A Theoretical Model for Fork Analysis in the Bitcoin Network

Conference Paper · July 2019

DOI: 10.1109/Blockchain.2019.00038

3 authors:

Yahya Shahsavari
École de Technologie Supérieure
4 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Kaiwen Zhang
École de Technologie Supérieure
75 PUBLICATIONS **504** CITATIONS
SEE PROFILE

Chamseddine Talhi
École de Technologie Supérieure
68 PUBLICATIONS **710** CITATIONS
SEE PROFILE

A Theoretical Model for Fork Analysis in the Bitcoin Network

Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi
Department of Software & IT engineering
École de Technologie Supérieure (ÉTS)
University of Quebec
Montreal, Quebec, Canada
Emails: yahya.shahsavari.1@ens.etsmtl.ca, {kaiwen.zhang, chamseddine.talhi}@etsmtl.ca

***Abstract*—Blockchain networks which employ Proof-of-Work in their consensus mechanism may face inconsistencies in the form of forks. These forks are usually resolved through the application of block selection rules (such as the Nakamoto consensus). In this paper, we investigate the cause and length of forks for the Bitcoin network. We develop theoretical formulas which model the Bitcoin consensus and network protocols, based on an Erdös-Rényi random graph construction of the overlay network of peers. Our theoretical model addresses the effect of key parameters on the fork occurrence probability, such as block propagation delay, network bandwidth, and block size. We also leverage this model to estimate the weight of fork branches. Our model is implemented using the network simulator OMNET++, and validated by historical Bitcoin data. We show that under current conditions, Bitcoin will not benefit from increasing the number of connections per node.**

***Index Terms*—Bitcoin, Blockchain, Blockchain fork, Theoretical modeling, Cryptocurrency, Peer-to-peer network.**

I. Introduction

A cryptocurrency network may face some inconsistencies that arise from its decentralized nature. For instance, Bitcoin uses a large and unstructured peer-to-peer (P2P) network, which is susceptible to fork-related inconsistencies when propagating blocks [1]. Forks can occur either due to propagation delay [2], [3] or poor connectivity leading to networking partitions [4], [5]. We differentiate between these type of *natural* forks (which are the focus of this paper), and *intentional* forks occurring from miners deviating from the protocol [6], [7], or changes in the code or protocol, which create a new independent network (which are outside the scope of this paper) [8].

Forks are undesirable since they create inconsistencies across the local copies of the ledger, which reduce the reliability of responses to queries about the blockchain data. Thus, the occurrence of forks implies that blockchain networks are eventually consistent and disturb the notion of immutability in blockchains. As a consequence, users have to wait for an amount of time (usually measured in block confirmations) to be reasonably certain that a transaction is finally committed to the blockchain [9]. For the current Bitcoin network, users typically wait for 6 confirmations [10], but each user is free to choose its own threshold. Forks caused by malicious manipulations such as Goldfinger attacks can destabilize a cryptocurrency with a significant loss of confidence and destroy its exchange rate [1]. As well, forks can be exploited by malicious entities such as dishonest miners to gain unfair profits by disturbing the normal operation of the system [7].

In light of the above, we can argue that it is imperative for blockchain designers and cryptocurrency analysts to have a comprehensive understanding of the causes and factors related to the formation of forks and have the ability to predict the occurrence of disruptive forks in order to apply countermeasures. To accomplish this, we propose a theoretical model for the analysis of forks in blockchain networks, particularly for Bitcoin. Our model can be useful to predict the long-term impact of proposed changes to Bitcoin, as well as assess the health of *altcoins* networks based on hard forks of the Bitcoin code.

In this paper, we extend our previous work presented in [4], which presents a theoretical model for performance modeling and analysis of the Bitcoin network using an Erdös-Rényi random graph model [11]. However, this previous work is concerned with the propagation of a single block at a time through different gossiping waves, using the Bitcoin inventory protocol, and assumes that no forks can occur. For the current paper, the main contributions are as follows:

1) We extend our wave-based model to consider the simultaneous propagation of concurrent blocks (forks).
2) We present equations to estimate the fork occurrence probability in the network, and the number of participating nodes in different fork branches (branch weight).
3) We implement our theoretical model using the network simulator OMNET++ and compare to historical Bitcoin data.
4) We provide a sensitivity analysis of various blockchain network parameters for forks, such as the number of connections per node and block propagation delay.

The paper continues with Section II which reviews the related literature. Section III gives a brief overview of the Bitcoin network and a background on forks. In Section IV, we present our analytical model to capture the dynamics of forks in the Bitcoin network. Section V is dedicated to the theoretical and simulation results of our model. Finally, Section VI concludes this paper.

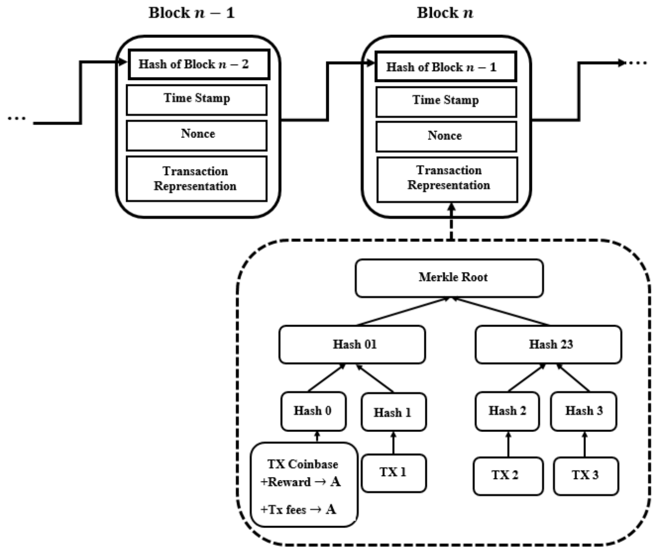


Fig. 1: Bitcoin blockchain data structure

II. Related Works

A. Analysis of forks in Bitcoin

There exist many works in the literature which address different kinds of forks in Bitcoin. But to the best of our knowledge, there is no comprehensive theoretical model for analyzing forks based on their input parameters and the network configurations. In addition, there exist several works which analyses forks using empirical data.

[3] analyses Bitcoin from a networking perspective. In this work, it is shown that the main cause of forks is the block propagation delay. According to the model presented in this network, the probability of forks is 1.78%, and an observed probability of 1.69% using their own empirical data. Our paper provides more detailed equations, which consider block size and network conditions, to accurately calculate fork probability.

[12] presents a taxonomy of blockchain forks. Furthermore, this work studies the generation of forks based on information delays and mining software upgrades. These two factors are orthogonal to our work.

In [13], an equation for estimating the fork occurrence probability based on 90% block propagation delay is presented. However, this work does not provide any theoretical proof or calculation approach for the mentioned equation.

[14] presents an empirical study on the propagation of blocks that led to forks in Bitcoin. According to this work, the probability that a certain block becomes a part of the main chain increases almost linearly with the time advantage over competing blocks. In addition, the observed frequency of two consecutive blocks mined by the same miner is significant, possibly attributed to selfish mining [15].

[16] presents a comparison between Proof-of-Work (PoW), Proof-of-Stake (PoS), and other hybrid forms. According to this work, forks are more probable in PoW-based blockchains while in PoS-based systems, the chance is minimal. This work is orthogonal to our own as we focus on PoW-based systems, which are most susceptible to forks.

Although the Bitcoin white paper [17] suggests modeling the block arrivals as a homogeneous Poisson process, authors in [18] have claimed that the block arrival times in the Bitcoin network follow instead a non-homogeneous Poisson process. Nevertheless, in our work, we build our model based on a homogeneous Poisson process to simplify our equations since it still provides a good approximation over a long period of time.

In summary, none of the works above provide a comprehensive and well detailed theoretical model for the analysis of forks in the Bitcoin network, which is the main contribution of our own paper.

B. Overlay P2P network model

Bitcoin runs on a P2P overlay network, which is well-studied in literature from a theoretical perspective. In [19], the minimum achievable file distribution time is studied based on the network parameters such as the number of servers, number of receiving nodes, file size, and the upload and download capacities of the P2P nodes. The minimal time to fully disseminate a file of M parts from a server to N end users for a centralized architecture is presented in [20].

These two works are related to our paper since block dissemination in blockchain networks is similar to file dissemination in P2P networks. However, none of the aforementioned works present a theoretical model can calculate the average file dissemination delay in a decentralized network.

In [21], probabilistic flooding (randomly choosing next neighbor node) protocol for a P2P network when the underlying network is a random graph is presented. This is relevant to our work since we use a random graph for modeling the P2P blockchain network. However, we present a novel approach for estimating the forwarding probability in a random graph.

III. Background on Bitcoin

In this section, we briefly describe some basic concepts related to information dissemination in the Bitcoin network which is necessary to understand this paper. We also briefly introduce the concept of forks in Bitcoin.

Although the scope of this paper and the background section is focused exclusively on Bitcoin, the theoretical model presented for analysis of the forks in the Bitcoin network can be adapted to other blockchains such as Ethereum [22].

There exist two types of information dissemination for Bitcoin: *(i) transaction dissemination* and (ii) *block dissemination*. In the following sections, we will show that the latter may introduce forks in the network.

A. Blocks and transactions

An overview of the Bitcoin blockchain data structure is depicted in Figure 1. To be stored in the ledger, every individual transaction in Bitcoin should be embedded in a data structure called a block. A subset of participating nodes in Bitcoin network, namely miners, gather the propagated transactions

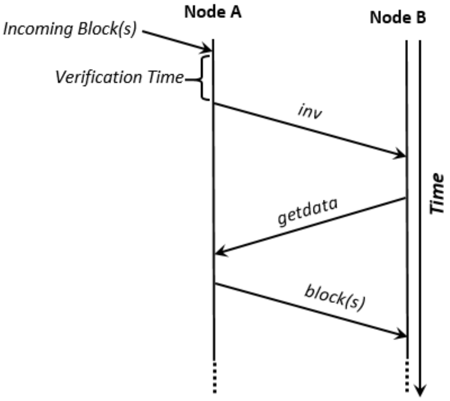


Fig. 2: Inventory-based block exchange

and group them in a proposal block. To be eligible to propagate this block over the network in order to add it to the blockchain as the next block, each miner has to calculate a *Proof-of-Work (PoW)*. To accomplish this, miners have to find a nonce value in an exhaustive and random process. The hash of the nonce combined with the hash of the previous block and the Merkle root [23] of the tree containing the transactions proposed by the miner, should be below a certain target threshold. This threshold is being used to adjust the mining difficulty.

For each Bitcoin transaction, the sum of the inputs should be equal or greater than the sum of the outputs. However, the *coinbase* transaction, included once in each block, has no inputs from previous blocks and grants a reward to the block creator (block reward and transaction fees). Since each miner will want to include its own coinbase transaction, each miner is thus mining on their own unique proposed block.

B. Overlay network protocol

Bitcoin uses a P2P overlay network operating over the Internet. It is a permissionless blockchain, which means that any node can freely join the network at any time. The term *node* refers to a physical device that acts as a logical entity and can have one or some of the following functions:

Routing: This functionality is responsible for message propagation and maintaining connections to other participating peers in the network. since the network is unstructured, each node can potentially communicate to any other node in the system. The exception is nodes running inside a cooperative mining pool. In this case, the pool server is responsible for routing duties.

Full blockchain storage: The node locally stores a complete and up-to-date version of the blockchain data. Full nodes do not need an external reference for verifying transactions or blocks and perform this duty autonomously.

Lightweight blockchain storage: This type of node only keep a lightweight version of the blockchain. Lightweight nodes verify transactions using a method called Simple Payment Verification (SPV) [24].

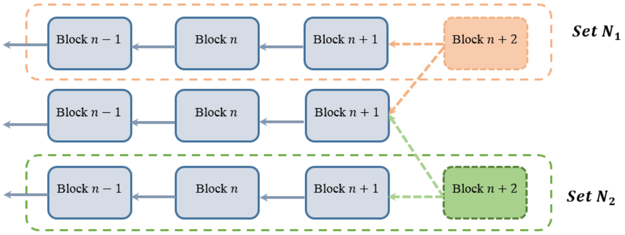


Fig. 3: Fork with two competing branches N_1 and N_2

Mining (Consensus participation): Miners participate in consensus to decide the next block to be added to the Bitcoin blockchain. Miners are typically equipped with specialized hardware in a competition to find the next PoW. The winner collects the coinbase transaction of the created block.

Wallet management: Typically, wallet nodes are transaction issuer nodes and are commonly responsible for transferring digital coins from the sender to the receiver. Most Wallet nodes, particularly those running on resource-constrained devices, are lightweight SPV nodes.

C. Block dissemination protocol

In order to disseminate information over the network, Bitcoin uses a gossiping protocol [25]. This protocol is an inventory-based protocol used to avoid saturating the network with redundant copies of transactions and blocks. According to this protocol, a node notifies the neighboring nodes about the availability of known blocks and transactions. To accomplish this, the node sends an inventory (*inv*) messages to all nodes in its connection pool. When a node receives an inv message for a block or transaction that it does not possess, the node replies with a $getdata$ message. The sender node will respond to the message with the requested block or transaction. This process is depicted in Figure 2.

D. Forks

Bitcoin is a blockchain system operating over a vast P2P network around the globe. In such a large-scale network, inconsistencies in the blockchain data may occasionally occur. The term *fork* refers to a situation in which different nodes have a different perspective of the blockchain. We can represent a fork as a directed tree of blocks. This concept is depicted in Figure 3 where two different set of nodes have a different state for the blockchain. In this case, the directed tree contains two separate branches. Generally, a fork can occur in one of the following cases:

Network isolation: Due to a poor connection between different nodes in the network, the network may become temporarily partitioned. This can occur if there are some very weak links acting as bottlenecks, or if there are not enough network connections in the system. As discussed in our previous work [4], for a network with N participating nodes, to prevent this situation with high probability, it is sufficient:

$$M \geq \lceil \frac{N-1}{N} log(N) \rceil \tag{1}$$

Where M is the average number of connections between participating peers. This kind of fork will prolong until the network becomes connected again.

Changes in core components of the blockchain protocol: Any change in core components of the protocol such as the format of a valid block or transaction, difficulty retargeting function, or any upgrade in the mining software can cause a fork in the system. Changes that are incompatible with previous versions cause a hard fork, as opposed to soft forks.

Miners deviation from the standard protocol: The most well-known fork caused by deviating miners is the double-spending attack. Other examples for this kind of attack are: temporary block withholding, selfish mining, feather forking attacks [1], etc.

Block propagation delay: In practice, two or more different miners may find a valid block almost at the same time. If the first miner has not fully disseminated its block due to propagation delay, a fork occurs if a second miner starts gossiping its own proposed block. Increasing the block size to include more transactions has the drawback of raising the propagation delay, which increases the probability of fork occurrence. Therefore, our paper considers the trade-off between the block size and fork occurrence probability.

According to the Nakamoto consensus, this type of inconsistency is usually solved at the next block if miners select the longest chain [17].

IV. Analytical Model

In this section, we first define the notion of *natural* fork caused by propagation delay in a blockchain network. Then, we propose a random graph model for theoretical modeling and analysis of block propagation in Bitcoin blockchain when a fork occurs. To accomplish this, we use a graph model which was introduced by Erdös and Rényi. This graph has properties suitable for modeling peer-to-peer overlay networks used by blockchain systems.

A. Fork model

Suppose a block b is the tip of a blockchain $\mathcal{B}$. We call the height h_b of block b as the number of blocks preceding this block in the blockchain starting from the genesis block. In other words, the height of a block is the length of the blockchain to reach it. For the genesis block, h_g is 0. As previously mentioned, the blockchain for Bitcoin is replicated to the full nodes connected together in a very wide P2P network. A fork occurs when there are at least two blocks b and b' that have the same height. Precisely, a fork caused by propagation delay exists when:

$$\exists \ b, b' \in \mathcal{B} \ and \ b \neq b' \mid h_b = h_{b'} \tag{2}$$

Assume a node with a blockchain of height of h_b receives block b' with height of $h_{b'}$ where $h_{b'} > h_b$. Because of the longest chain selection rule, it will switch its blockchain tip to block b'. Blocks b and b' can be either in the same branch (then block b is the ancestor of block b') or in different branches. In the prior case, the node will retrieve the intermediate blocks from the network and in the latter case, block b will become orphaned (removed from the main branch). After this process, the local state of the blockchain maintained by the node should be consistent with that of the rest of the network, assuming this is the longest chain known.

B. Fork probability analysis

PoW mining for the Bitcoin network can be modeled as a Poisson process and the inter-block time (the time difference between two consecutive mined blocks) follows an exponential distribution. We calculate that the probability density function (PDF) of a block to be mined as follows:

$$f(t; \lambda) = \lambda e^{-\lambda t} \tag{3}$$

where λ is obtained from the following equation:

$$\lambda = \frac{1}{E[T]} = \frac{1}{t_B} \tag{4}$$

where t_B is referred to as the inter-block time and is the average time required to mine a new block. This process is *memoryless* (i.e., the probability distribution is independent of its history) and the probability that another block is mined before the currently proposed block is fully disseminated over the network can be obtained as follows:

$$F(t) = P(T \leq t_{prop}) = \int_0^{t_{prop}} f(t) \ dt = 1 - e^{-\frac{t_{prop}}{t_B}} \tag{5}$$

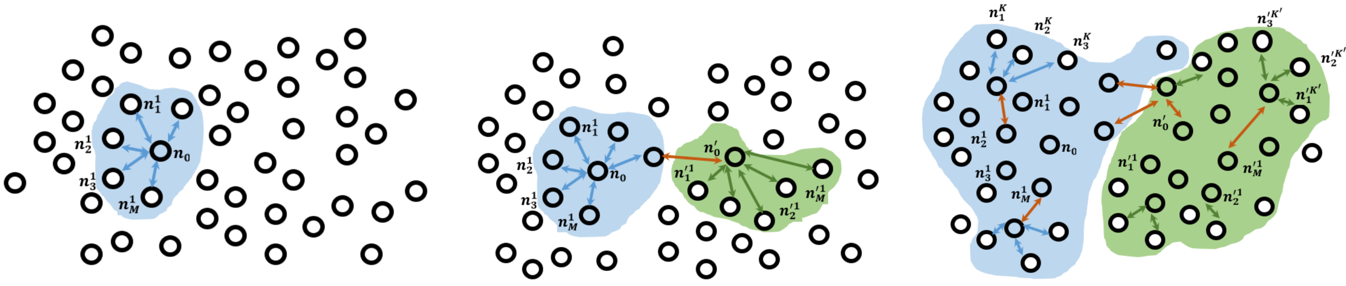
where t_{prop} is the time needed for a block to fully propagate over the network. Equation (5) expresses the probability of forks as a function of block propagation time and inter-block time. However, block propagation time itself is derived using several parameters, e.g., block size, bandwidth, the average number of connections per node, the total number of participating nodes, which are already discussed in our previous paper [4].

C. Block propagation model

To model the P2P overlay network, we use an Erdös-Rényi graph $G(V, L)$ where V and L are the set of vertices and the set of links between participating nodes respectively. For example, if there is a link between node i and node j, then $(i, j) \in L$.

Moreover, we represent our random graph using $G_p(N)$, where N is the total number of participating nodes and p is the independent probability that there exists a link between any two arbitrary selected nodes in the P2P overlay network. In this paper, we assume that N is significantly large (e.g., $N \approx 10000$ in the current Bitcoin network [26].

Consider a P2P overlay network which consists of N participating nodes: suppose a node mines a block b with height h_b at time t. We refer to this initial node as n_0.



(a) Wave W_1

(b) Wave W_1'

(c) Waves W_K and W_k'

Fig. 4: Different waves of block dissemination in the proposed blockchain network. A fork occurs when node n_0' starts to propagate its block (b') while the block from n_0 (b) is not fully disseminated. Blue nodes are the nodes which have already received the block b and green nodes are the nodes which have already received the block b'. Accordingly, blue and green arrows indicate successful block transfer. Red arrows show timed-out inv messages.

According to Section III-C, node n_0 sends an inv message to the set $\mathcal{W}_1 = \{n_1^1, n_2^1, ..., n_M^1\}$ of neighboring nodes in its connection pool. For simplicity, we assume all nodes are connected to M nodes in their connection pool. We assume the sending node sends the inv message to its neighbors in succession with a very small negligible delay ϵ between each message.

Since this is the first time that an inv message is being sent for this block, none of the $\mathcal{W}_1$ neighboring nodes have the block and will respond the inv message with the $getdata$ message with 100% certainty. Upon receiving the $getdata$ response, n_0 will send the complete block to the requesting neighbors. We call this first step of the block dissemination process wave W_1 (depicted in Figure 4a). The time taken from sending the inv message until receiving the block by the neighboring nodes is called the *wave length* and denoted by T.

For this wave W_1, we define a random variable called the *forwarding probability* p_{f_1}. The forwarding probability is the probability that an inv message, containing tne information about newly mined block, will be accepted the neighboring nodes contacted by a sender node. For wave 1, it is clear that $p_{f_1} = 1$, as the sender node is the block creator n_0.

Suppose another miner node in the network which is not a member of $\mathcal{W}_1$, finds block b' with height $h_{b'}$ at time t', where $t < t' < t + T$ and $h_b = h_{b'}$. We denote this second initiator node as n_0'. Similarly to n_0, node n_0' sends an inv message to the set $\mathcal{W}_1' = \{n_1'^1, n_2'^1, ..., n_M'^1\}$ of neighboring nodes in its connection pool. Again, we assume the sending node sends the inv message to its neighbors in succession with a very small negligible delay ϵ' between each message. We call this process wave W_1' (depicted in Figure 4b).

However, some of the nodes may have already received a block with a height of h_b in wave W_1 and thus will not accept the block propagated in the wave W_1'. Hence, we calculate the forwarding probability for the wave W_1' as follows:

$$p_{f_1'} = \frac{N-1-|\mathcal{W}_1|}{N-1} = \frac{N-1-Mp_{f_1}}{N-1} \tag{6}$$

Subsequently, the nodes which have received block b will send inv messages for that block to the neighbors in their connection pool. We call this process wave W_2. Some of the receiving nodes may not accept the block, if they have already received blocks b or b' and will not send a $getdata$ message. Hence, forwarding probability for wave W_2 can be calculated as follows:

$$p_{f_2} = \frac{N-1-|\mathcal{W}_1|-|\mathcal{W}_1'|}{N-1} = \frac{N-1-Mp_{f_1}-Mp_{f_1'}}{N-1} \tag{7}$$

The set of nodes that will reply positively with a $getdata$ message in wave W_2 can be expressed as: $\mathcal{W}_2 = \{n_1^2, n_2^2, ..., n_{|\mathcal{W}_2|}^2\}$ where $|\mathcal{W}_2| = \lceil p_{f_1} p_{f_2} M^2 \rceil$.

The forwarding probability for wave W_2 is:

$$p_{f_2'} = \frac{N-1-|\mathcal{W}_1|-|\mathcal{W}_1'|-|\mathcal{W}_2|}{N-1} = \frac{N-1-Mp_{f_1}-Mp_{f_1'}-p_{f_1}p_{f_2}M^2}{N-1} \tag{8}$$

Each subsequent wave of both blocks will follow a similar pattern as above. In general, we express the forwarding probability of wave W_i as follows:

$$p_{f_i} = \frac{(N-1) - \sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f_k'}}{N-1} \quad (1 < i \leq K) \tag{9}$$

where K is the total number of waves needed for the block b to be propagated over the blockchain network. Note that some nodes in the network may never accept the block (if it has received block b' first). Concerning block b', the forwarding probability of wave W_i' can be calculated as follows:

$$p_{f'_i} = \frac{(N-1) - \sum_{j=1}^{i} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f'_k}}{N-1} \qquad (1 < i \leq K') \tag{10}$$

where K' is the total number of waves needed for block b' to be propagated over the blockchain network.

For values of t other than $t < t' < t+T$, Equations (9) and (10) can generalized for $t + mT < t' < t + (m+1)T$ where $m = 0, 1, 2, ...$:

$$p_{f_i} = \frac{(N-1) - \sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-1-m} M^j \prod_{k=1}^{j} p_{f'_k}}{N-1} \qquad (1 < i \leq K) \tag{11}$$

and

$$p_{f'_i} = \frac{(N-1) - \sum_{j=1}^{i} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-m-1} M^j \prod_{k=1}^{j} p_{f'_k}}{N-1} \qquad (1 < i - m \leq K') \tag{12}$$

Equations (11) and (12) are recursive equations and enables us to calculate the number of waves required for all participating nodes to receive *either one* of the blocks b or b':

$$\sum_{i=1}^{K} M^i \prod_{j=1}^{i-1} p_{f_j} + \sum_{i=1}^{K'} M^i \prod_{j=1}^{i-1} p_{f'_j} = N - 1 \tag{13}$$

Figure 4c illustrates the process of block dissemination during waves W_k and W'_k. To calculate K and K', we follow an iterative approach using Algorithm 1. In the first iteration, $p_{f_1} = 1$. Then, we calculate $p_{f'_1}$. We substitute them into Equation (13). If the equality is satisfied, the algorithm stops and return K and K'. Otherwise, we start the second iteration and calculate p_{f_2} using p_{f_1} and $p_{f'_1}$. Then again substitute in the Equation (13). If it is now satisfied, we stop and take the K and K'. Otherwise, we continue to find $p_{f'_2}$. This process continues until it finds the appropriate values of K and K'.

The first term in Equation (13) indicates the total number of nodes which accept block b as the blockchain head and second term indicates the nodes which accept the block b'. We call these terms the *weight* of each branch. The branch weight allows determining if the fork is carry over beyond b and b'. If there is a skew towards one branch, the total hash rate working behind that branch is likely to be greater than the other branch, resulting in a greater difference in block mining time, which increases the probability that the next block of the faster branch will fully propagate and orphan the slower branch. Conversely, if both branches have similar weight, the expected block time will be similar and the likelihood of competing blocks being propagated simultaneously increases.

Algorithm 1: Calculating number of waves K and K'

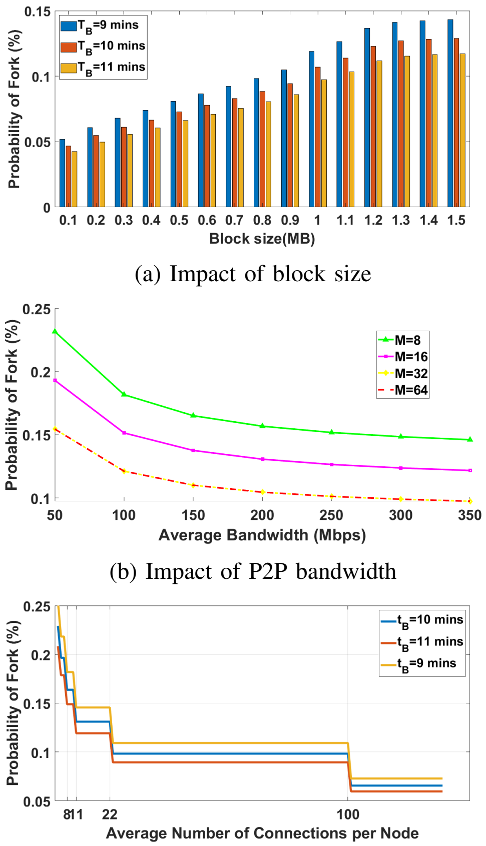
```
Input  : The values of $N$, $m$, $M$
Output: $K$, $K'$, Matrix $\mathcal{P}[p_{f1}...p_{f_K}]$, and $\mathcal{P}'[p_{f'1}...p_{f'_K}]$
$\mathcal{P}[1] \leftarrow 0$
$\mathcal{P}'[1] \leftarrow 0$
$K \leftarrow 1$
$K' \leftarrow 0$
$C \leftarrow 0$    // Controller
while $C$=0 do
    for $i = 1$ to $K$ do
        if $i - m > 0$ then
            $\mathcal{P}[i]$=$\frac{(N-1)-\sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-1-m} M^j \prod_{k=1}^{j} p_{f'_k}}{N-1}$
            if $\mathcal{P}$ [i] $\geq 0$ then
                $K = K + 1$
                $\mathcal{P}'[i-m] = \frac{(N-1)-\sum_{j=1}^{i} M^j \prod_{k=1}^{j} p_{f_k} - \sum_{j=1}^{i-1-m} M^j \prod_{k=1}^{j} p_{f'_k}}{N-1}$
                if $\mathcal{P}'$ [i - m] $\geq 0$ then
                    $K' = K' + 1$
                else
                    $C \leftarrow 1$
                end
            else
                $C \leftarrow 1$
            end
        else
            $\mathcal{P}[i]$=$\frac{(N-1)-\sum_{j=1}^{i-1} M^j \prod_{k=1}^{j} p_{f_k}}{N-1}$
            if $\mathcal{P}$ [i] $\geq 0$ then
                $K = K + 1$
            else
                $C \leftarrow 1$
            end
        end
    end
end
return $K$, $K'$, $\mathcal{P}$, and $\mathcal{P}'$
```

V. Results and Analysis

In this section, we first present the results of fork probability analysis in the Bitcoin network based on the following parameters: the block size, the average P2P bandwidth, and the average number of connections per node (see Figure 5). Then, we present the results of a fork branch weight analysis based on the difference in the time at which the two blocks are mined (see Figure 6).

A. Settings and implementation

Parameters: To assess the impact of block size on the fork occurrence probability, we use the empirical data provided by [26]. We extract ∼1500 Bitcoin blocks (blocks 504016 through 522429) with reported propagation delay values and block sizes. We consider median values for block size intervals of 100 kB. For all other experiments in this paper, we consider

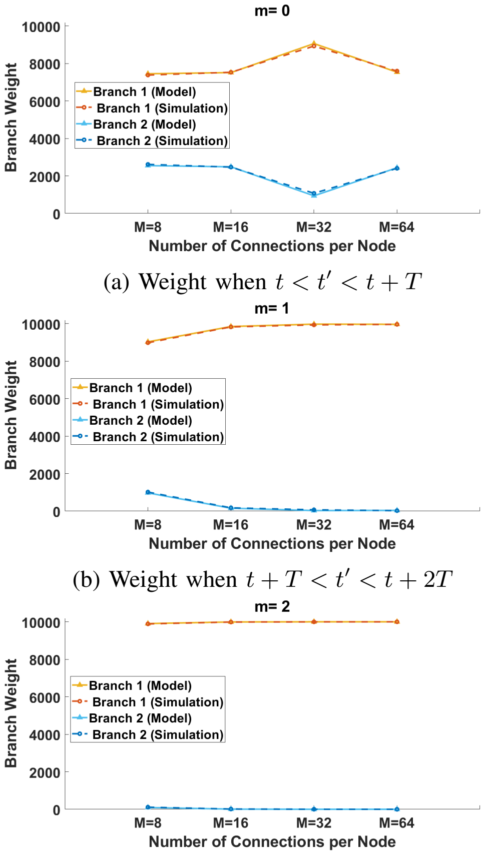


(a) Impact of block size

(b) Impact of P2P bandwidth

(c) Impact of the average number of connections

Fig. 5: Fork probability analysis based on network parameters



(a) Weight when $t < t' < t + T$

(b) Weight when $t + T < t' < t + 2T$

(c) Weight when $t + 2T < t' < t + 3T$

Fig. 6: Branch weight analysis based on t'

the average block size as 1MB. For the network settings, we set all parameters the same as the settings in our previous paper [4].

Implementation: We implement our model using a discrete event-based simulator called OMNET++ [27]. Simulation results are calculated using the event log files generated by the simulation. We also used $Matlab$ for theoretical analysis.

B. Results

Fork probability experiments: To study the impact of different parameters on the probability of fork occurrence in the Bitcoin blockchain, we conduct three experiments. In the first experiment, we estimate the fork probability in current Bitcoin blockchain as a function of the block size. We carry out this experiment for three different values of inter-block time ($t_B = 9, 10, 11\ mins$) where $t_B = 10\ mins$ is the average inter-block time in Bitcoin and $t_B = 9, 11\ mins$ are 1 minute deviations from the current average. Results of this experiment are depicted in Figure 5a. The fork occurrence probability increases proportionally to the block size. Also note that decreasing the block time (t_B) increases the fork probability and vice versa. Therefore, the probability of fork can be manipulated by adjusting the mining difficulty.

To validate the above results, we extract the historical data of 99,120 Bitcoin blocks (blocks 90392 through 189512). During that period, we observe that 90 forks happened, for a reported rate of fork occurrence of 0.09% in the Bitcoin network. Our model results of 0.1% at the block size of 1MB are therefore very close to the empirical result.

In another set of experiments, we study the joint impact of the bandwidth and the average number of connections per node M. As depicted in Figure 5b, the probability of fork decreases when the bandwidth or the number of connections increases. For $M = 32$ and $M = 64$, since the block propagation delay is the same for both (shown in [4]), the curves are almost identical. This corroborates the fact that forks are caused by network delay.

Figure 5c shows a detailed sensitivity analysis for the average number of connections per node M. There is no significant difference in the probability of fork occurrence when $22 \leq M \leq 99$, which is a significant margin. Since M is around 32 for the real network, we can therefore claim that Bitcoin is not sensitive to the value of M, currently centered around 32. This experiment also confirms that a lower value of t_B increases the chance of having a fork.

Branch weight experiments: We conduct another set of experiments based on the time instance at which the competitor node begins to propagate its block while another node has already started propagating its proposed block. For this set of experiments, we study the sensitivity of the network with respect to the average number of connections M.

In the first experiment, we set $m = 0$ (see Equation (12)). This means that the second miner finds and starts to propagate its block (we call it $block$ 2) during the first wave of dissemination for $block$ 1. Figure 6a shows the results of this experiment. The simulation results closely match our theoretical results. Generally, increasing M yields a heavier weight for the first branch. In particular, note the special case of $M = 32$ where the difference between the two branches is maximal. This is justified by the fact that the Bitcoin network experiences minimal traffic overhead when $M = 32$ (as shown in our previous paper [4]): the first block is disseminated much more efficiently during each wave.

In the next experiments, we set $m = 1$ and $m = 2$, which means the second miner starts propagation its block during second and third waves of dissemination for $block$ 1, respectively. The results of these experiments are shown in Figures 6b and 6c. Again, the simulation results and the theoretical results are almost the same. Increasing the M or m will increase the weight of the first branch. When both M and m have high values, the weight of the second branch is near zero and we can thus claim that the impact of this fork is almost negligible on the consistency of the Bitcoin network: $block$ 2 will quickly be orphaned.

VI. Conclusion

In this paper, we present an analytical model to estimate the probability of fork occurrence in the Bitcoin network, using a random graph to model the Bitcoin overlay network and dissemination waves to model the inventory-based block propagation protocol. We investigate the effect of several blockchain and network parameters on the probability of forks. Our results show that reducing the block time compromises the security of the blockchain by increasing the probability of fork. The average number of connections per node currently has no impact on the probability of forks, since Bitcoin currently operates within a stable range of 22-99 connections. In addition, we investigate the impact of the time difference between two concurrent blocks and the average number of connections per node on the weight of fork branches. If the later miner starts to propagate its block too late and the number of connections per node is sufficiently high, the impact of the fork on the network is almost negligible.

References

[1] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton University Press, 2016.

[2] A. M. Antonopoulos, *Mastering Bitcoin: Programming the open blockchain*. " O'Reilly Media, Inc.", 2017.

[4] Y. Shahsavari, K. Zhang, and C. Talhi, "Performance modeling and analysis of the bitcoin inventory protocol," in *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, USA, 2019.

[3] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*. IEEE, 2013, pp. 1–10.

[5] Z. Yao, X. Wang, D. Leonard, and D. Loguinov, "Node isolation model and age-based neighbor selection in unstructured p2p networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 1, pp. 144–157, 2009.

[6] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim, "Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 195–209.

[7] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[8] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[9] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 104–121.

[10] "Bitcoin wiki: Confirmation," https://en.bitcoin.it/wiki/Confirmation, accessed: 2019-04-11.

[11] P. Erdős and A. Rényi, "On random graphs i." *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959 1959.

[12] B. Biais, C. Bisiere, M. Bouvard, and C. Casamatta, "The blockchain folk theorem," *Swiss Finance Institute Research Paper*, no. 17-75, 2018.

[13] U. Klarman, S. Basu, A. Kuzmanovic, and E. G. Sirer, *bloXroute: A Scalable Trustless Blockchain Distribution Network WHITEPAPER*, 2018 (accessed March 04, 2019). [Online]. Available: https://bloxroute.com/wp-content/uploads/2018/03/bloXroute-whitepaper.pdf

[14] T. Neudecker and H. Hartenstein, *"Short Paper: An Empirical Analysis of Blockchain Forks in Bitcoin"*, 2019 (accessed March 04, 2019). [Online]. Available: https://dsn.tm.kit.edu/bitcoin/forks

[15] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong, "A deep dive into blockchain selfish mining," *arXiv preprint arXiv:1811.08263*, 2018.

[16] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain." *Journal of Information processing systems*, vol. 14, no. 1, 2018.

[17] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[18] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Block arrivals in the bitcoin blockchain," *CoRR*, vol. abs/1801.07447, 2018.

[19] R. Kumar and K. W. Ross, "Optimal peer-assisted file distribution: Single and multi-class problems," in *Proceedings of IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB)*. IEEE, 2006.

[20] J. Mundinger, R. Weber, and G. Weiss, "Analysis of peer-to-peer file dissemination," *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 3, pp. 12–14, 2006.

[21] K. Oikonomou and I. Stavrakakis, "Performance analysis of probabilistic flooding using random graphs," in *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2007, pp. 1–6.

[22] V. Buterin *et al.*, "Ethereum white paper, 2014," *URL https://github.com/ethereum/wiki/wiki/White-Paper*, 2013.

[23] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Conference on the theory and application of cryptographic techniques*. Springer, 1987, pp. 369–378.

[24] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," *Available: https://blockstream.com/sidechains.pdf*, 2014.

[25] M.-J. Lin and K. Marzullo, "Directional gossip: Gossip in a wide area network," in *European Dependable Computing Conference (EDCC)*. Springer, 1999, pp. 364–379.

[26] "Bitnodes API v1.0," https://bitnodes.earn.com, accessed: 2018-011-1.

[27] "Omnet++ simulator," *available at http://www. omnetpp. org*.