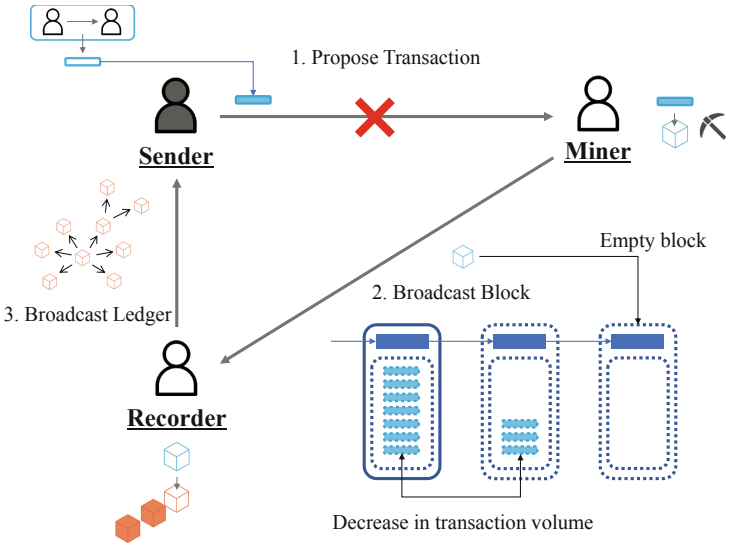

Fig. 2. Cycle of user role when the sender's consent is in the blockchain.

relay network is maintained, network availability can be reduced to nearly zero if there are problems with block propagation.

In summary, the availability of the network is achieved by rational property. Therefore, our starting point for thinking about closing blockchain is to model the processes that lead to irrational conditions. The properties needed to model closing blockchain is referred to as the block and node states in an attack.

4 New Framework for Closing Blockchains

To discretize the unanimity of closing blockchains within the community and to organize the contributing factors, we propose a framework based on the cycle described in Sect. 2.

First, closing the blockchain requires all users to agree with. However, there are no explicit rules on how to gain this unanimity. One method is to build rules around user behavior (e.g., joining and leaving) that can represent implicit user consent. According to Fig. 1, the three user roles include sender, miner, and recorder. If all three roles continue to be filled, they essentially agree to maintain the blockchain, and the cycle runs smoothly. However, if one of these user roles goes unfilled, the closing process begins. The abandonment of all three roles is equivalent to unanimity. The following subsections describe each step of closing a blockchain as implied by the three user roles, focusing on abandonment behaviors and the impact on the cycle.

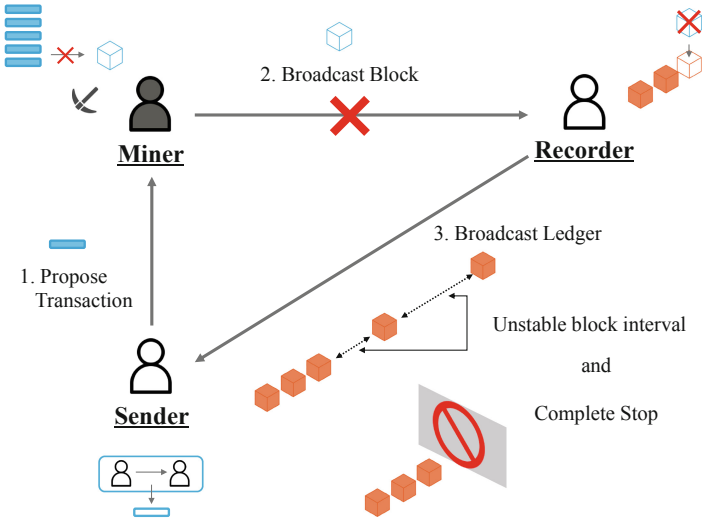

Fig. 3. Cycle of user role when the miner's consent is in the blockchain.

4.1 Sender's Consent

Here, we explain the case where only the sender consents to closing the blockchain. Because the sender's role is to propose transactions, giving up the role means that transaction proposes will cease. Figure 2 shows how the cycle is affected by the sender's role going unfilled, thus stopping the proposes.

First, when the sender stops proposing, the number of transactions stored by the miner in the block gradually reduces. Eventually, an empty block is created. The cost of storing an empty block is lower than that of storing a full block, because the recorder always keeps an empty block. This means that, in many of the protocols applied by the blockchain, the miner does not receive a transfer fee, which is economically damaging. In the case of the recorder, the cost of storage is lower than when storing a full block, because the recorder always saves an empty one. Regardless, it is significant that the largest user role, the sender, no longer exists. Importantly, the miner and the recorder roles recognize the lack of sender participation by examining the number of transactions. In conjunction with the sender's implied signal of closing consent, this suggests that the miner and the recorder will be close behind.

4.2 Miner's Consent

When only miners imply their closing consent, it helps to understand the hard-fork situation that occurred with Ethereum. The role of the miner is to validate the broadcast transactions in the blockchain and to record them in a new block. Hence, giving up the role means that new blocks will cease to be created. Figure 3 shows how the cycle is affected by the miner ceasing activity. When this occurs,

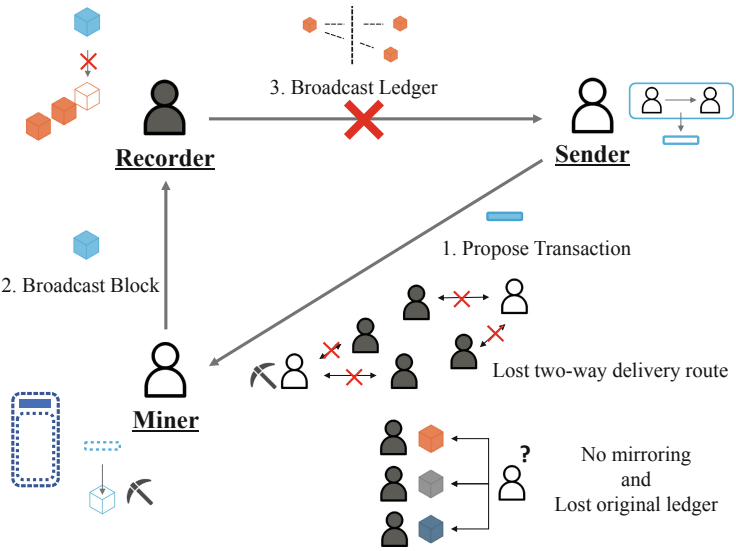

Fig. 4. Cycle of user role when the recorder's consent is in the blockchain.

time lags appear in the block interval, and the block updates eventually stop. The recorder is not able to chain the new blocks, and the block header stops updating. Shortly thereafter, the recorder will not be able to determine whether the block was stopped, or the PoW difficulty was delayed block creation. No matter how many transactions the sender broadcasts, they will not be verified. The recorder's resources are then overloaded, and all transactions are pooled. The miner's consent is easier to determine than the sender's consent.

4.3 Recorder's Consent

When only the recorders agree to closing the blockchain, it represents a final consent, unlike those of the sender and the miner. The recorder's role is to store all data to the initial block and update the ledger with new blocks validated by the miner and to maintain the route by which transactions and blocks are propagated. To lose the recorder role means that data cannot be recorded, and they are thus destroyed. Figure 4 shows the impact on the cycle that occurs when a recorder stops maintaining a distributed ledger. When recorder gives up, the sender can no longer determine the network status. The miner cannot catch new transactions and continues to generate empty blocks through self-mining. When this occurs, it is difficult to determine whether there is a network or software problem, compared with the sender's and the miner's consent. No longer able to validate the original block, the blockchain reaches its final closing state.

5 Empirical Analysis

In this section, we confirm the closing blockchain framework defined in the previous section against the actual state of a public blockchain to support an empirical understanding of the application research and to analyze the characteristics of each type of consent.

5.1 Data Collection

Our analysis covers 219 minor cryptocurrencies listed in cryptoID[5], including those listed in the middle of the period from June 1, 2019 to May 31, 2020. cryptoID provided the dataset for the blocks, transactions, and nodes on the blockchain network. We covered the entire observation period for blocks and transactions, but we only collected node information on the network from every 10 min and only for the most-recent month's data. The dataset about nodes in the network is minimal, owing to the limitations of the data source. Note that the theme of this analysis has potential limitations, unlike typical cryptocurrencies (e.g., Bitcoin and Ethereum). We targeted cryptocurrencies that cannot be observed stably. This is because of the need to continuously acquire the data source in a self-consistent manner; however, this data source may be lost in the near future.

5.2 Characteristics of Transactions and Blocks

Our framework proposed for finding unanimity for closing the blockchain focuses on the number of transactions included in a block and the new blocks. Note that checking the recorder's consensus is not measurable, because we need to observe that there were no nodes in the network, according to the framework. Thus, Fig. 5 shows a period of zero transactions and zero blocks in a day to confirm the status of the defined sender's and miner's consent. For the number of transactions, we did not include transactions that transferred generated mined coins to separate the cases where the number of transactions was zero but blocks were mined. The white area indicates a period where more than one transaction or block was identified in a day, and the black line indicates a period of zero transactions or blocks in a day. The periods with zero blocks matches the period with zero transactions.

As shown in the linear pattern, 124 currencies did not contain a single transaction during the day in the observation period. There were 46 cryptocurrencies for which no new blocks were created at least once during the day. Note that, even with Bitcoin, there were blocks that did not contain transactions, depending on the timing. However, they did not last all day. The decline in the number of transactions is a well-known cause of coin death in the community, but we can classify it into more detailed stages using the empirical results shown in Fig. 5. If we match the notation of the closing consensus with the notation of death, it

[5] https://chainz.cryptoid.info/.

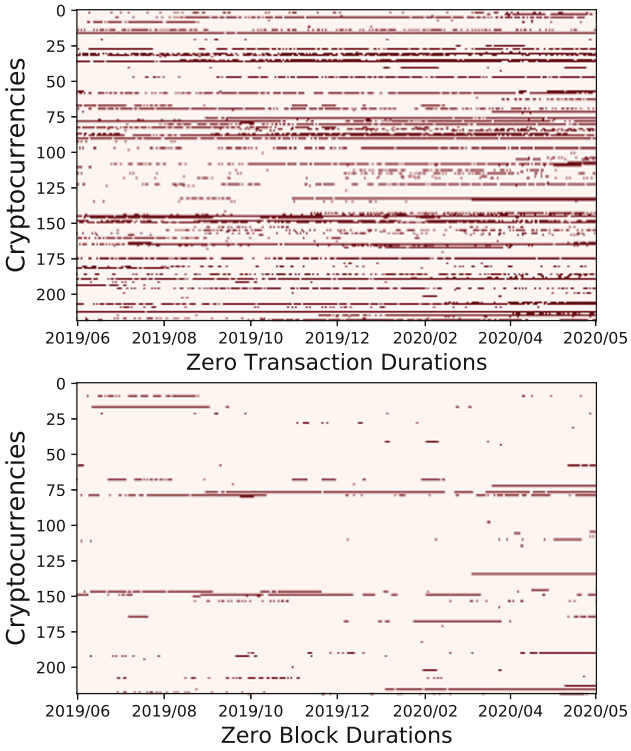

Fig. 5. Zero transactions and zero block duration in cryptocurrencies.

means that there is a stage of death in which there is not only a superficial death where the transaction is zero, but there is also an even deeper death where the block is zero. We arrive to this fact from the result that the pattern in the top figure showing the zero transactions does not match the pattern in the bottom figure showing the zero blocks.

Following the proposed framework, zero transactions depend on the actions of the sender, and zero blocks depend on the miner. As confirmed by the cycle flow, the creation of empty blocks has validity, reflecting that the sender had no intention of using the system. However, the miner intended to maintain the system to contribute to the cycle. Meanwhile, with the refusal to create a new block, there was an improvement in the investigation that required checking that transactions were sent throughout the network to determine which one the sender was on.

5.3 Case of Closing a Blockchain

To provide a concrete example, Fig. 6 shows an excerpt of the NPCcoin transactions and blocks included in zero transactions and zero blocks. The history of NPCcoin is an example of how, during the observation period, their genesis

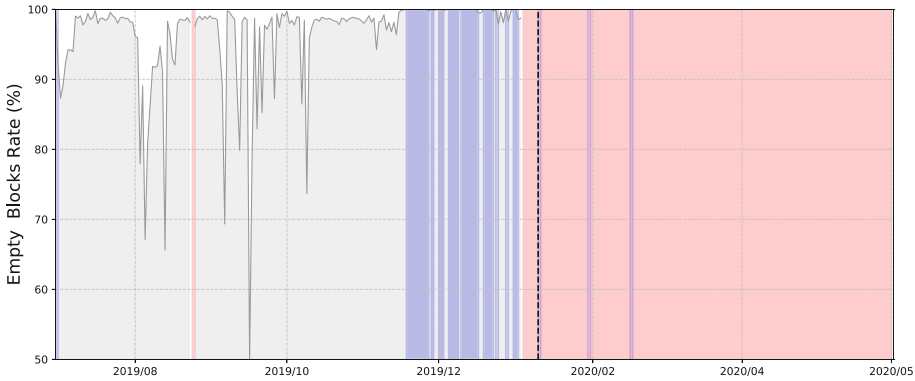

Fig. 6. Transaction and block lifecycle in NPCcoin. (Color figure online)

block of June 30, 2019, started. However, after about 6 months of operation, the developers announced that the community would shut down on January 10, 2020. The gray area in the figure shows the case where the empty block rate was less than 100% during the daytime. The blue area shows 100%, and the red area shows the period when no new blocks were created. The black dotted line represents January 10, 2020, the day the shutdown took place. The two vertical lines around February 2020 are blue. The blue area just before the shift to the red area starts on November 18, 2019. The most continuous period lasts 8 days, from December 10 through 17, 2019. Note that the blue area is not continuous.

We found two interesting things about closing consensus in Fig. 6. One is that the creation of a new block stops after an empty block. Of course, it is possible that this result is based on the announcement on the official site. However, there were no senders when mining resumed in February, 2020, after mining new blocks stopped. If the sender's broadcasted transaction was in the transaction memory pool, a new block that included transactions would be generated at this point. However, an empty block was generated. In other words, a sender could not be shown to exist; thus, the miner stopped mining. This is a good case of reproducing the cycle of propagation in which the sender first loses interest, then the miner loses interest. Additionally, the tendency for the number of transactions to gradually decrease, as the state just before the number of transactions reaches zero, is intuitively consistent with the situation where users gradually stop using the system when they lose interest. The other thing is that it is difficult to reach unanimity on a minor currency that is as close to closure as possible. As we find from the fact that mining has resumed, the ledger maintained an average of five servers until recently, regardless of the announcements. Thus, the cycle of unanimity remains consistent with the rotation of sender, miner, and recorder. Moreover, we focused on the timing of the block's delivery stoppage to clarify the potential signs of the dynamism that drove the closing blockchain.

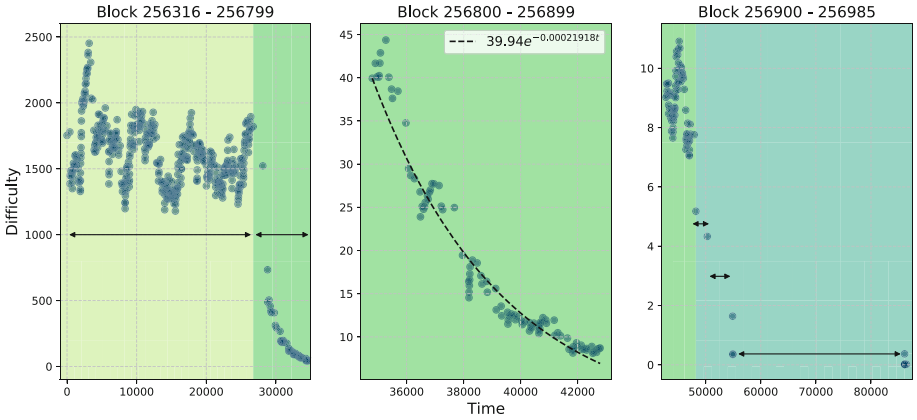

Fig. 7. Relationship trend between mining difficulty and mining interval just before the block mining stops.

Figure 7 shows the relationship between mining difficulty and mining interval from Block 256,985 on January 3, 2020, which is the day before block mining stopped. This confirms the trend. Each figure is empirically classified into three levels based on changes in the level of mining difficulty and interval. The left figure shows the discontinuous stage at which mining stopped after a period of continuous mining. The middle figure shows a stage in which the mining difficulty decreased in conjunction with the decrease in hash power caused by mining stoppage. The right figure shows the last stage at which a critical hash power was reached, at which point mining became difficult.

These results provide us with a useful empirical rule for isolating the early states. Thus, we can assume that following a model in which mining difficulty is stable and mean-regressive, a non-sequential change in mining difficulty that exponentially decreases can be an indication of miner dropout. The critical point is shown in the figure. We expected instability during stops in the framework. We confirmed that, in the last stage just before the mining stopped, the block interval was longer, and the block delivery speed as less stable.

We should note that our target blockchain implemented a block-by-block difficulty adjustment. However, we can assume that, even in the case of a blockchain, the difficulty adjustment occurs at constant intervals, similar to Bitcoin. The change in hash power is not immediately reflected as an exponential function, but it is similar to a change in the step function. From these results, we believe that there is validity in considering the proposed framework in the order of the cycle and for each role.

5.4 Characteristics of Nodes

As noted in the previous sub-section, the nodes that maintain the ledger are the root of the system, and the recorder is the final voter in blockchain closure. Here,

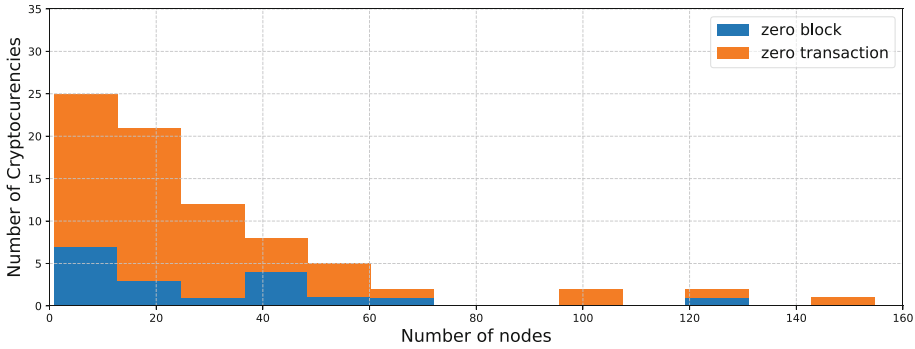

Fig. 8. The number of nodes in the cryptocurrency network with zero transactions and zero blocks.

we understand the characteristics of the unanimity by analyzing the distribution of the most recent orders. Figure 8 represents the distribution of the average number of nodes found on May 31st for the 80 cryptocurrencies that experienced an empty block or a new block suspended during the most recent period from May 1 to 31, 2020. The 80 cryptocurrencies were broken down into 60 empty blocks and 20 new blocks that were suspended.

First, for statistical information, we found 2,344 unique nodes, of which 2,174 (92.75%) were unduplicated, and the rest contained two or more duplicates. The largest duplicate nodes were duplicated in 35 cryptocurrencies. We noted that this is the server provided by the data-source site, which is used for this research. The regions in which the nodes were located were in 90 countries, with the most node-rich regions being, in order, the United States (530), Germany (469), Russia (103), The Netherlands (97), and Italy (92). Minor providers, including Hetzner, Choopa, OVH, and Contabo, were selected as node providers. These providers tended to be less expensive than major providers, such as Amazon Web Services, Microsoft Azure, or the Google Cloud Platform. This result is not significantly different from those reported in previous studies [15,18] that investigated the node distribution of networks in major cryptocurrencies.

Next, we found that more than half of the cryptocurrencies that created an empty block or stopped a new block maintained an average of fewer than 40 nodes. The distribution trends were the same in both cryptocurrencies, and they were maintained with fewer nodes. This number of nodes was less than 1% of those maintained by Bitcoin and Ethereum, each. This number approximates the final number of users. Thus, we controlled the availability of data with this unanimity to the number of people. However, we are certain that it was not easy to execute, because, even if one unit was running, it would be very difficult to close.

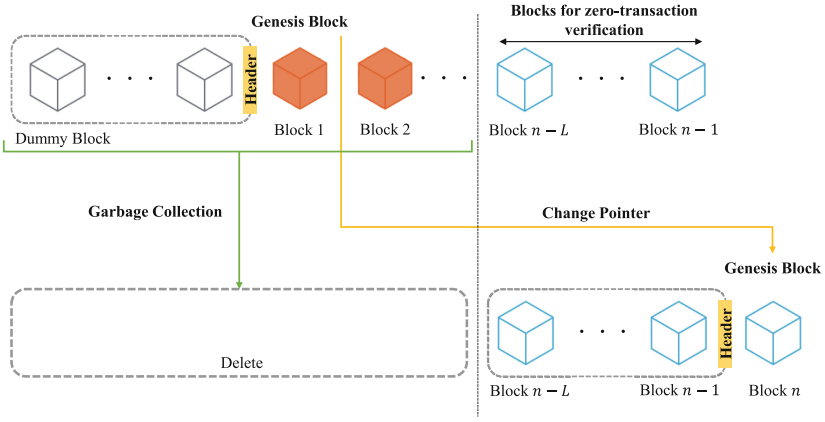

Fig. 9. Dynamically changing genesis block.

6 Discussion

From the framework and empirical analysis, we found that the number of transactions and blocks was significant for the closing unanimity. In this section, we discuss the method of embedding closure mechanisms into the blockchain based on this trend. In the design, full closure of the blockchain is needed to ensure that there are zero nodes storing the ledger. However, this situation means that we must verify that there are zero users in the network. Thus, we note that the mechanism that reduces availability is challenging and very difficult to implement. Therefore, we aim to achieve closing unanimity via the lack of new transactions and blocks, wherein the sender and the miner' agrees to close the blockchain by relaxing the conditions.

Focusing on the agreement of no new blocks, we find that this state is equivalent to the first time the blockchain is launched. In other words, we can create the same situation as closing through initialization, which overwrites the data previously stored in the ledger and erases the historical data. The idea of this mechanism is illustrated in Fig. 9, which demonstrates the change of a genesis block to a dynamic pointer.

As a necessary configuration, we set a threshold of block-length L, which describes a block with zero transactions when the blockchain is launched. The first time the block is judged, it is handled with L dummy blocks, and these correspond to the header to which the genesis block points. Note that the block headers (L blocks) are necessary to judge the changing pointer.

After building the block, if all L blocks are zero-transaction blocks at the creation of the nth block, the genesis block is changed to the n^{th} block, and the blocks before $n - L$ are deleted. This behavior is similar to unused allocated values in memory being removed via garbage collection.

The initialization of the ledger propagates to all existing recorders in the network by the n^{th} block. When a new recorder connects to the network, the

existing recorder mirrors the initialized ledger, so that the discarded blocks are never restored. The configuration of the basic blockchain is the same as the basic blockchain structure, excluding the design of the timing for block deletion. Thus, this design is equivalent to a blockchain having no block destruction by setting the value of L to 0.

Unlike the case of hard-coding the genesis block, using it as a dynamic pointer in the network intentionally limits its integrity and the design lifetime. Here, we consider the block length, L, using empirical values for a deeper discussion. Referring to the number of blocks from 8 days and the longest period of zero transactions in the empirical analysis of Fig. 6, the block length, L, is about 10,000, because the mining interval is 60 s. However, considering the verification of every block propagation, we assume that we need to set a shorter L. Of course, it is difficult to delete after data are deployed on the network. However, this idea is an effective approach to stop them from being propagated to new users. We also assume that, if an update occurs, we can consequently follow the same path as the original block shown in Fig. 4 of the recorder in the framework, where we cannot prove that any past block is the original.

7 Conclusion

In this research, to deepen the discussion on finding the unanimity required to close blockchains, we proposed a framework for mechanism analysis that focuses on the three user roles (i.e., sender, recorder, and miner) in the blockchain. We considered user behavior that did not contribute to the cycle of roles as "voting without interest" and visualized the impact of the flow of blocks and transactions in the system. Then, we confirmed the validity of the proposed framework on the minor cryptocurrencies using a blockchain that we assumed to be near closing.

We confirmed the empirical consistency of our proposed framework in which the flow of the block stopped when the flow of the transaction reached zero, using the target measured from the launch to the mining stop of the block during the observation period. We also reported on discontinuous connections of mining difficulty and changes in decay and mining intervals that made them unstable at the critical point at which the miner stopped a new block. We noted the difficulty of stopping and the high remaining block availability, because some nodes still maintained ledgers, even after excluding duplicate nodes, despite the zero flow of transactions and blocks.

Finally, based on the framework and empirical analysis, we discussed methods of implementing a mechanism for closing the blockchain. Because it is the closing that is the target of this paper, we emphasized the approach via initialization rather than modification or deletion.

In a future work, it will be necessary to clarify the operational issues using a pilot implementation program that applies a stopping mechanism. We assume that discussions are necessary for businesses to responsibly close-out decentralized data using the blockchains across companies in view of security risks and data protection. To achieve this, we must set limits on the integrity of data in the blockchain and implement a system lifetime.

Acknowledgement. This work was partly supported by Grant-in-Aid for Scientific Research (B) (19H04107).

References

1. Al-Jaroodi, J., Mohamed, N.: Blockchain in industries: a survey. IEEE Access **7**, 36500–36515 (2019)
2. Ali, S.T., McCorry, P., Lee, P.H.J., Hao, F.: Zombiecoin 2.0: managing next-generation botnets using bitcoin. Int. J. Inf. Secur. **17**(4), 411–422 (2018)
3. Apostolaki, M., Zohar, A., Vanbever, L.: Hijacking bitcoin: routing attacks on cryptocurrencies. In: IEEE Symposium on Security and Privacy (S&P), pp. 375–392. IEEE (2017)
4. Babaioff, M., Dobzinski, S., Oren, S., Zohar, A.: On bitcoin and red balloons. In: Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 56–73. ACM (2012)
5. Back, A., et al.: Hashcash-a denial of service counter-measure (2002). http://www.hashcash.org/papers/hashcash.pdf
6. Baden, M., Torres, C.F., Pontiveros, B.B.F., State, R.: Whispering botnet command and control instructions. In: 2019 Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 77–81. IEEE (2019)
7. Bartoletti, M., Cimoli, T., Pompianu, L., Serusi, S.: Blockchain for social good: a quantitative analysis. In: Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good, pp. 37–42. ACM (2018)
8. Cunico, H.A., Dunne, S., Harpur, L.S., Silva, A.: Blockchain lifecycle management. US Patent App. 15/848,036. 20 June 2019
9. DuPont, Q.: Experiments in algorithmic governance: a history and ethnography of "the DAO," a failed decentralized autonomous organization. Bitcoin and beyond, pp. 157–177 (2017)
10. Fantazzini, D., Zimin, S.: A multivariate approach for the simultaneous modelling of market risk and credit risk for cryptocurrencies. J. Ind. Bus. Econ. **47**(1), 19–69 (2020)
11. Gervais, A., Ritzdorf, H., Karame, G.O., Capkun, S.: Tampering with the delivery of blocks and transactions in bitcoin. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 692–705. ACM (2015)
12. Guo, L., Li, X.J.: Risk analysis of cryptocurrency as an alternative asset class. In: Härdle, W.K., Chen, C.Y.-H., Overbeck, L. (eds.) Applied Quantitative Finance. SC, pp. 309–329. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54486-0_16
13. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: 24th USENIX Security Symposium, pp. 129–144. USENIX Association (2015)
14. Imamura, M., Omote, K.: Difficulty of decentralized structure due to rational user behavior on blockchain. In: Liu, J.K., Huang, X. (eds.) NSS 2019. LNCS, vol. 11928, pp. 504–519. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36938-5_31
15. Kim, S.K., Ma, Z., Murali, S., Mason, J., Miller, A., Bailey, M.: Measuring Ethereum network peers. In: Proceedings of the Internet Measurement Conference 2018, pp. 91–104. ACM (2018)

16. Matzutt, R., et al.: A quantitative analysis of the impact of arbitrary blockchain content on bitcoin. In: Meiklejohn, S., Sako, K. (eds.) FC 2018. LNCS, vol. 10957, pp. 420–438. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-58387-6_23
17. Motlagh, S.G., Misic, J., Misic, V.B.: Modeling of churn process in bitcoin network. In: 2020 International Conference on Computing, Networking and Communications (ICNC), pp. 686–691. IEEE (2020)
18. Park, S., Im, S., Seol, Y., Paek, J.: Nodes in the bitcoin network: comparative measurement study and survey. IEEE Access **7**, 57009–57022 (2019)
19. Sato, T., Imamura, M., Omote, K.: Threat analysis of poisoning attack against ethereum blockchain. In: Laurent, M., Giannetsos, T. (eds.) WISTP 2019. LNCS, vol. 12024, pp. 139–154. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41702-4_9
20. Satoshi, N.: Bitcoin: a peer-to-peer electronic cash system (2008). http://www.bitcoin.org/bitcoin.pdf
21. Tran, M., Choi, I., Moon, G.J., Vu, A.V., Kang, M.S.: A stealthier partitioning attack against bitcoin peer-to-peer network. In: IEEE Symposium on Security and Privacy (S&P). IEEE (2020)
22. Voigt, P., Von dem Bussche, A.: The EU General Data Protection Regulation (GDPR). A Practical Guide, 1st edn, Springer International Publishing, Cham (2017)
23. Walck, M., Wang, K., Kim, H.S.: Tendrilstaller: block delay attack in bitcoin. In: 2019 IEEE International Conference on Blockchain (Blockchain), pp. 1–9. IEEE (2019)