Stochastic Models and Wide-Area Network Measurements for Blockchain Design and Analysis

Nikolaos Papadis†, Sem Borst⋆, Anwar Walid⋆, Mohamed Grissa⋆, Leandros Tassiulas†

⋆Nokia Bell Labs, 600 Mountain Avenue, Murray Hill, NJ 07974, USA
†Department of Electrical Engineering, and Yale Institute for Network Science,
Yale University, New Haven, CT 06520, USA

***Abstract*—The Blockchain paradigm provides a popular mechanism for establishing trust and consensus in distributed environments. While Blockchain technology is currently primarily deployed in crypto-currency systems like Bitcoin, the concept is also expected to emerge as a key component of the Internet-of-Things (IoT), enabling novel applications in digital health, smart energy, asset tracking and smart transportation.**

As Blockchain networks evolve to industrial deployments with large numbers of geographically distributed nodes, the block transfer and processing delays arise as a critical issue which may create greater potential for forks and vulnerability to adversarial attacks. Motivated by these issues, we develop stochastic network models to capture the Blockchain evolution and dynamics and analyze the impact of the block dissemination delay and hashing power of the member nodes on Blockchain performance in terms of the overall block generation rate and required computational power for launching a successful attack.

The results provide useful insight in crucial design issues, e.g., how to adjust the 'difficulty-of-work' in the presence of delay so as to achieve a target block generation rate or appropriate level of immunity from adversarial attacks. We employ a combination of analytical calculations and simulation experiments to investigate both stationary and transient performance features, and demonstrate close agreement with measurements on a wide-area network testbed running the Ethereum protocol.

I. Introduction

The Blockchain is a peer-to-peer (P2P) distributed ledger technology for establishing trust and consensus. The Blockchain is the underlying digital fabric for currently deployed crypto-currency systems like Bitcoin [15] and Ethereum [18] with billions of dollars in market capitalization [4]. Different types of information or digital assets can be stored in a Blockchain, and the network implementing the Blockchain defines the type of information contained in the transactions. The advantages of the Blockchain and the application of associated smart contracts [5] in enabling trust and data integrity in distributed system environments have recently spurred interest in its use in the Internet-of-Things (IoT) as a key mechanism for supporting novel applications in digital health, smart energy, asset tracking and smart transportation [5], [14].

Blockchain is enabled by the integration of several technologies such as distributed content dissemination and storage, cryptographic hash, asymmetric digital signature and distributed consensus mechanisms. The Blockchain architecture consists of three components: (i) a distributed P2P network interconnecting member nodes, where data is spread from one node to another using gossip or broadcast, (ii) a shared ledger (data record) collectively updated by the member nodes, and (iii) digital transactions (issued by applications) that are validated by member nodes and included in the ledger as a sequence of chained blocks. Member nodes see identical copies of the ledger and contribute to the collective process of validating the digital transactions.

Consensus based on Proof-of-Work (PoW): In crypto-currency Blockchains, such as Bitcoin and Ethereum, nodes rely on Proof-of-Work (PoW) [15] as the basis for validating new blocks and achieving consensus on appending them to the ledger. PoW involves searching for a random number (nonce), which when hashed (e.g., using SHA-256) in combination with the hash of the previous block, results in a hash that begins with a specific number of leading zero bits, see Figure 1 and further details in [15]. The average work required in PoW is exponential in the number of zero bits required and hence may require considerable CPU investment. A block is considered successfully 'mined' by a node if all the transactions grouped within it are valid and PoW is done. Network nodes that execute PoW are called 'miners' and they receive rewards for mining blocks that get included in the Blockchain. Note that, interestingly, verification that a block is mined correctly can be easily done by executing a single hash [15].

Network-wide consensus: The goal in the Blockchain is that distributed nodes (miners) collectively build a ledger as a linear chain of chronologically ordered blocks (of transactions). New transactions are broadcast to all nodes. Each miner collects new transactions into a block and begins to execute PoW, and when done, it broadcasts the block (including the found nonce) to all other nodes. Nodes accept a block with valid transactions and PoW, append it to the chain, and begin to work on the next block using the hash of the accepted block, see Figure 1. Due to the distributed nature of the network, disagreement among the nodes might arise regarding the most recent added block to build upon. This can be due to network communication and processing delays, and the fact that multiple valid blocks might be generated nearly simultaneously by different miners. Therefore, branches (or forks) may result in different views of the ledger by different nodes. Consensus among the nodes is ultimately achieved by considering the longest chain to be the correct one and working on extending it. This unfortunately leads to 'orphaned' blocks on the abandoned forks where expended PoW is wasted. The growth rate of the main branch of the Blockchain is a crucial performance measure as it relates to how fast transactions are

The research of N. Papadis and L. Tassiulas was supported partly by the US Office of Naval Research (ONR) under award N00014-14-1-2190 and the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001.

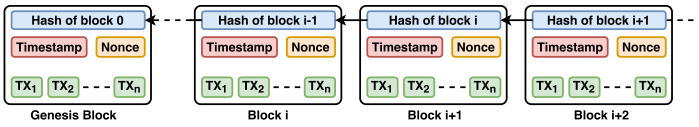

Figure 1. Schematic diagram of Blockchain structure

processed and added to the ledger [17].

As Blockchain networks evolve to large-scale industrial deployments with geographically distributed sites, block transfer and processing delays arise as a critical issue. These delays will cause more frequent inconsistencies and create greater potential for forks, which will slow down the overall growth rate. The higher frequency of forks and decline in the overall growth rate will indirectly also increase the vulnerability to an adversarial attacker with significant hashing power whose objective is to change history or execute 'double spend', i.e., successfully spend some money (bitcoins) more than once [1], [2], [11], [12], [16]. To improve security, a block is considered 'confirmed' as part of the Blockchain only if a certain minimum number of blocks are appended after it, adding more complexity to understanding Blockchain performance.

Motivated by these issues, we develop stochastic network models to capture the impact of the block dissemination delay and the network hashing power on the Blockchain evolution. We leverage the models to analyze the Blockchain performance in terms of the growth rate and the amount of wasted work as a function of the delay. This would allow adjusting the 'difficulty-of-work' to achieve a certain growth rate in the presence of delay [13]. We further investigate the impact of delay on the required adversarial hashing power to mount a successful attack, and show that the requirement can be below what is commonly believed based on heuristic arguments (i.e., 51% of the hashing power [8]). We employ a combination of analytical calculations and simulation experiments, and demonstrate close agreement between results obtained from our model and from measurements in a wide-area network testbed running the Ethereum protocol.

Specifically, the key contributions and insights may be summarized as follows.

1) We introduce a stochastic network model to describe the joint dynamics of the 'frontier' process, tracking the newly mined or received blocks in the individual branches of the various miners beyond the common ledger. We show how the long-term growth rate and the occurrence of forks can be expressed in terms of the stationary distribution of the 'frontier' process.

2) We provide explicit expressions for the stationary distribution of the 'frontier' process for the case of two miners, and use that to obtain the long-term growth rate. We derive asymptotic expansions for the stationary distribution in a low-delay regime, and demonstrate that the associated growth rate estimate is in close agreement with measurements in the Ethereum testbed.

3) We also establish asymptotic expansions for the stationary distribution and long-term growth rate in symmetric scenarios with an arbitrary number of nodes. To corroborate the analytical estimates, we show that they provide an excellent match with measurements in the Ethereum testbed, even when the various miners are not strictly homogeneous.

4) We examine the adverse impact of delay on the security of the Blockchain, and use a combination of analysis and simulation to determine the probability of a successful attack as a function of the delay parameter. The results show that delay can significantly increase the vulnerability to an attack, unless the confirmation threshold is increased, at the expense of a correspondingly longer lag in finalizing transactions.

The remainder of the paper is organized as follows. In Section II we further discuss the basic Blockchain mechanics and main modeling assumptions, and provide a high-level description of the Ethereum testbed. In Section III we consider the stationary behavior and long-term growth rate of the Blockchain in two-node scenarios, and then turn in Section IV attention to scenarios with an arbitrary number of nodes. Section V examines the implications of delay for the vulnerability of Blockchain networks to adversarial attacks.

II. Basic Blockchain mechanics and preliminaries

In this section we introduce the basic Blockchain mechanics, discuss the main modeling assumptions we have adopted, and provide a high-level description of the testbed setup for the Ethereum measurements we have conducted.

A. Basic Blockchain mechanics

We will consider Blockchain dynamics in a system with $K \geq 2$ nodes. The evolution of the Blockchain may be described by $L_0(t)$, representing the length of the main branch at time t (or possibly a vector-valued variable specifying the detailed contents of the common ledger), in conjunction with the 'frontier' $M(t) = (M_1(t), M_2(t), \ldots, M_K(t))$ tracking the progress in the individual branches of the various nodes beyond the common ledger. Specifically, $M_k(t) = (M_{k1}(t), M_{k2}(t), \ldots, M_{kL_k(t)}(t))$ represents the newly mined or received blocks held by the k-th node at time t beyond the main branch, $k = 1, \ldots, K$. The n-th component $M_{kn}(t) \in \{1, 2, \ldots, K\}$ indicates which node originally mined the n-th block in the individual branch of node k at time t, and $L_k(t)$ denotes the length of the individual branch of node k at time t.

When node k mines a block, a transition occurs from state M to state M' with $M'_k = (M_k, k)$ and $M'_l = M_l$ for all $l \neq k$. When nodes k and l communicate and node k has a strictly shorter branch, i.e., $L_k < L_l$, it abandons its blocks, and proceeds from the (strictly) longer branch of node l. Thus a transition occurs from state M to state M' with $M'_k = M_l$ and $M'_m = M_m$ for all $m \neq k$. Let $c \geq 0$ be the number of blocks that were common to all nodes but node k just before its exchange with node l. Then the length of the main branch will increase by c, and the first c blocks of each of the individual branches will join the common ledger, thus increasing $L_0(t)$ by c, clipping the first c components of each of the vectors $M_k(t)$, and reducing all the $L_k(t)$ variables by c, $k = 1, \ldots, K$. If the branches of nodes k and l are of equal length when they communicate, i.e., $L_k = L_l$, then both simply continue mining.

B. Modeling assumptions

The proof-of-work consensus mechanism involves nodes performing hashing operations at a huge rate, each of which

independently results in a block completion with extremely low probability. Thus nodes essentially conduct Bernoulli trials at a vast pace, each with an extremely low success probability, meaning that successes roughly occur as a Poisson process. Hence we will make the common assumption [11] that block completions at node k occur as a Poisson process of rate λ_k. As observed in [6], block propagation delays follow roughly an exponential distribution, and hence we will assume that two nodes $k \neq l$ communicate at exponentially distributed intervals with parameter $\mu_{k,l}$.

By virtue of the Markovian assumptions, $M(t)$ evolves as a Markov process. Defining $X_k(t) = L_0(t) + L_k(t)$, $k = 1, \ldots, K$ (the total length of the branch of node k), the process $X(t) = (X_1(t), \ldots, X_K(t))$ behaves in a Markovian fashion as well. Specifically, transitions from state x to state $x + e_k$ occur at rate λ_k, with e_k denoting the k-th unit vector, $k = 1, \ldots, K$. Transitions from state x to state x' with $x'_k = x'_l = \max\{x_k, x_l\}$ and $x'_m = x_m$ for all $m \neq k, l$, occur at rate $\mu_{k,l}$ if $x_k \neq x_l$.

C. Growth rate

In order to determine the long-term growth rate γ of the main branch, it is convenient to describe the evolution of the Blockchain in terms of a process $(X_0(t), Y(t))$ with $Y(t) = (Y_1(t), \ldots, Y_K(t))$. Here $X_0(t)$ is a variable which is set equal to $\min_{k=1,\ldots,K} X_k(t) = L_0(t) + \min_{k=1,\ldots,K} L_k(t)$ right after every communication between two nodes, while $Y(t) = (Y_1(t), \ldots, Y_K(t))$, with $Y_k(t) = L_0(t) + L_k(t) - X_0(t) \geq 0$, $k = 1, \ldots, K$. The process $Y(t)$ inherits the Markovian characteristics of the process $X(t)$. Transitions from state y to state $y + e_k$ occur at rate λ_k. Transitions from state y to state y' with $y'_k = y'_l = \max\{y_k, y_l\} - z$ and $y'_m = y_m - z$ for all $m \neq k, l$, with $z = \min\{\max\{y_k, y_l\}, \min_{m \neq k,l} y_m\}$, occur at rate $\mu_{k,l}$ if $y_k \neq y_l$.

It can be shown that the process $Y(t)$ is positive-recurrent and that the process $(L_1(t), \ldots, L_K(t))$ is stochastically bounded, assuming that all nodes can communicate, possibly indirectly (this condition can be expressed in terms of the matrix of communication rates $\mu_{k,l}$ being irreducible). Now observe that $Y_k(t)$ is incremented whenever $L_k(t)$ is incremented, and decremented by z whenever $X_0(t)$ is incremented by z. This means that the long-term growth rate $\lim_{t\to\infty} X_0(t)/t$ of $X_0(t)$ exists and can be expressed in terms of the stationary distribution $\pi(y)$ of the process $Y(t)$. Because of the identity $L_0(t) + L_k(t) \equiv X_0(t) + Y_k(t)$, and the fact that $0 \leq Y_k(t) \leq L_k(t)$ by definition, this also implies that the long-term growth rate $\gamma = \lim_{t\to\infty} L_0(t)/t$ of the length of the main branch exists and equals that of $X_0(t)$.

We conclude that once the stationary distribution $\pi(y)$ is obtained, the long-term growth rate may be calculated as

$$\begin{aligned}\gamma &= \sum_{y\in\mathbb{N}^K} \pi(y) \sum_{k=1}^{K}\sum_{l=1}^{K} \mu_{k,l} z_{k,l}(y) \mathbb{I}\{y_k \neq y_l\} \\ &= \sum_{y\in\mathbb{N}^K} \pi(y) \sum_{k=1}^{K} \lambda_k \mathbb{I}\left\{y_k = \max_{l=1,\ldots,K} y_l\right\},\end{aligned}$$

where $z_{k,l}(y) = \min\{\max\{y_k, y_l\}, \min_{m\neq k,l} y_m\}$. Due to orphaned blocks, the long-term growth rate will be less than the aggregate block completion rate $\lambda = \sum_{k=1}^{K} \lambda_k$, and the wastage rate may be computed as

$$\theta = \lambda - \gamma = \sum_{y\in\mathbb{N}^K} \pi(y) \sum_{k=1}^{K} \lambda_k \mathbb{I}\left\{y_k < \max_{l=1,\ldots,K} y_l\right\}.$$

D. Measurement testbed setup

We now provide a description of the testbed configuration for the Ethereum measurements that we have conducted to experimentally validate the analytical calculations. Our experiments are built on top of the GENI cloud platform [9], [3] using the Ethereum protocol. We have created a testbed of multiple virtual machines (VMs) all running Ubuntu 16.04, each having 1 GB of RAM, and an Intel Xeon X5650 2.67 GHz CPU. Each of these VMs plays the role of a miner in our experiments, and runs a *geth v1.6.6* Ethereum client [10], which is the official *golang* implementation of the Ethereum protocol. We modify the difficulty of the Proof-of-Work D in the *geth* implementation, so that instead of using the block canonical difficulty formula described in the Ethereum yellow paper [18], we set it to different values for the purpose of our experimental investigation. The values of D may be chosen to match the mining power of the system. In the real Ethereum system, these values are very large because they are set proportional to the total mining power of the system [7]. The higher the value of D is, the harder it is for the miners to solve the proof-of-work.

In our experiments, we have considered two topologies: a two-miner and a five-miner topology. For each topology we have tested three values of D: 100, 500 and 1000. We chose these values to match the hashing power of the VMs in our testbed. We measured the average block generation rate, the percentage of orphaned blocks, and the system delay for various difficulty levels and for different topologies. The block transfer delays observed in the measurements are consistent with the actual block delays in the live Ethereum Blockchain [7]. The detailed measurement results for the two-miner and five-miner topologies will be presented and used to corroborate the analytical predictions in Subsections III-C and IV-B, respectively.

III. Two-node scenarios

In order to gain some initial insight, we focus in this section on two-node scenarios, using a combination of analytical calculations and testbed experiments.

In case of $K = 2$ nodes, the Blockchain dynamics considerably simplify, as communication between the two nodes either leaves the state unaltered or yields a transition to state $(0, 0)$. Specifically, the process $(Y_1(t), Y_2(t))$, as introduced in the previous section, makes transitions from state (y_1, y_2) to state $(y_1 + 1, y_2)$ at rate λ_1, to state $(y_1, y_2 + 1)$ at rate λ_2, and to state $(0, 0)$ at rate $\mu \equiv \mu_{1,2}$ if $y_1 \neq y_2$. The stationary distribution $\pi(y_1, y_2)$ satisfies the balance equations

$$\lambda\pi(0,0) = \mu \sum_{y_1\neq y_2} \pi(y_1, y_2)$$

and $(\lambda + \mu\mathbb{I}\{y_1 \neq y_2\})\pi(y_1, y_2) =$

$$\lambda_1 \mathbb{I}\{y_1 > 0\}\pi(y_1 - 1, y_2) + \lambda_2 \mathbb{I}\{y_2 > 0\}\pi(y_1, y_2 - 1)$$

for $(y_1, y_2) \neq (0,0)$. The solution is obtained in [11], but the normalization constant is not provided explicitly, and involves a numerical computation.

For the purpose of analysis, it will be useful to also introduce a slightly different version of the process $(\tilde{Y}_1(t), \tilde{Y}_2(t))$ where transitions from state (y_1, y_2) to state $(0,0)$ occur even when $y_1 = y_2$. Thus the stationary distribution $\tilde{\pi}(y_1, y_2)$ satisfies the balance equations

$$(\lambda + \mu)\tilde{\pi}(0,0) = \mu$$

and $(\lambda + \mu)\tilde{\pi}(y_1, y_2) =$

$$\lambda_1 \mathbb{I}_{\{y_1 > 0\}} \tilde{\pi}(y_1 - 1, y_2) + \lambda_2 \mathbb{I}_{\{y_2 > 0\}} \tilde{\pi}(y_1, y_2 - 1)$$

for $(y_1, y_2) \neq (0,0)$, and can be readily solved from these equations in a recursive manner:

$$\tilde{\pi}(y_1, y_2) = \frac{\mu}{\lambda + \mu} \frac{(y_1 + y_2)!}{y_1! y_2!} \left(\frac{\lambda_1}{\lambda + \mu}\right)^{y_1} \left(\frac{\lambda_2}{\lambda + \mu}\right)^{y_2}. \quad (1)$$

The probability generating function of the stationary distribution $\pi(y_1, y_2)$ is easily obtained as

$$\mathbb{E}\left\{z_1^{\tilde{Y}_1} z_2^{\tilde{Y}_2}\right\} = \frac{\mu}{\mu + \lambda_1(1 - z_1) + \lambda_2(1 - z_2)}. \quad (2)$$

The long-term growth rate may be calculated from the stationary distribution $\tilde{\pi}(y_1, y_2)$ as

$$\begin{aligned} \gamma &= \sum_{(y_1, y_2) \in \mathbb{N}^2} \tilde{\pi}(y_1, y_2)(\lambda_1 \mathbb{I}_{\{y_1 \geq y_2\}} + \lambda_2 \mathbb{I}_{\{y_1 \leq y_2\}}) \\ &= \mu \sum_{(y_1, y_2) \in \mathbb{N}^2} \tilde{\pi}(y_1, y_2) \max\{y_1, y_2\}. \end{aligned}$$

The wastage rate may be computed as

$$\begin{aligned} \theta &= \sum_{(y_1, y_2) \in \mathbb{N}^2} \tilde{\pi}(y_1, y_2)(\lambda_1 \mathbb{I}_{\{y_1 < y_2\}} + \lambda_2 \mathbb{I}_{\{y_1 > y_2\}}) \\ &= \mu \sum_{(y_1, y_2) \in \mathbb{N}^2} \tilde{\pi}(y_1, y_2) \min\{y_1, y_2\}. \end{aligned}$$

While the stationary distribution $\pi(y_1, y_2)$ is not needed in order to calculate the growth rate and the wastage rate, and is more involved than the distribution $\tilde{\pi}(y_1, y_2)$, we now show that it can in fact be obtained in closed form as well by exploiting the fact that the processes $(Y_1(t), Y_2(t))$ and $(\tilde{Y}_1(t), \tilde{Y}_2(t))$ are closely related as

$$(Y_1(t), Y_2(t)) = (\tilde{Y}_1(t), \tilde{Y}_2(t)) + U(t)(1,1),$$

with the random variable $U(t)$ representing the common value of $Y_1(u(t)^+)$ and $Y_2(u(t)^+)$ right after the most recent communication event before time t at time $u(t)$, which is independent of $\tilde{Y}_1(t)$ and $\tilde{Y}_2(t)$.

It is not difficult to derive the probability generating function of the stationary distribution of the random variable $U(t)$ (the details are skipped because of page constraints):

$$\mathbb{E}\left\{z^U\right\} = \frac{1 - \phi}{1 - \phi\sqrt{\frac{1 - 4\psi}{1 - 4\psi z}}} = \frac{1 - \frac{\mu}{\lambda+\mu}\frac{1}{\sqrt{1-4\psi}}}{1 - \frac{\mu}{\lambda+\mu}\frac{1}{\sqrt{1-4\psi z}}}, \quad (3)$$

with $\phi = \dfrac{\mu}{\lambda + \mu} \dfrac{1}{\sqrt{1 - 4\psi}}$ and $\psi = \dfrac{\lambda_1 \lambda_2}{(\lambda + \mu)^2} < \dfrac{1}{4}$.

By virtue of the independence between $(\tilde{Y}_1(t), \tilde{Y}_2(t))$ and $U(t)$, we obtain from (2) and (3)

$$\mathbb{E}\left\{z_1^{Y_1} z_2^{Y_2}\right\} = \frac{\mu}{\mu + \lambda_1(1 - z_1) + \lambda_2(1 - z_2)} \frac{1 - \frac{\mu}{\lambda+\mu}\frac{1}{\sqrt{1-4\psi}}}{1 - \frac{\mu}{\lambda+\mu}\frac{1}{\sqrt{1-4\psi z_1 z_2}}}.$$

This in particular yields an explicit expression for the normalization constant in [11]:

$$\pi(0,0) = \mathbb{P}\{(Y_1, Y_2) = (0,0)\} = \frac{\mu}{\lambda + \mu} \mathbb{P}\{U = 0\} =$$

$$\frac{\mu}{\lambda + \mu} \frac{1 - \frac{\mu}{\lambda+\mu}\frac{1}{\sqrt{1-4\psi}}}{1 - \frac{\mu}{\lambda+\mu}} = \frac{\mu}{\lambda}\left(1 - \frac{\mu}{\lambda + \mu}\frac{1}{\sqrt{1 - 4\psi}}\right).$$

A. Asymptotic expansions in low-delay regime

We now analyze a low-delay regime where the mean block propagation delay $\delta = 1/\mu$ is small compared to the mean block completion times $1/\lambda_k$, i.e., $\mu \gg \lambda_k$, $k = 1, 2$. This asymptotic regime provides not only mathematical tractability, but is also highly relevant as block propagation delays, while substantial, tend to be two orders of magnitude smaller than block completion times (e.g., seconds vs. minutes) [6], [11].

The next proposition provides an asymptotic expansion for the stationary distribution $\tilde{\pi}(y_1, y_2)$ in the low-delay regime, and follows from (1) through the series expansion

$$\frac{\mu}{\lambda + \mu} = 1 - \lambda\delta + \lambda^2\delta^2 - \lambda^3\delta^3 + \mathrm{o}(\delta^2).$$

Proposition 1 (Third-order expansion of stationary distribution in two-node scenario).

$$\begin{aligned}
\tilde{\pi}(0,0) &= \frac{\mu}{\lambda + \mu} = 1 - \lambda\delta + \lambda^2\delta^2 - \lambda^3\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(1,0) &= \frac{\lambda_1 \mu}{(\lambda + \mu)^2} = \lambda_1\delta - 2\lambda\lambda_1\delta^2 + 3\lambda^2\lambda_1\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(0,1) &= \frac{\lambda_2 \mu}{(\lambda + \mu)^2} = \lambda_2\delta - 2\lambda\lambda_2\delta^2 + 3\lambda^2\lambda_2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(1,1) &= \frac{2\lambda_1\lambda_2 \mu}{(\lambda + \mu)^3} = 2\lambda_1\lambda_2\delta^2 - 6\lambda\lambda_1\lambda_2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(2,0) &= \frac{\lambda_1^2 \mu}{(\lambda + \mu)^3} = \lambda_1^2\delta^2 - 3\lambda\lambda_1^2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(0,2) &= \frac{\mu\lambda_2^2}{(\lambda + \mu)^3} = \lambda_2^2\delta^2 - 3\lambda\lambda_2^2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(2,1) &= \frac{2\lambda_1^2\lambda_2 \mu}{(\lambda + \mu)^4} = 3\lambda_1^2\lambda_2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(1,2) &= \frac{2\lambda_1\lambda_2^2 \mu}{(\lambda + \mu)^4} = 3\lambda_1\lambda_2^2\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(3,0) &= \frac{\lambda_1^3 \mu}{(\lambda + \mu)^4} = \lambda_1^3\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(0,3) &= \frac{\lambda_2^3 \mu}{(\lambda + \mu)^4} = \lambda_2^3\delta^3 + \mathrm{o}(\delta^3) \\
\tilde{\pi}(y_1, y_2) &= \mathrm{o}(\delta^3) \qquad \text{for all } y_1 + y_2 \geq 4.
\end{aligned}$$

B. Asymptotic expansions of the long-term growth rate

The next proposition provides asymptotic expansions for the growth and wastage rates in the low-delay regime based on the expansions stated in Proposition 1 for the stationary distribution $\tilde{\pi}(y_1, y_2)$.

Proposition 2 (Third-order expansion of growth rate in two-node scenario). In a two-node scenario the growth rate behaves as

$$\gamma = \lambda - \lambda_1\lambda_2[2\delta - 3\lambda\delta^2 + 4(\lambda_1^2 + 3\lambda_1\lambda_2 + \lambda_2^2)\delta^3] + o(\delta^3), \quad (4)$$

and the wastage rate behaves as

$$\theta = \lambda_1\lambda_2[2\delta - 3\lambda\delta^2 + 4(\lambda_1^2 + 3\lambda_1\lambda_2 + \lambda_2^2)\delta^3] + o(\delta^3). \quad (5)$$

The first-order term in the above asymptotic expansions may be explained as follows. Block completions occur at a total rate λ, and given that a block is completed, this occurs with probability λ_k/λ at node k, $k = 1, 2$. Then node $3 - k$ will generate a block before the next communication event if an exponential time with parameter μ exceeds an exponential time with parameter λ_{3-k}, which happens with probability λ_{3-k}/μ. Since the two nodes will be in sync an overwhelming fraction of the time in the low-delay regime, the block generated by node $3 - k$ will be at the same height as that generated by node k with high probability, so one of the two will ultimately be wasted. Thus wasted blocks are generated at roughly rate $\lambda\left[\frac{\lambda_1}{\lambda}\frac{\lambda_2}{\mu} + \frac{\lambda_2}{\lambda}\frac{\lambda_1}{\mu}\right] = 2\lambda_1\lambda_2\delta$, which corresponds to the leading term in Equation (5) for the wastage rate.

Equation (5) demonstrates that, asymptotically, a reduction of the delay δ by a factor α reduces the wastage rate θ by the same factor α. Or equivalently, increasing the block completion rates λ_k by a factor α increases the wastage rate θ asymptotically by the same factor, and thus increases the wastage rate relative to the total block completion rate λ by a factor α. We further observe that, for a given value of λ, the wastage rate achieves its maximum $\lambda^2\delta/2$ when λ_1 and λ_2 are equal and its minimum 0 when λ_1 or λ_2 is zero.

It can be checked that the above asymptotic expansions for the stationary distribution $\pi(y_1, y_2)$ are identical to those in a (fictitious) modified system where the block generation process is halted as soon as one of the states $(0, 3)$, $(1, 2)$, $(2, 1)$ or $(0, 3)$ is reached, i.e., the state space is truncated to $\{(y_1, y_2) \in \mathbb{N}^2 | y_1 + y_2 \leq 3\}$. More generally, truncating the process at $y_1 + y_2 = L$ does not affect the stationary distribution up to the L-th order. At the same time, it can be shown that the growth rate is reduced when the block generation process is ever halted. **Thus, truncating the state space at level L yields a conservative estimate (lower bound) for the growth rate, which is asymptotically exact up to L-th order in the low-delay regime.** We will later leverage this observation to obtain asymptotic approximations for scenarios with an arbitrary number of nodes, where the Blockchain dynamics are fundamentally more complex.

C. Comparison between analytical calculations and testbed measurements

We now present the measurement results from the Ethereum testbed described in Subsection II-D for a two-node topology, and compare these with the analytical predictions for the long-term growth rate derived in Proposition 2.

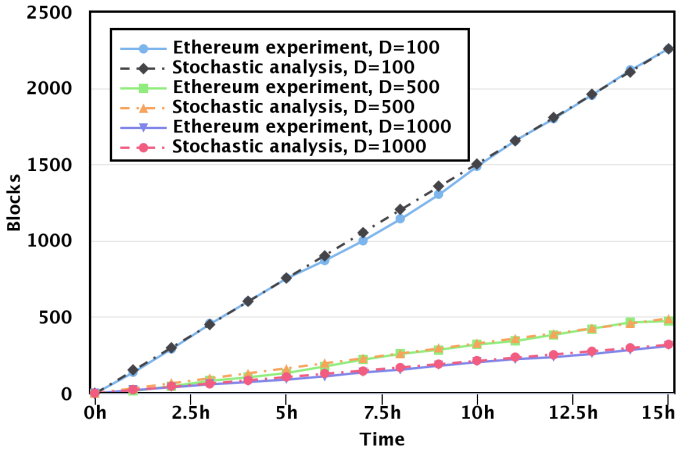

Figure 2. Blockchain growth for a two-node network over time as a function of the difficulty-of-work D obtained from stochastic analysis vs. Ethereum testbed measurements

Figure 2 plots the evolution of the Blockchain over time for various difficulty levels inferred from the measurements, along with the analytical estimates calculated from (4). Observe that the measurement values and analytical estimates show an excellent match, providing strong experimental validation for the stochastic analysis. Further, as D increases, the block completion rate decreases, as shown in Table I, which results in a decrease in the growth rate as reflected in Figure 2.

Table I. EXPERIMENTAL MEASUREMENTS FOR TWO-NODE NETWORK

Difficulty	Block completion rate (blocks per second)	Forks (%)	Delay (seconds)
100	0.022	4.58	3.96
500	0.0047	2.48	7.48
1000	0.0029	0.48	6.84

IV. ARBITRARY NUMBER OF NODES

In the previous section we focused on two-node scenarios to gain some initial insight. In this section we turn attention to scenarios with an arbitrary number of nodes. As reflected in the description in Section II, the Blockchain dynamics in terms of the process $(Y_1(t), Y_2(t), \ldots, Y_K(t))$ are fundamentally more complex with three or more nodes, since communication between two nodes will generally no longer cause a transition to the all-zero state. Hence we will focus on symmetric scenarios in the low-delay regime, with identical block completion rates $\lambda_k \equiv \lambda_0$ for all $k = 1, \ldots, K$ and communication rates $\mu_{k,l} \equiv \mu$ for all $k \neq l$.

Because of the symmetry among the nodes, we do not need to keep track of the state of each individual node, but only count the number of nodes in a given state. Specifically, define $Z_i(t) = \sum_{k=1}^{K} \mathbb{I}_{\{Y_k(t) = i\}}$ as the number of nodes with i blocks at time t, and denote $Z(t) = (Z_i(t))_{i=0,\ldots,L}$. As we consider a low-delay regime where it is rare for any individual node k to have a large value of Y_k, we will assume

that the block generation process at each node k is halted as soon as Y_k reaches some value L. Then $\{Z(t)\}_{t\geq 0}$ evolves as a Markov process with transitions from state z to state $z+e_i-e_{i-1}$ at rate $\lambda_0 z_{i-1}$, $i=1,2,\ldots,L$, and transitions from state z to state $z+e_i-e_h$ at rate $\mu z_h z_i$, $0\leq h<i\leq K$. The state space is $\mathcal{S}=\{(z_0,\ldots,z_L)\in\{0,1,\ldots,K\}^{L+1}:\sum_{l=0}^{L} z_l = K\}$. For any $(z_0,z_1,\ldots,z_L)\in\mathcal{S}$, denote by $\hat{\pi}(z)=\lim_{t\to\infty}\mathbb{P}\{Z(t)=z\}$ the stationary distribution of the $Z(t)$ process.

The next proposition provides asymptotic expansions for the stationary distribution $\hat{\pi}(z_0,z_1,z_2)$ in the low-delay regime, and follows from inspection of the relevant balance equations for the truncated process.

Proposition 3 (Second-order expansion of stationary distribution in symmetric three-node scenario).

$$\begin{aligned}
\hat{\pi}(3,0,0) &= 1-3\lambda_0\delta+\frac{43}{4}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(2,1,0) &= \frac{3}{2}\lambda_0\delta-\frac{27}{4}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(1,2,0) &= \frac{3}{2}\lambda_0\delta-\frac{15}{2}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(2,0,1) &= \frac{3}{4}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(0,3,0) &= \frac{1}{2}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(1,1,1) &= \lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(1,0,2) &= \frac{5}{4}\lambda_0^2\delta^2+\mathrm{o}(\delta^2)\\
\hat{\pi}(0,2,1) &= \mathrm{o}(\delta^2)\\
\hat{\pi}(0,1,2) &= \mathrm{o}(\delta^2)\\
\hat{\pi}(0,0,3) &= \mathrm{o}(\delta^2).
\end{aligned}$$

The next proposition provides asymptotic expansions for the growth and wastage rates in the low-delay regime based on the expansions stated in Proposition 3 for the stationary distribution $\hat{\pi}(z_0,z_1,z_2)$.

Proposition 4 (Second-order expansion of growth rate in symmetric three-node scenario). In a three-node scenario the growth rate behaves as

$$\gamma=\lambda_0[3\hat{\pi}(3,0,0)+\hat{\pi}(2,1,0)+2\hat{\pi}(1,2,0)+3\hat{\pi}(0,3,0)+\hat{\pi}(1,1,1)+2\hat{\pi}(1,0,2)+\hat{\pi}(0,2,1)+2\hat{\pi}(0,1,2)+3\hat{\pi}(0,0,3)]=3\lambda_0-\frac{9}{2}\lambda_0^2\delta+\frac{65}{4}\lambda_0^3\delta^2+\mathrm{o}(\delta^2), \quad (6)$$

and the wastage rate is

$$\theta=\lambda_0[2\hat{\pi}(2,1,0)+\hat{\pi}(1,2,0)+2\hat{\pi}(2,0,1)+2\hat{\pi}(1,1,1)+\hat{\pi}(1,0,2)+2\hat{\pi}(0,2,1)]=\frac{9}{2}\lambda_0^2\delta-\frac{65}{4}\lambda_0^3\delta^2+\mathrm{o}(\delta^2). \quad (7)$$

The first-order term in the above asymptotic expansions may be explained, and in fact extended to an asymmetric scenario, as follows. Block completions occur at a total rate λ, and given that a block is completed, this occurs with probability λ_k/λ at node k, $k=1,2,3$. Let us say that it is node 1. Then one of the nodes 2 or 3 will generate a block before the next communication event with node 1 if an exponential time with parameter 2μ exceeds an exponential time with parameter $\lambda_2+\lambda_3$, which happens with probability $(\lambda_2+\lambda_3)/(2\mu)$. Since the three nodes will be in sync an overwhelming fraction of the time in the low-delay regime, the block generated by either node 2 or 3 will be at the same height as that generated by node 1 with high probability, so one of the two will ultimately be wasted. The communication will leave either node 2 or 3 behind by a block with equal probability. Let us say that it is node k. Then node k will generate a block before the next communication event with one of the other nodes if an exponential time with parameter 2μ exceeds an exponential time with parameter λ_k, which happens with probability $\lambda_k/(2\mu)$, in which case yet a further block will eventually be wasted. Thus, the expected number of wasted blocks right after the block completion at node 1 is

$$\frac{\lambda_2+\lambda_3}{2\mu}+\frac{1}{2}\frac{\lambda_2}{2\mu}+\frac{1}{2}\frac{\lambda_3}{2\mu}=\frac{3(\lambda_2+\lambda_3)}{4\mu}=\frac{3(\lambda-\lambda_1)}{4\mu}.$$

Thus, in total wasted blocks are generated at roughly rate

$$\lambda\sum_{k=1}^{3}\frac{\lambda_k}{\lambda}\frac{3(\lambda-\lambda_k)}{4\mu}=\frac{3(\lambda_1\lambda_2+\lambda_1\lambda_3+\lambda_2\lambda_3)}{2\mu}.$$

In case $\lambda_k\equiv\lambda_0$ for $k=1,2,3$, this reduces to $\frac{9}{2}\lambda_0^2\delta$, which corresponds to the leading term in Equation (7) for the wastage rate.

Equation (7) reveals that, asymptotically, a reduction of the delay δ by a factor α yields a reduction in the wastage rate θ by the same factor α, just like in the two-node scenario. Or equivalently, increasing the block completion rates λ_k by a factor α increases the wastage rate θ asymptotically by the same factor, and thus increases the wastage rate relative to the total block completion rate $\lambda=3\lambda_0$ by a factor α. Noting that $\lambda_1\lambda_2+\lambda_1\lambda_3+\lambda_2\lambda_3=\frac{1}{2}\left(\lambda^2-\sum_{k=1}^{3}\lambda_k^2\right)$, we further observe that, for a given value of λ, the wastage rate achieves its maximum $\lambda^2\delta/2$ when λ_1, λ_2 and λ_3 are equal, and its minimum 0 when only one of them is non-zero.

We now turn to scenarios with an arbitrary number of $K\geq 4$ nodes and truncation level $L=1$, and with minor abuse of notation write $\hat{\pi}(k)=\hat{\pi}(K-k,k)$ for brevity.

The next proposition provides an asymptotic expansion for the stationary distribution $\hat{\pi}(k)$ in the low-delay regime, which can be obtained from the relevant balance equations.

Proposition 5 (Expansion of stationary distribution in symmetric K-node scenario with epidemic block propagation).

$$\begin{aligned}
\hat{\pi}(0) &= 1-\lambda_0\delta KG(K)+\mathrm{o}(\delta),\\
\hat{\pi}(k) &= \lambda_0\delta Kg(k)+\mathrm{o}(\delta),\quad k=1,\ldots,K-1,
\end{aligned}$$

with $G(K)=\sum_{k=1}^{K-1}g(k)$ and $g(k)=\frac{1}{(K-k)k}$.

The next proposition provides asymptotic expansions for the growth and wastage rates in the low-delay regime based on the expansions stated in Proposition 5 for the stationary distribution $\hat{\pi}(k)$.

Proposition 6 (Expansion of growth rate in symmetric K-node scenario with epidemic block propagation). In a

symmetric K-node scenario, the growth rate behaves as

$$\gamma = K\lambda_0 \left(1 - \lambda_0 \delta \sum_{k=1}^{K-1} \frac{1}{k}\right) + \mathrm{o}(\delta), \tag{8}$$

and the wastage rate behaves as

$$\theta = \lambda_0^2 \delta K \sum_{k=1}^{K-1} \frac{1}{k} + \mathrm{o}(\delta). \tag{9}$$

The expressions for the wastage rate (and hence the growth rate) may be heuristically interpreted as follows. Suppose that replicas of a single object need to be obtained by K nodes, which engage in pairwise encounters at an exponential rate μ, and can either generate the object themselves at an exponential rate $\lambda_0 \ll \mu$, or duplicate it when they encounter another node that is already in possession of the object. Then the expected amount of time required for the $(k+1)$-th replica to be obtained after the k-th replica, is $\mathbb{E}\{T_k\} = \frac{1}{(K-k)k\mu}$. Thus the expected total number of replicas that are generated rather than duplicated over the course of this hybrid generation / epidemic propagation process, after the initial one, is

$$\sum_{k=1}^{K-1} \lambda_0 (K-k) \mathbb{E}\{T_k\} = \frac{\lambda_0}{\mu} \sum_{k=1}^{K-1} \frac{1}{k}.$$

This corresponds exactly to the fraction of redundant blocks generated as reflected in Equation (9) for the wastage rate.

A. Alternative block dissemination mechanism

So far we have assumed an epidemic type block dissemination mechanism where nodes also propagate blocks that they received from other nodes in a pure peer-to-peer manner. An alternative block dissemination scheme is where nodes only transfer blocks that they created themselves.

Assuming a truncation level $L = 1$, we now need to distinguish between nodes with one block that they generated themselves and nodes with one block that they obtained from another node. We will denote the numbers of these nodes by Z_1^a and Z_1^b, respectively. Then $\{(Z_0(t), Z_1^a(t), Z_1^b(t))\}_{t\geq 0}$ evolves as a Markov process with transitions from state z to state $z + e_1^a - e_0$ at rate $\lambda_0 z_0$ and transitions from state z to state $z + e_i^b - e_0$ at rate $\mu z_0 z_1^a$.

The next proposition provides an asymptotic expansion for the stationary distribution $\hat{\pi}(z)$ in the low-delay regime, which can be obtained from the relevant balance equations.

Proposition 7 (Expansion of stationary distribution in symmetric K-node scenario with controlled block propagation).

$$\begin{aligned} \hat{\pi}(K,0,0) &= 1 - \lambda_0 \delta K H(K) + \mathrm{o}(\delta), \\ \hat{\pi}(K-k,1,k-1) &= \lambda_0 \delta K h(k) + \mathrm{o}(\delta),\ k = 1,\ldots,K-1, \end{aligned}$$

with $H(K) = \sum_{k=1}^{K-1} h(k)$ and $h(k) = \frac{1}{K-k}$.

The next proposition provides asymptotic expansions for the growth and wastage rates in the low-delay regime based on the expansions stated in Proposition 7 for the stationary distribution $\hat{\pi}(z)$.

Proposition 8 (Expansion of growth rate in symmetric K-node scenario with controlled block propagation). In a symmetric K-node scenario, the growth rate behaves as

$$\gamma = K\lambda_0 \left(1 - \lambda_0 \delta (K-1)\right) + \mathrm{o}(\delta), \tag{10}$$

and the wastage rate behaves as

$$\theta = \lambda_0^2 \delta K (K-1) + \mathrm{o}(\delta). \tag{11}$$

The expressions for the wastage rate (and hence the growth rate) may be intuitively explained as follows. Suppose that replicas of a single object need to be obtained by K nodes, which engage in pairwise encounters at an exponential rate μ, and can either generate the object themselves at an exponential rate $\lambda_0 \ll \mu$, or duplicate it when they encounter the very first node that generated the object. Then the expected amount of time required for the $(k+1)$-th replica to be obtained after the k-th replica, is $\mathbb{E}\{U_k\} = \frac{1}{(K-k)\mu}$. Thus the expected total number of replicas that are generated rather than duplicated over the course of this hybrid generation / controlled dissemination process, after the initial one, is

$$\sum_{k=1}^{K-1} \lambda_0 (K-k) \mathbb{E}\{U_k\} = \frac{\lambda_0}{\mu}(K-1).$$

This corresponds exactly to the fraction of redundant blocks generated as reflected in Equation (11) for the wastage rate.

B. Comparison between analytical calculations and testbed measurements

We now present the measurement results from the Ethereum testbed described in Subsection II-D for a five-node topology and compare these with the analytical estimates for the long-term growth rate derived in Propositions 6 and 8.

Figure 3 shows the evolution of the Blockchain over time for various difficulty levels as obtained from the measurements, along with the analytical prediction computed from (8). As before, note that the measurement values and analytical estimates show close agreement, corroborating the results from the stochastic analysis for larger number of nodes as well. Also, as D increases, the block completion rate decreases, as displayed in Table I, which leads to a decrease in the growth rate as shown in Figure 3.

Comparison of Tables I and II additionally shows that the fraction of orphaned blocks increases as the number of miners increases. This is expected due to the fact that with more miners in the network, it is more likely for several miners to solve the proof-of-work around the same time. On the other hand, increasing the difficulty D of the proof-of-work makes this probability smaller, as is well reflected by the experimental figures in Tables I and II.

The asymptotic formula (8) provided a more accurate prediction for the growth rate observed in the measurements than formula (10), especially in an absolute sense. However, a detailed inspection of the statistics for the number of forks and orphan blocks revealed that the counterpart (11) of the latter formula produced a slightly better estimate for the wastage rate at high difficulty-of-work values.

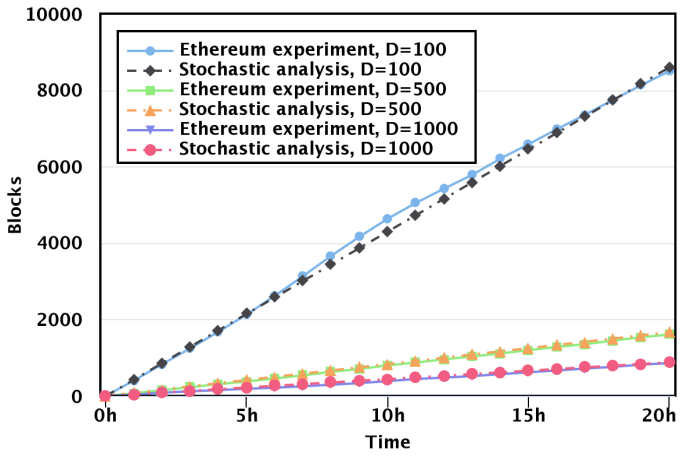

Figure 3. Blockchain growth for a five-node network over time as a function of the difficulty-of-work D obtained from stochastic analysis vs. Ethereum testbed measurements

Table II. EXPERIMENTAL MEASUREMENTS FOR FIVE-NODE NETWORK

Difficulty	Block completion rate (blocks per second)	Forks (%)	Delay (seconds)
100	0.026	9.4	1.5
500	0.0048	6	3.5
1000	0.0025	4.17	4.45

V. DELAY IMPLICATIONS FOR ADVERSARIAL ATTACKS

So far, both in the analysis and in the experiments, we have implicitly assumed that all the miners behave honestly, following the protocol. In this section, we consider an attack scenario, where an external node, the 'attacker', tries to double-spend. For example, the attacker purchased goods from a merchant using a certain number of coins and then reuses the same coins for other purposes. Before launching the attack, it secretly mines a block containing a double-spending transaction [16], which extends one of the current blocks of the common Blockchain. Then it starts mining on top of its block, until it manages to create a strictly longer chain than the chain known at the time by the honest network. As soon as this happens, the attacker disseminates its own longer double-spending branch, the honest nodes abandon the old branch and adopt the attacker's branch, and the attack is considered successful.

In the Blockchain protocol, a feature that serves to limit the possibility of a double-spend is that a merchant who makes a transaction does not dispatch the goods until a certain number of blocks have been added to the Blockchain on top of the block containing the transaction. This number of blocks is called the 'confirmation threshold', and we denote it by n. In this section, we show through simulations the critical role of delay for the vulnerability of the Blockchain, as well as the beneficial effect of a larger confirmation threshold.

We conduct simulations of the model described in Section II for a network of $K = 100$ miners, with the same fixed block generation rate (which is proportional to the hash power) $\lambda_0 = 1/30$ sec^{-1} per miner, and the same delay $\delta = 1/\mu$ for the communication between any two miners, for different values of δ. Note that the time scale is arbitrary, since what really matters is the relative values of the communication and block generation rates, but for convenience we choose the time scale of seconds. The attacker's block generation rate is λ_a. We are interested in the time it takes the attacker to mine more blocks than the honest network, only for chain lengths greater than n (otherwise, if an attack were launched at length less than n, the merchant, who would not have dispatched the goods yet, would realize and would not do so afterwards either).

We denote by p the hash power percentage of the honest network and by q the hash power percentage of the attacker ($q = \lambda_a/(K\lambda_0 + \lambda_a), p + q = 1$). We use and compare four methods for calculating the probability of a successful attack, in each of which the attacker generates blocks according to a model that takes delay into account in a different manner.

Method 1: We assume that the number of blocks m found by the attacker in the time T_n it takes the honest network to mine n blocks follows a negative binomial distribution: it is 'the number of successes (blocks found by the attacker) before n failures (blocks found by the honest network), with a probability q of success' [16]. Delay plays no role. We use the formula for the probability r of a successful attack as derived in [16], eq. (1):

$$r = \begin{cases} 1 - \sum_{m=0}^{n} \binom{m+n-1}{m}(p^n q^m - p^m q^n), & \text{if } q < p \\ 1, & \text{if } q \geq p \end{cases}$$

Method 2: We consider a similar negative binomial distribution for m, but with success probability q' adjusted to reflect delay. Specifically, we set $q' = \lambda_a/(\lambda_a + \Lambda')$, with $\Lambda' = n/\text{mean}(T_n) < K\lambda_0$, where the different values of T_n are obtained from simulating the honest network and then calculating the time difference of the first and the last block for different windows of n blocks.

Method 3: We generate m for multiple windows of n blocks from the simulation, as a Poisson random variable with rate $\lambda_a T_n$. We find the sampled distribution $P(m)$ and calculate $r = \sum_m P(m) \min\left(\frac{q}{p}, 1\right)^{\max(n-m,0)}$ [16], where the summation is over the m's for which the empirical $P(m)$ is nonzero.

Method 4: We calculate $r = \mathbb{P}\left\{\bigcup_{k \geq n}\{B_k < T_k\}\right\}$, where B_k and T_k are the times needed for the attacker and the honest network respectively to generate k blocks. The calculation is done as the percentage of the values of T_k from the simulation for which the event $\bigcup_{k \geq n}\{B_k < T_k\}$ occurs, where B_k is the sum of k exponentially distributed time intervals with parameter $1/\lambda_a$.

The probability of a successful attack r as a function of the communication delay δ is shown in Figure 4. The initial claim in [15] was that the Blockchain is secure against double-spend attempts as long as $q < 50\%$. However, as Figure 4 shows, increasing delay can increase the probability of a successful attack even for significantly less power by several orders of magnitude, even in the order of 10^{-1} in a high-delay regime. The curve for Method 1 gives a constant lower bound, since it does not account for the delay, in contrast to Method 2, which gives an upper bound. The curves for Methods 3 and 4 move together and between the two bounds, indicating

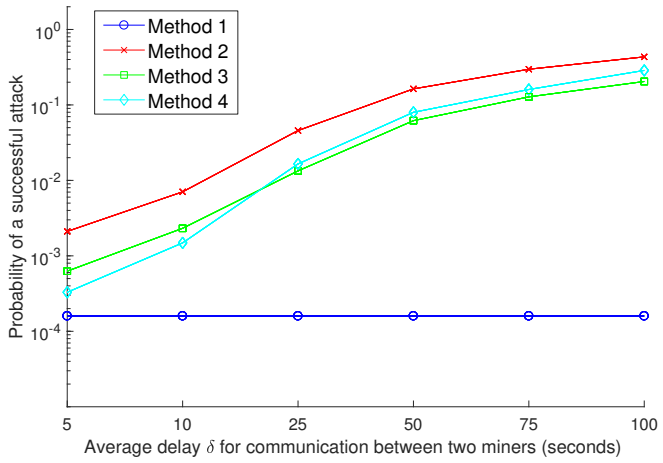

Figure 4. Success probability r with block confirmation threshold $n = 5$ for an attacker with $q = 6\%$ of the total hashing power, and different values of the communication delay δ

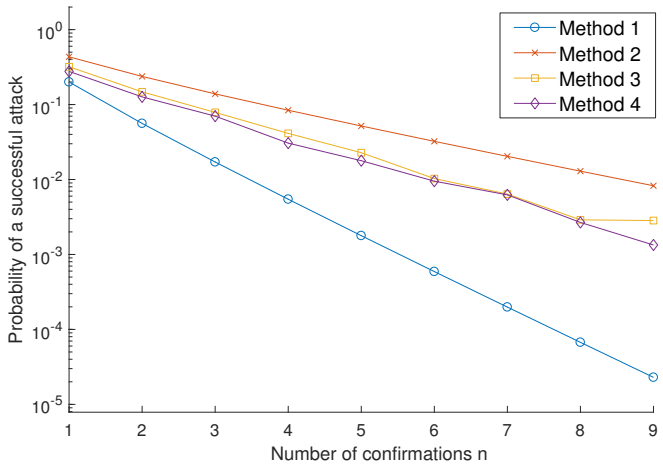

Figure 5. Success probability r for an attacker with $q = 10\%$ of the total hashing power, for $\delta = 10$ sec and different values of the confirmation threshold n

that the adjusted Poisson process assumption of Method 2 is conservative and that the actual block generation process is slightly better behaved.

The probability r of a successful attack for $q = 10\%$ and for different values of the confirmation threshold n is shown in Figure 5. Again, delay plays an important role, with the probability values that account for the delay (Methods 2-4) being higher by orders of magnitude than the ones for Method 1, even for large thresholds n. With small n, the success probability is very high (in the order of 10^{-1} or higher), but it decreases exponentially for larger thresholds. This demonstrates the trade-off between security and transaction confirmation speed in Blockchain systems: a larger n will enhance the system's security, but at the expense of a significant lag in transaction finalization. Hence, a moderately large confirmation threshold, like $n = 6$ for Bitcoin and $n = 5$ for Ethereum (in the implementation we used), chosen in accordance with the delays involved, is necessary in practice.

VI. Conclusion

We developed a stochastic model for the evolution and dynamics of Blockchain networks. Our model captures important Blockchain characteristics such as the number of miners, their hashing power (block completion rates), block dissemination delays among distributed nodes, and block confirmation rules. We leveraged analytical techniques for evaluating the impact of the block dissemination delay on key performance metrics, such as the transaction processing rate as measured by the Blockchain growth, and the integrity of the Blockchain as measured by the probability of a successful attack. Extensive experimental results demonstrate that our model accurately predicts crucial system properties and performance characteristics, and we hope that the analysis provides deeper understanding of the dynamic behavior of Blockchain networks.

In future work we will examine in more detail the various system relationships and trade-offs, and how they may be exploited to achieve given performance and integrity targets. This is potentially useful for building future and private Blockchains.

References

[1] L. Bahack (2013). Theoretical Bitcoin attacks with less than half of the computational power. https://arxiv.org/pdf/1312.7013.pdf

[2] T. Bamert, C. Decker, L. Elsen, S. Welten, R. Wattenhofer (2013). Have a snack, pay with Bitcoin. In: *Proc. 13th IEEE Int. Conf. Peer-to-Peer Comput.*

[3] M. Berman, J.S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, I. Seskar (2014). GENI: A federated testbed for innovative network experiments, *Computer Networks* **61**, 5–23.

[4] Cap: https://coinmarketcap.com/

[5] K. Christidis, M. Devetsikiotis (2016). Blockchains and smart contracts for the Internet-of-Things. *IEEE Access* **4**, 2292–2303.

[6] C. Decker, R. Wattenhofer (2013). Information propagation in the Bitcoin network. In: *Proc. 13th IEEE Int. Conf. Peer-to-Peer Comput.*

[7] Ethereum Network Status, https://ethstats.net/, Accessed: 07/27/2017.

[8] I. Eyal, E.G. Sirer (2013). Majority is not enough: Bitcoin mining is vulnerable. In: *Financial Cryptography and Data Security*. Springer, 436–454.

[9] GENI: Global Environment for Networking Innovation, http://www.geni.net, Accessed: 07/27/2017.

[10] "Geth", https://github.com/ethereum/go-ethereum/, Accessed: 07/27/2017.

[11] J. Göbel, H.P. Keeler, A.E. Krzesinski, P.G. Taylor (2016). Bitcoin Blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Perf. Eval.* **104**, 23–41.

[12] G.O. Karame, E. Androulaki, S, Capkun (2012). Two bitcoins at the price of one? Double-spending attacks on fast payments in Bitcoin. In: *Proc. Conf. Comp. Commun. Security*.

[13] D. Kraft (2016). Difficulty control for Blockchain-based consensus systems. *Peer-to-Peer Netw. Appl.* **9**, 397–413.

[14] L.A. Linn, M.B. Koo (2016). Blockchain for health data and its potential use in health IT and health care related Research. Use of Blockchain for Healthcare and Research Workshop.

[15] S. Nakamoto (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

[16] M. Rosenfeld (2014). Analysis of hashrate-based double-spending. https://arxiv.org/pdf/1402.2009.pdf

[17] Y. Sompolinsky, A. Zohar (2013). Accelerating Bitcoin's transaction processing. Fast money grows on trees, not chains. https://eprint.iacr.org/2013/881.pdf

[18] G. Wood (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151. http://gavwood.com/paper.pdf