Securely Boosting Chain Growth and Confirmation Speed in PoW Blockchains

Ovia Seshadri*, Vinay J Ribeiro† and Aditya Kumar*
*Department of Computer Science and Engineering, Indian Institute of Technology Delhi
†Department of Computer Science and Engineering, Indian Institute of Technology Bombay
{ovia.seshadri@cse, aditya.kumar@alumni}.iitd.ac.in, vinayr@iitb.ac.in

2021 IEEE International Conference on Blockchain (Blockchain) | 978-1-6654-1760-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BLOCKCHAIN53845.2021.00028

***Abstract*—Proof-of-Work (PoW) blockchains add blocks, and consequently the chain weight, randomly. The blocks added also have a significant network delay owing to their large size. Large delay combined with randomness causes forks that are responsible for many security problems. One can reduce fork occurrences by designing a system with large block intervals and size but this design compromises performance aspects such as confirmation time guarantees. The trade-off between security and performance in PoW blockchain is a well discussed topic in the literature.**

In this paper, we aim to reduce the conflict between security and performance through our novel concept of *Links*. Links are small, fast and frequent structures that can be incorporated on any new or existing PoW blockchains. Links help reduce the confirmation time of its underlying blockchain while preserving its consistency security guarantees.

Our novel lower-bound growth rate for PoW systems with links in the presence of a general adversary in a partially synchronous network, shows that links provide a more steady chain growth rate than classic PoW systems without links. On a well-established, secure system like Bitcoin, we use the derived growth rate to show that when links are incorporated to Bitcoin, they help reduce its confirmation time from 60 minutes by half to 30 minutes, while retaining the consistency threshold guarantees of the original Bitcoin system. We provide a proof of concept emulation of links in a Bitcoin test bed consisting of 210 nodes having real world latencies between them. We benchmark a system with links against Bitcoin to show links cause negligible network overheads and no compromises on security in terms of orphaned weight due to forks. Our theoretical analyses and experiments can easily be extended to other Nakamoto style PoW systems.

***Index Terms*—Blockchain, Security, Confirmation Time, Consistency, PoW, Double Spend, Forks, Links, Bitcoin**

I. Introduction

In this paper, we consider proof-of-work(PoW), permissionless blockchains such as Bitcoin, Ethereum, Litecoin etc. [1]–[4]. Each block appended to these blockchains, effectively adds PoW 'weight' to that chain based on how difficult the PoW puzzle for the block was to mine. Weight accumulated on a chain directly translates to a more secure system. An honest functioning of the blockchain dictates that a miner chooses the heaviest chain known to extend, in case of competing chains. For simplicity, we can assume all blocks have the same difficulty of mining, hence, have the same weight.

Blocks created by PoW have two drawbacks. Firstly, PoW consensus generates blocks randomly. This randomness is explained as a Poisson random process in several prior works [1], [5]–[7]. Thus, weight gets added to the blockchain at random instants. Secondly, blocks contain transactions and are large in size. Works such as Decker et. al. [8] have shown that delay of blocks increases with increase in block size. Later in this paper, we make similar conclusions for our network model (Section II) and emulation test bed (Section V). This large delay of blocks with their random arrival causes natural forks in blockchain systems. Forks are an ambiguous state of the system where there are multiple chains (called branches), having the same weight, appended to the same block. Natural forks are those formed via block randomness and delays without any adversary influence.

Forks are detrimental to the security of blockchain systems. They keep the system in a divided, uncertain state where the honest miners are not in consensus about the chain to extend which is a window of opportunity for an adversary to execute attacks [9]–[16]. The eventual resolution of a fork (when a single branch of a fork becomes heavier than the others which usually takes one block interval) leads to the orphaning of the other branch(es) and consequently the weight they had acquired. It leads to financial loss for the miners whose blocks were orphaned as their block reward is forgone and mining effort is wasted for all miners trying to extend the orphaned chain. This is a dissipation in mining power utilization where honest mining effort is not translated into securing the system. A common work around is to try to add the orphaned block weight back into securing the chain at later blocks which creates other security vulnerabilities [17]–[21].

Forks become frequent events when block interval is small i.e. at lower PoW puzzle difficulty. Earlier studies [10], [11], [22], [23] show the decline of security when block interval and size are lowered. We make similar claims in our evaluation of the behavior of forks and orphaned weight with respect to lowered block interval and size in Section V.

To meet a security threshold in terms of consistency and orphaned weight due to forks, the mean block interval and mean block size of the system can be increased. At larger block intervals, the time taken for a transaction to enter a block (liveness) and for it to eventually be accepted as permanent (confirmation time) significantly increases which is undesirable for practical reasons and makes PoW blockchains unsuitable for fast payments. Therefore, PoW blockchains must decide on a trade-off [22], [23] between a secure design with large block interval and size that offers better consistency and lower orphaned weight versus a performance oriented design with small block interval and size that provides better confirmation time guarantees.

Our Contributions: In this paper, we propose *Links* (§II),

978-1-6654-1760-0/21/$31.00 ©2021 IEEE
DOI 10.1109/Blockchain53845.2021.00028

which are small, frequent and fast structures having PoW weight that can be implemented on any new or existing PoW blockchain system without causing significant CPU, latency or bandwidth overheads to the underlying blockchain. Links help its underlying blockchain by (i) Maintaining chain stability by steadily adding PoW weight, aiding faster chain growth that helps mitigate double spend and selfish mining attacks. We derive in § III a rigorous lower bound for chain growth and show that it is higher than growth established in PoW blockchain systems from [6]; and (ii) Securely improving the block confirmation time. Using the derived growth rate, we provide strong theoretical guarantees for the confirmation time reduction possible with Links (§ IV-A) and prove that consistency is preserved in a system with Links (§ IV-B). We show the feasibility of incorporating links on PoW blockchains via an *emulation of Links in a reference Bitcoin network consisting of 210 nodes having real world latencies* set between them in Section V. We test latency overheads in the presence of Links and conduct experiments to benchmark their forking behavior in terms of weight orphaned.

II. Links

In this Section we define Links and discuss their mechanism in a PoW blockchain system. Later, we define a network, system and adversary model based on which we will construct theoretical guarantees on security in a system with links in the course of the paper. Note that at times we relate our discussion with Bitcoin, the most familiar PoW blockchain system, for ease of understanding but the idea can be translated to any PoW blockchain system.

A. Links

Definition II.1. *A **Link** is a small bit string consisting of (i) hash of its parent (which may be a block or another link) in the chain, and (ii) a solution of a PoW puzzle that is different and easier than that of a block. It optionally contains information such as a coinbase transaction/an address of the entity creating it which can be used to reward its creator. It contains no transactions other than the coinbase.*

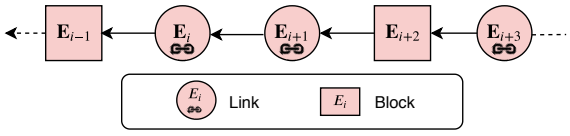

Fig. 1. Sample blockchain with links

Structure: Links can be incorporated on any PoW blockchain system. A link is similar to the block of its system in terms of structure and its generation. Henceforth, we collectively refer to a block or a link as an *entity*. The blockchain system with links consists of a chain of entities where either entity may point to the previous block/link as depicted in Figure 1 as a chain of entities $\mathbf{E}_i$. In a system like Bitcoin using links, links would be entities consisting of an 80 byte header exactly like the block header and an optional coinbase transaction in its body which can be used to identify the creator. With reference to Figure 2, the fields of relevance in the link header (that are also present in the block header) are: the hash of the previous block header which behaves as the pointer to the parent block or link; the agreed difficulty of the PoW puzzle on which a block should be generated; a nonce field which can be tweaked in order to find the valid PoW solution for block or link; and a merkle root of transactions held in the body at the time of its generation. Unlike links, a block additionally has several transactions in its body which makes them much larger in size.

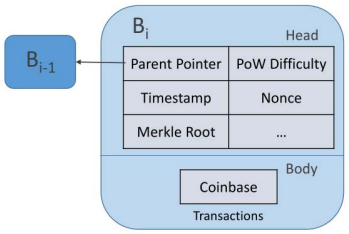

Fig. 2. Sample block contents (B_i is the i^{th} block in the chain). Links have the same structure as block except the transactions in its body.

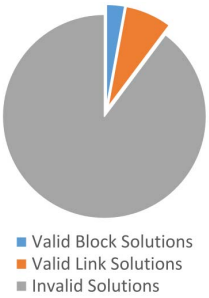

Fig. 3. PoW solutions Space - the range of all header hash values possible

Links being fixed small size structures, have *low broadcast latency*. To see why, consider the following **delay model** for latency of a message of size M to travel from peer i to peer j. Suppose the bandwidth and speed of light delay between the peers are $C_{i,j}$ and $D_{i,j}$ respectively, then the total time to transfer and process a message M from peer i to peer j is: $D_{i,j} + \frac{M}{C_{i,j}} + \kappa M$. In this model the *first bit* of the message takes $D_{i,j}$ to travel across the network at the speed of light and then it takes a further $\frac{M}{C_{i,j}}$ time for the entire message to reach peer j. After this, peer j validates and verifies it. We assume that the verification time is proportional to the size of the message M with proportionality constant κ. Now the total time to broadcast will be the sum of such delays on the peer-level path from the origin to the last peer that receives the message. This model is consistent with [8], [24]. Due to their small size, links' broadcast latency consists mainly of speed of light delays, whereas for blocks the last two terms can dominate. Therefore, we can assume propagation delay of links is less than that of blocks. Entity size vs delay is also demonstrated in our emulation in Section V. Before we discuss links further, let us define a chain extension protocol based on chain weight for miners to extend the blockchain.

Chain Weight: In this paper, we assume miners extend the latest entity on the *heaviest chain* known called **Heaviest Chain Rule** or HCR. The weight of an entity is proportional to the difficulty of their generation, meaning links weigh less than blocks. The heaviness or total weight of the blockchain would be the sum of the weights of all the blocks and links along the chain. We use the following notation to formally define chain weight. Consider a miner who has received a blockchain of N entities consisting of blocks and links. Call the i^{th} entity $\mathbf{E}_i$ ($i = 1, \ldots, N$) and let Θ_i denote the weight of $\mathbf{E}_i$. The total weight of the chain can be written as $\sum_i \Theta_i$. If links were l times easier to generate than blocks (consequently, links arrive

l times more frequent), links weigh l times less i.e. $\frac{1}{l}$ units. Thus, there are frequent links with small weights contributing towards *steady chain weight gain*. Therefore, in our system, $\Theta_i \in \{\alpha, 1\}$ based on it being a link or a block i.e. each link contributes $\alpha = \frac{1}{l}$ units of weight to the chain and every block contributes 1 unit. Parameter l can be seen in many ways such as the frequency of arrivals of links to blocks in the system, as ratio of the target sizes of links to block or as the ratio of weight contributed by blocks to links. They all have the same meaning. In this paper we fix $0 < \alpha \leq 1$ i.e. $l \geq 1$.

Link Generation and Processing: Every miner participates in a PoW election to win a chance to add the next block. While competing in this election, a valid PoW solution may either be a block or a link. For example, in hash-based PoW blockchains like Bitcoin, every miner varies the nonce value of the header and generates a header hash. She checks if this hash falls in a predetermined solution space for a valid block. Figure 3 represents this solution space division. If unsuccessful, she checks if it is a valid link based on a larger non intersecting solution space agreed by the community. This ensures minimum computational mining effort for links. It is similar to the 2-for-1 PoW technique used in [24]–[26]. In case of a valid link, she appends the link on her chain as the new chain tip and updates her chain weight. She then publishes the header and its coinbase as a valid link along with the merkle hash of the transaction tree excluding the coinbase. This merkle hash allows another miner to verify that the coinbase is indeed part of the header that was received based on the merkle root value in the header. Mining proceeds on top of the new link. Note that some values in the link header such as the PoW difficulty and the coinbase were originally intended for the next block and since a trial solution became a link these fields could not be updated. In a valid link, the coinbase helps identify the link's creator for rewarding purposes and will not be added to the UTXO set. When a miner receives a link she can use HCR to determine the chain to extend. The absence of transaction makes them verifiable in O(1) time. The cost of a hard fork to deploy Links in classic PoW blockchains like Bitcoin, is greatly compensated by the significant improvements in the confirmation speeds it offers along with security guarantees.

Rewards: We suggest rewarding links proportional to their weight to the address provided in the coinbase i.e. *link_reward* $= \alpha *$ *block_reward*. The advantage of rewarding links is that even smaller miners reap rewards more regularly than in blockchains without links. This can possibly obviate the need to join mining pools [27]–[30] for small payments. A full discussion of a rewarding scheme for a rational miner is left as future work.

We now define a system model and describe adversary limits which will be used to derive theoretical security limits in systems with links.

B. System and Adversary Model

Let there be a hash-based PoW blockchain system comprised of n miners, connected in a p2p network similar to Bitcoin. We will assume the network is reliable i.e. honest miners see all entities created by each other. Without loss of generality, we will assume the miners control equal mining power. Let an adversary control at most a fraction q of the total mining power of the network where $q < 0.5$.

The analyses in this paper assumes an execution of the blockchain in rounds that model time steps similar to [6], [7], [31]. Let the maximum bounded network delay of blocks and links be Δ_b and Δ_l time rounds respectively. Based on the delay model explained, we can assume $\Delta_b > \Delta_l$. An impartial oracle determines at each time round, with probability $np(1 + \frac{1}{\alpha})$ the successful mining of a valid entity. Parameter p is the hardness for the proof of work and is set such that a block is expected to be mined in $cn\Delta_b$ attempts i.e. $p = \frac{1}{cn\Delta_b}$. Here, c is the expected number of block delays before the next block in mined (block interval in terms of Δ_b). These notations are consistent with [6], [31].

We shall denote the event of the oracle accepting a entity in a time round as a *hit*. Upon a hit, the oracle decides whether it is a block or link with probabilities $\frac{1}{1+\alpha}$ and $\frac{\alpha}{1+\alpha}$ respectively. It then chooses a miner at random to extend their chain with that entity to broadcast in that time round. This simulates PoW and the chosen miner in any round is assumed to have generated that entity.

Honest miners will always be working on the heaviest chain known to them. If a node is controlled by the adversary, they can be working on any entity in the tree of entities from genesis known to them and choose when to broadcast their entities. We can assume at any time the oracle knows which nodes are controlled by the adversary. In other words, the probability of a hit of an honest block in a round is $G = (1 - q)np$ and probability of honest link in a round is $\frac{G}{\alpha}$. Let C_i^r denote the heaviest chain at node i in round r and let $|C_i^r|$ denote its weight.

Selection of link frequency parameter (l): In a system with links, higher frequency of parameter l gives more steady addition of weight but also may lead to fork formations. So we want l to be large, but not so large that it causes too many forks. We try to leverage this trade-off with the following back of the envelope analysis of forks caused by consecutive blocks and consecutive links. The probability of consecutive arrivals of two links forming a fork is $\left[1-\left(1-\frac{G}{\alpha}\right)^{\Delta_l}\right]$, which should be less than the probability of two consecutive arrivals of blocks forming a fork which has the probability $\left[1 - (1 - G)^{\Delta_b}\right]$. These can be approximated to $G\Delta_b$ for blocks and $\frac{G\Delta_l}{\alpha}$ for links. Forks caused by two consecutive links is controlled by the parameter l such that the chance of a link being orphaned is less than that of a block i.e. $G\Delta_b > \frac{G\Delta_l}{\alpha}$. Therefore, the beneficial range of l should be between 1 and $\frac{\Delta_b}{\Delta_l}$. We will estimate this value in our test bed and discuss the behavior of parameter l in more detail in Section V.

III. Chain Growth with Links

In this section, we establish a rigorous lower bound honest growth rate for a system with links in the presence of a general

adversary in a partially synchronous network based on the adversary model defined in Section II. Chain growth (υ) is loosely defined as the minimum growth in chain weight for any honest miner in a time round. Growth for classic PoW systems with the Nakamoto consensus protocol in a partially synchronous was first studied by Pass et.al. [6] who identified $\upsilon_{\text{classic}} = \frac{G}{G\Delta_b+1}$ as lower bound honest growth in a round. Growth rate is an important security metric for blockchain systems as many attacks, such as the whole group of private chain mining attacks, depend on the adversary's ability to generate a heavier chain than any honest chain in a given window of time. The growth metric limits what the adversary can achieve in any window if $\upsilon > \beta$ where β represents the expected adversary growth in unit time. Modeling links on the same chain as blocks is a non trivial extension to [6] as it requires including two types of entities having different weight, frequencies and propagation times apart from looking at the system in terms of weight as opposed to length as considered in [6].

The following lemma shows that if some honest player has a chain of weight w at round r, then all other honest players will have a chain of weight at least w by round $r+\Delta_b$. Given $\Delta_b > \Delta_l$, the proof is an obvious one.

Lemma III.1. *In a blockchain with links, if nodes i, j are honest, then $|C_j^{r+\Delta_b}| \geq |C_i^r|$.*

Theorem III.2. *(Chain Growth) For any $0 < \delta < 1$ and any interval $[s, s+t]$ where $t > 2\Delta_b$ rounds, system with links achieves an honest chain growth of at least $(1-\delta)\upsilon t$ in weight except with probability $e^{-\Omega(\delta^2\upsilon t)}$ which is negligible in t and honest growth rate parameter per round is,*

$$\upsilon = G\left[\frac{1}{G\Delta_b+1} + \frac{\max\left(0, 1-\frac{2G\Delta_b}{G\Delta_b+1}\right)\alpha}{G\Delta_b+\alpha}\right]$$

Proof. Consider the arrivals of blocks and links in the interval of interest, $[s, s+t]$, from the perspective of any honest node. In this interval we will mark certain blocks and links in such a way that the spacing between consecutive marked entities is larger than Δ_b. To ensure that only the blocks that have reached all the other honest nodes during the interval of interest is marked, we remove the first and last Δ_b rounds in this interval. Therefore, the interval of interest is now $[s+\Delta_b, s+t-\Delta_b]$.

The marking procedure for blocks is as follows. Mark the first honest block, skip Δ_b rounds and mark the next honest block and so on. Let N_b be the total number of marked blocks in $[s+\Delta_b, s+t-\Delta_b]$. We next mark honest links in between every pair of consecutive marked blocks in a similar fashion as we marked blocks in the interval $[s, s+t]$. Say marked block B_i arrived at t_i and the next marked block B_{i+1} arrived at t_{i+1}. To ensure that no previous marked entity is in transmission over the network before marking a new entity, we remove the first and last Δ_b rounds in the interval $[t_i, t_{i+1}]$. Let interval I_i, where $i = 1 \ldots N_b - 1$, denote the time available to mark links between marked blocks B_i and B_{i+1} after removing the first and last Δ_b rounds. We define I_i as

$$I_i = \begin{cases} [t_i + \Delta_b, t_{i+1} - \Delta_b], & \text{if } t_{i+1} - t_i > 2\Delta_b \\ \phi, & \text{otherwise} \end{cases}$$

Within I_i we mark the first honest link, skip ahead by Δ_b, mark the next honest link, and so on. Let N_l be the total number of marked links in the interval $[s+\Delta_b, s+t-\Delta_b]$.

Note that this construction ensures a gap of at least Δ_b between any two consecutive marked entities (say E_{j-1} and E_j) even if E_{j-1} is a link. This guarantees that all ancestors of a marked entity are known to all honest miners before marking that entity. Let the marked entities be E_j created by miners M_j at time rounds T_j where $j = 1, ..(N_b + N_l)$. Note that the same miner may generate multiple marked entities, hence we allow $M_j = M_k$ for $j \neq k$. Let $\theta_j \in \{\alpha, 1\}$ denote the weight of the entity E_j. Let the weight of the heaviest chain of miner M_j at time T_j after the addition of E_j be represented as $|C_{M_j}^{T_j}|$.

At $T_{j-1} + \Delta_b$ for any j (we skip Δ_b after every marked entity), miner M_j must have seen E_{j-1} and all its ancestors, before mining E_j. From Lemma III.1, even if E_{j-1} is not on M_j's heaviest chain, she must have extended a chain at least as heavy as $|C_{M_{j-1}}^{T_{j-1}}|$ when creating E_j. E_j further added θ_j units to M_j's heaviest chain. Therefore, we can say,

$$|C_{M_j}^{T_j}| \geq \theta_j + |C_{M_{j-1}}^{T_{j-1}}| \tag{1}$$

At $T_j + \Delta_b$ for any j, any honest miner m would have seen E_j and must have chosen to extend the heaviest chain known to her. From Lemma III.1 the chain she extends would be at least as heavy as M_j's heaviest chain at T_j i.e
$|C_{M_j}^{T_j}| \leq |C_m^{T_j+\Delta_b}|$.
This can be written as $\theta_j + |C_{M_{j-1}}^{T_{j-1}}| \leq |C_m^{T_j+\Delta_b}|$ from eq 1.

The interval $[s, T_{N_b+N_l} + \Delta_b]$ is completely enclosed in the interval $[s, s+t]$ as we have a given a pad of Δ_b at the end of $[s, s+t]$. Therefore, the weight gained by m in $[s, T_{N_b+N_l} + \Delta_b]$ must be less than or equal to the weight gained by m in $[s, s+t]$. We take the difference in weight gain by miner m in $[s, T_{N_b+N_l} + \Delta_b]$ to find the lower bound weight gained in the interval $[s, s+t]$.

$$\begin{aligned} |C_m^{s+t}| - |C_m^s| &\geq |C_m^{T_{N_b+N_l}+\Delta_b}| - |C_m^s| \\ &\geq \theta_{N_b+N_l} + |C_{M_{N_b+N_L}-1}^{T_{N_b+N_l}-1}| - |C_m^s| \end{aligned}$$

Note that based on the reasoning in given for equation 1, $|C_{M_1}^{T_1}| \geq \theta_1 + |C_{M_1}^{s+\Delta_b}|$. From Lemma III.1 we know $|C_{M_1}^{s+\Delta_b}| \geq |C_m^s|$. Recursively simplifying we get, $|C_m^{s+t}| - |C_m^s| \geq \sum_j \theta_j \geq N_b + \alpha N_l$.
By estimating N_b and N_l we obtain, with high probability in t, a lower bound on the honest chain growth in t.

N_b **estimation:** The marked blocks form an arrival process with i.i.d. inter-arrival times, each of which is equal to a Geometric random variable (with mean $\frac{1}{G}$) plus Δ_b. Define $t' = t - 2\Delta_b$ and $\overline{N_b} := \frac{Gt'}{G\Delta_b+1}$. Let e_1 denote the event that

$N_b \in \left[(1-\delta_1)\overline{N_b}, (1+\delta_1)\overline{N_b}\right]$ for some $\delta_1 \in (0,1)$. Then using the chernoff bound from Lemma VIII.1 in appendix, $Pr[e_1] \geq 1 - e^{-\Omega_1(t')} = 1 - e^{-\Omega_2(t)}$.

N_l **estimation conditioned on** e_1**:** Suppose N_b blocks have been marked. Call the total time remaining for marking links t''. Then $t'' \geq t' - 2\Delta_b N_b$, since each marked block essentially reduces the time available for marking links by at most $2\Delta_b$. When e_1 occurs, we must have

$$t'' \geq \tau := \max\left(0, t' - (1+\delta_1)(2\Delta_b\overline{N_b})\right) \tag{2}$$

To obtain N_l, we first consider an arrival process in time τ whose inter-arrival times are i.i.d., each equal to the sum of a Geometric random variable (with mean $\frac{\alpha}{G}$) and Δ_b. Then the number of arrivals of this process is dominated from above by the distribution of N_l since $t'' \geq \tau$. Let e_2 be the event that $N_l \geq \frac{(1-\delta_2)G\tau}{G\Delta_b+\alpha}$ for some $\delta_2 \in (0,1)$. Then from equation (2) and Lemma VIII.1(in appendix) we have $Pr[e_2|e_1] \geq 1 - e^{-\Omega_3(\tau)} = 1 - e^{\Omega_4(t)}$.

Applying Bayes' theorem, we have $Pr[e_1 \cap e_2] = Pr[e_2|e_1]Pr[e_1] \geq \left(1 - e^{-\Omega_4(t)}\right)\left(1 - e^{-\Omega_2(t)}\right) = 1 - e^{-\Omega(t)}$. This implies that w.h.p. in t', the event $e_1 \cap e_2$ occurs, that is, N_l and N_b are simultaneously greater than $\frac{(1-\delta_2)G\tau}{G\Delta_b+\alpha}$ and $\frac{(1-\delta_1)Gt'}{G\Delta_b+1}$ respectively. This gives us the desired growth in t' rounds as $\upsilon t' = \frac{Gt'}{G\Delta_b+1} + \frac{G\tau\alpha}{G\Delta_b+\alpha}$. □

Note 1: Pass et. al. [6] identified $\upsilon_{\text{classic}} = \frac{G}{G\Delta_b+1}$ for classic PoW systems. We can notice $\upsilon_{\text{classic}} \leq \upsilon$ as the first term in υ is $\upsilon_{\text{classic}}$.

Note 2: We can assume the expected adversary growth per round is $\beta = 2qnp$ as they function as a single entity.

Note 3: The lower bound growth rate derived here is conservative as the fact that link have lesser network delay ($\Delta_l < \Delta_b$) is not exploited. Therefore, it is possible to further increase υ by loosening the interval (I_i) for marking links in the above proof. This will be explored in a future extension of this paper.

IV. Confirmation Time (CT) and Consistency

Confirmation Time (CT) is the time a seller must wait to believe, with enough confidence, that an adversary (holding a q fraction of power) cannot orphan a block containing a transaction of interest. CT is a practical measure of usability of a blockchain system. Lower CT allows the system to be used for fast payments and is a step towards making blockchains an alternative to centralized payment networks such as VISA. We find the confirmation time (CT) reductions possible through links via an analysis of the double spend attack which has been traditionally accepted [1], [5] as the method to find CT.

Later in this section we discuss the consistency threshold in a system with links. Consistency is an important security property that tell us the maximum adversary power at which an honest block, that is accepted by all honest peers and has stayed on their chain for at least t rounds, will (with high confidence) continue to remain on honest chains forever even under influence of an attack. Consistency can be defined as ensuring with overwhelming probability increasing with t, that at any two rounds r and s such that $r < s$, the chains of any honest players i, j at r and s respectively i.e. C_i^r and C_j^s, have at least one common entity created after round $r - t$.

CT is a measure of time defined for a fixed adversary power (q) and a probability of success of a specific double spend attack. Consistency, on the other hand, is a bound that tries to find the maximum adversary power under which the system is shielded from any type of adversary attack.

A. Double Spend Attack

Nakamoto [1] and Rosenfeld [5] originally studied the double spend attack in a synchronous network model. Their result stated that sellers should wait till there were 6 blocks, from the time their transaction enters a block, to confirm payment. We calls this the **number of confirmation blocks** denoted by k. In Bitcoin, $k = 6$ translates to a mean 60 minutes waiting time. We show that a Bitcoin system with links reduces this CT by half to 30 minutes for any adversary power. We start with an analysis of CT in zero-delay model like that of [5] and factor in delay of blocks and links using the growth result from Theorem III.2.

In a *Double Spend Attack*, miners are divided into honest and Byzantine. Consider a blockchain without links formed of blocks B_i where $i = 0, 1, 2 \ldots$. A Byzantine adversary with q fraction of total hashing power wants to double-spend some money already spent in a transaction in block B_{i+1} in the publicly accepted heaviest chain. Assume that the recipient of that money waits for k blocks (the CT), that is till B_{i+k}, to be confident that the transaction will be in the chain with high probability. To do a double spend, the attacker includes a double-spend in a privately mined chain forked from B_i and releases it any time after B_{i+k} is generated, provided it is at least as long as the public chain at the time of release. Note that the adversary has focused all their mining power on the growth of only their private chain from B_i in this attack.

Factoring Delay: Recall our system and adversary model from section II. To factor in block and link delays and to allow a most direct and impactful comparison of results with [5], we adjust the honest and adversary mining power in this analysis such that it approximates the partially synchronous network honest growth rate derived in Theorem III.2. This way the adversary gains an advantage over its actual mining power q while the honest miner power is restricted to a value less than $1-q$. For the rest of our analysis we will assume the adversary holds a fraction q' such that $q < q' < 0.5$. Accordingly, the revised honest mining power becomes $1-q'$ which is less than $1-q$. In a synchronous network we expect the ratio of growth of the honest chain weight to the adversary chain weight in unit time to be equal to the ratio of their mining powers. Thus, to estimate q', we find growth rates υ and β using q such that $\frac{\upsilon}{\beta} = \frac{1-q'}{q'}$. We compare the CT results of using links against Bitcoin by substituting the following values to calculate honest (υ) and adversary (β) growth rates: $c = 60$, $\Delta_b = 10^{13}$ and $q < 0.5$ which is consistent with [6] to represent the Bitcoin

system. Using the new higher fraction of mining power for the adversary, q', we perform the analysis similar to [5]. By changing the parameters c, Δ_b we can compare any other PoW blockchain system.

Assumptions and Notations: We fix $k = 6$ given by [1], [5]. Note that any $k > 6$ will only improve the security guarantee. This analysis can be easily extended to any value of k. The process of entity generation is modeled as Poisson distributions similar to [5] with different intensities depending on q' and α. We have the probability of success of the attack by an adversary holding q' fraction of power as, $\mathbb{P}_{q'}(\text{success}_{\text{adv}}) = \sum_z P(z)Q(z)$ where $P(z)$ is the probability that the honest chain is ahead by z in weight over the attacker's chain at the time of confirmation i.e. at the arrival of B_{i+k} and $Q(z)$ is the probability of the attacker catching up to the honest chain given that the attacker's chain differs by z in weight. We study the $\mathbb{P}_{q'}(\text{success}_{\text{adv}})$ after $k = 6$ honest confirmations in two systems - Bitcoin and systems with links. Figure 4 plots the result for Bitcoin and systems with links at $l = 2$.

Computation of $P(z)$: Computation of $P(z)$ for apriori adversary success with links can be modeled as a sequencing problem of ordering the arrivals of $k = 6$ honest blocks, H honest links, D adversary blocks and R adversary links where $z = ((k + H\alpha) - (D + R\alpha))$ and the last arrival is the k^{th} honest block i.e.

$$P(z) = \sum_{H,D,R} \binom{N}{H}\binom{N-H}{D}\binom{N-H-D}{k-1} P_b^k P_h^H P_d^D P_r^R$$

over a wide range of H, D, R such that $z = (k+H\alpha)-(D+R\alpha)$ and a wide range of z is covered such that $\sum_z P(z)$ is almost 1. Here $N = H+D+R+k-1$ signifies the number of arrivals to sequence, the last arrival being an honest block. Given the arrival of an entity, P_b, P_h, P_d and P_r represent the probabilities, of arrivals of an honest block, honest link, adversary block and adversary link respectively. They are equal to $\frac{(1-q')\alpha}{(1+\alpha)}, \frac{(1-q')}{(1+\alpha)}, \frac{q'\alpha}{(1+\alpha)}$ and $\frac{q'}{(1+\alpha)}$ respectively. The computations are done for a wide range of H,D and R such that $\sum_z P(z)$ is almost 1.

Computation of $Q(z)$: $Q(z)$ is the probability that the attacker will catch up to the honest chain given the honest chain is ahead by z. It follows that $Q(z) = 1$ when the attackers chain is longer that the honest chain. An upper bound to $Q(z)$ can be estimated by modeling this problem as a random walk starting from z to 0, over all z with 4 jump possibilities (± 1, and $\pm\alpha$) due to occurrence of two types of arrivals - links and blocks. This can be calculated from a well known result [32]. Calculation for classic blockchains is similar to Rosenfeld [5] i.e. $Q(z) = q^z$ for $z > 0$ and 1 otherwise.

Observations: Rosenfeld [5] showed that at $q = 10\%$ the chance of double spend with $k = 6$ is $0.6 * 10^{-3}$. By the inclusion of links at just $l = 2$, the probability of adversary success plunges down to an insignificant $0.12 * 10^{-5}$ at the same $q = 10\%$ and $k = 6$. However, adverasary power of 10% is no longer realistic. The Bitcoin hashrate distribution [33] shows the largest mining pool size varying between 19 to 23%. At $q = 25\%$, we still see 2 orders of magnitude difference between links and without links as shown in Figure 4. Intuitively we see that confirmation time can reduce significantly with the help of links. To know by how much, we simulate the probability of success for $k = \{2, 3, 4, 5\}$ with links at $l = 2$ and compare it to $k = 6$ without links. In Figure 4, we see that the plot line $k = 3$ (with links) is below the line $k = 6$ (without links). Any $l > 2$ gives better results at $k = 3$. Therefore we conclude, links are capable of reducing the number of confirmation blocks to at least $k = 3$ at the same level of security as $k = 6$ without links.

B. Consistency

Consistency bound for PoW systems was first defined by Garay et.al. [34] in a synchorous model and was extended to partially synchrounous model by Pass et.al. [6]. It was later improved by [31], [35]. Recent works by Gazi et. al. [7] and Dembo et. al. [36] made a breakthrough for tight consistency bounds in PoW blockchains that follow the Nakamoto consensus. They proved that the 51% attack is the worst possible attack. A 51% attack is when the adversary gains control of more than 50% of a network's mining power. Attackers with majority control of the network can interrupt the recording of new blocks. They would also be able to reverse transactions that were confirmed, meaning they could double-spend coins. The 51% attack scenario is rare, largely because of the logistics, hardware and costs required to carry one out.

This implies that as long as the honest growth rate is higher than adversary growth rate in unit time i.e. $\upsilon > \beta$, consistency is always maintained. This result applies directly for links implemented on a system like Bitcoin as it is a PoW blockchain following the Nakamoto consensus. With the help of the growth result in general adversary model defined in theorem III.2, we show the lower bound consistency threshold in systems with links in Figure 5.

We numerically compute the maximum threshold of q at typical block intervals c, used in practice under which consistency holds i.e. $\upsilon > \beta$. For instance Bitcoin can be studied at $c = 60$ [6], [31] and Ethereum is represented at $c = 5$ [37]. We evaluate this for $n = 10^5$, $\Delta_b = 10^{13}$ and $l = 1$ to 5 shown in Figure 5. The comparison made by Figure 5 is between tight consistency threshold of classic PoW chains [7], [36] and a lower bound consistency in systems with links. This is because the growth used for links is conservative. However, at $c = 60$ which translates to a mean 10 min block inter arrival time as set in Bitcoin, we can see the adversary threshold q is at least 49.5% in all values of l tested. In classic PoW systems, the state of the art analysis by [7], [36], tolerates q of up to 49.57% at $c = 60$. At $c = 5$ which translates to a mean 20 secs block inter arrival time as set in Ethereum, the state of the art analysis tolerates an adversary threshold q is 47.5% while a system with links tolerates 45.7% and 43% at $l = 1, 5$ respectively. With a tightened growth we expect

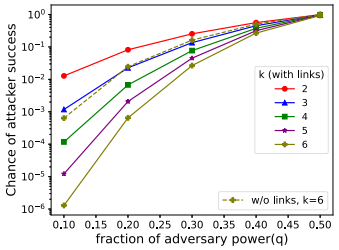

Fig. 4. CT comparison at frequency $l = 2$ at various block confirmations (k) of systems with and without links

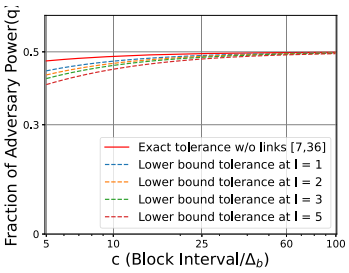

Fig. 5. Maximum adversary tolerance at various block intervals between classic PoW systems and system with links

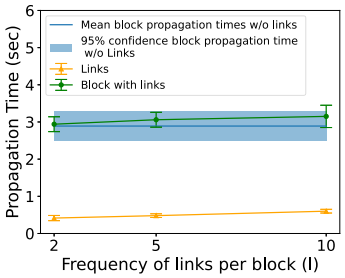

Fig. 6. Delay of ≈1MB blocks with and without the presence of links

links to close the threshold difference as in classic systems for $c < 60$. This study is left as future work.

This result shows that PoW blockchains with links has similar consistency lower-bounds as classic PoW blockchains at higher $c \geq 60$ and differs by at most a factor of 8% at $c < 60$ from the analysis of [7], [36]. Thus, we can conclude links preserve the consistency of the underlying classic blockchains they are implemented on.

V. Experimental Evaluation

In this section, we present experimental results using a reference implementation of a Bitcoin system that incorporates links. We benchmark this implementation against classic Bitcoin, in a set of experiments which test: (1) delay of blocks and links, (2) orphaned weight in classic PoW blockchains at varying block size and interval, (3) orphaned weight in classic PoW blockchain vs links at a fixed block size and interval that offers the optimal orphaned weight threshold (found from (2)) and (4) fork resolution times in classic PoW blockchain vs links. We also demonstrate the practical link frequency (l) upper bound. We do not consider a link reward structure as it is inconsequential to our experimental hypotheses. Security guarantees against a general adversary is proved in Section IV and hence adversary presence is not considered in our emulations.

Modifications made to Bitcoin: We use Bitcoin Core client v0.16 [38] as a base and add support for links. This includes a new link message type and a single new RPC - generatelink, which creates a valid link on the current tip of the chain before broadcasting it. This RPC is modeled closely after the existing generate call, which provides similar functionality for blocks. The only other change was adding methods to process links upon receiving them i.e. updating the chain weight and invoking HCR.

Network overlay structure: Our experiments were conducted on a 210 Bitcoin node testbed, running on a cluster of 40 machines (8 GB RAM, 4-8 CPU cores). Similar to other works that emulate Bitcoin [39], we construct a connected topology where each node is connected to 4-8 other nodes, chosen uniformly at random. The network set-up phase also includes adding additional delays between every pair of connected nodes. In order to mirror real world conditions, we introduce delays to mimic real world latencies between nodes. A group of nodes represents a major city in the world, and each connection between two groups has corresponding inter city latency taken from [40].

A. Observations on Propagation Delays

Propagation time (or delay) of any message is the difference between the time of its broadcast at the source node and the time of its arrival at the destination. To remove extraneous outliers, we calculate the 95th percentile of propagation times similar to other works [8] that study delays in network emulations. All experiments were conducted with the default compact block relay in Bitcoin. All experiments on propagation delays were conducted on standard average block interval of 600 secs as set in Bitcoin. Nodes are also given the responsibility of generating their own transactions every few seconds. This is to ensure healthy traffic during the emulation and non-empty mempools.

We conducted the experiments in 4 configuration modes: a system without links ($l = 0$), and systems with link frequencies per block as l = 2, 5 and 10. Every run for each configuration was scheduled for 120 minutes, creating a blockchain of 12 blocks on average. Results are discussed over 30 runs in each mode.

Effect of block delay with and without links: In our implementation, we see that all links have a fixed size of 264 bytes which includes a block header with a coinbase transaction. The mean block size observed in runs involving unmodified blocks size in our experiments is 1.18 MB. A maximum delay of 0.57 secs was observed for links across all values of l considered. Figure 6 plots the mean propagation time of blocks with and without links in unmodified bitcoin. Blocks of ≈1.18 MB had 2.89 sec delay on average. Therefore, links are at least 5 times faster than blocks in a system incorporating both entities. Our hypothesis that a link will propagate faster than a block is verified here. We can see that the propagation times of blocks with the inclusion of links increases only by a factor of 0.02, 0.06 and 0.09 for l = 2, 5 and 10 respectively in our network. Thus, links work with the system without creating drastic bandwidth or latency overheads.

Propagation delay vs Block sizes: True to our hypothesis, the network delay for blocks increases linearly with increase in block size. The results are shown in Figure 7. We observed that links are at least faster by factor of 3 for 100 KB blocks, 5 for unmodified bitcoin blocks(≈1.18MB) and 10 for 8MB blocks. Thus, links propagate faster than all block sizes considered.

B. Observations on Orphaned Weight

We are interested in observing the ratio of the orphaned weight to the total weight of any chain generated over a sufficiently long blockchain. Orphaned weight is a more meaningful metric to judge the behavior of forks in a blockchain. This is because, it estimates the differences in chain growth, mining rewards and mining power utilization caused as opposed to simply counting the number of forks in systems dealing with entities of different sizes and arrival rates. In this section we compare the orphaned weight of classic blockchains with varied block size and interval at a fixed transaction throughput. Based on our inference we compare orphaned weight of classic blockchain with a system incorporating links. We also discuss the time taken for the resolution of forks in the systems considered.

In Bitcoin, the block interval time is set particularly high to make forks rare events and so studying orphaned weight or fork resolution times is difficult. We could not observe any forks in the 600 secs block interval setting runs for any value of l in our short experimental runs. Therefore, in order to get good estimates of forking behavior in short experiments, we scale down the block interval from 600 seconds to 30 seconds, so that natural forks occur based only on network delays in every run. This reduction mimics lower block interval systems such as Ethereum [2]. This scaling did not affect any entity propagation delays. The link generation is set with the Poisson mean interval as $\frac{30}{l}$ secs. Every run for each configuration was scheduled for 30 minutes, with number of blocks and links per run varying based on inter-arrival times.

Varying block size and interval at a fixed throughput: When studying the behavior of orphaned weight ratio in classic blockchain systems at various block sizes and intervals for a fixed transaction throughput, we found that the orphaned weight percentage increases with decreasing size and interval. This is shown in Figure 8. This is because, even at small size and interval, blocks have random arrivals and larger network delay owing to their transactions and thus creating more forks. Each fork orphans a large weight. It logically follows that larger sizes (>1MB) and interval have a better orphaned weight tolerance, however, such systems will have a trade off in transaction liveness and confirmation time. Thus, we will not consider such systems in this study.

Classic blockchain vs links and Fork Prevention: Since the classic bitcoin like blockchain at 1.18 MB blocks size showed the least orphaned weight ratio, we will consider it an tolerable orphaned weight ratio and study if links meets this threshold. With reference to Figure 9, we see that links at $l < 5$ differ with the established orphaned weight threshold by less than 1%. Links were able to meet this threshold despite having higher number of unique forks created.

This phenomenon occurs due to two important reasons. Firstly, due to links having less PoW weight, at every fork caused by two near-simultaneously generated links, links orphan only a fraction of the weight that two blocks causing a fork would have. Secondly, and more interestingly, links have the ability to prevent forks caused by two blocks. We consider *fork prevention* as a scenario, occurring at each node, where a link arrives on a block earlier than a sibling block that would otherwise create a fork in the chain, preventing the fork before it could occur. Therefore, although links create forks that resolve faster, they also help prevent some forks caused by blocks that would have orphaned higher weight.

Resolution Time of Forks (RTF): RTF is calculated by every miner as the difference between the time a fork was realized on her chain and the time at which the next block or link increased the chain weight on one of the branches of the fork, effectively resolving it. We observe that links reduce the mean fork resolution time by 53, 78 and 91 percent, with l = 2, 5 and 10 respectively compared to classic bitcoin ($l = 0$). Figure 10 shows the results. Forks are resolved at each miner, based on the chain she observes locally. We found that with links, for every fork, the branch chosen by the miner to resolve it was the same across the network. Thus links succeed in reuniting mining power (divided across the branches of a fork) to the same chain. This is true across all link frequencies. Thus, links work as a signaling scheme to give early insights to miners about the hashing power division in the network during forks that helps in faster fork resolutions.

Evaluation of link frequency parameter (l): In Section II, we saw that the beneficial range of l should be between 1 and $\frac{\Delta_b}{\Delta_l}$. Our propagation experiments on our emulation shows that $3 < \frac{\Delta_b}{\Delta_l} < 10$ for wide range of block sizes considered. At $\frac{\Delta_b}{\Delta_l} = 5$ which was observed for bitcoin like ≈1MB blocks, we showed that CT is reduced by half and consistency is preserved for $l = 2$ in Section IV-A. Our orphaned weight experiments also show that system with links at $l = 2$ lowers the orphaned weight ratio of classic PoW system with bitcoin like ≈1MB blocks. This suggests that PoW blockchains with links has similar security as classic PoW blockchains and possibly better lower-bounds when $\frac{\Delta_b}{\Delta_l}$ is large and $l \ll \frac{\Delta_b}{\Delta_l}$. Thus links succeed in reducing CT while preserving the security gurantees of the underlying blockchain.

We can hereby conclude that, for a fixed transaction throughput, the system with links i.e. large blocks and small links, is a better design than reduced block sizes in frequent intervals in terms of forks created and weight orphaned. Links are frequent and have low network delay, hence they resolve forks faster and have same orphaned weight when $l < 5$ as compared a classic Bitcoin (≈ 1 MB block size with 2.89 secs delay) system.

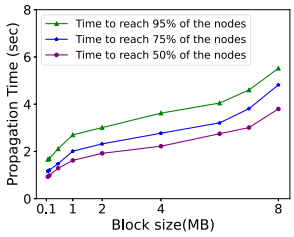

Fig. 7. Propagation delay of various block sizes

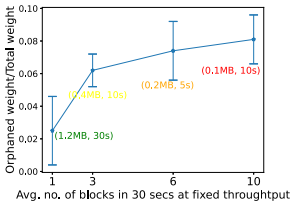

Fig. 8. Orphaned weight over classic PoW blockchains with varied blocks (size, intervals) at a fixed transaction throughput

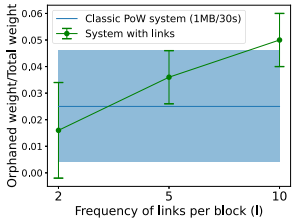

Fig. 9. Orphaned weight between classic PoW system, system with links

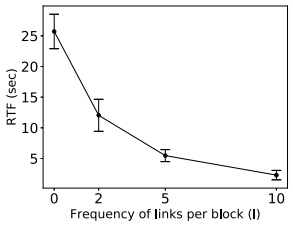

Fig. 10. Mean resolution time of forks in system without links (l=0) and with links (l=2,5,10)

VI. Related Work

We divide and discuss the related work in two categories: (1) Works that achieve faster growth and confirmations in classic systems by reducing their block intervals and size, and (2) Works that significantly alter their blockchain structure to improve confirmation speeds or other performance/security issues in PoW blockchains,

Works that reduce block interval and size like [2], [19]–[21], [23], [41] to achieve better growth and confirmation speed have a severe security trade-off as shown in previous sections. We have shown via emulations and formal proofs that links achieve the same result without hampering security. Reducing block interval and size induces forks and hence orphans a large weight reducing mining power utilisation as shown in Section V. In contrast, incorporating links without changing block interval and size does not orphan any additional weight over the underlying blockchain without links would. These solutions that reduce block size and interval resort to either including orphaned blocks deviating from a linear chain to a DAG or they make use of the GHOST protocol [17] which includes orphaned weight in an effort to resolve forks quickly. Links and GHOST are non-intersecting solutions which can be implemented together.

Some works such as [24], [42] offer a multi-blockchain system designed to address the scalability problem of high CT and low transaction throughput without reducing block intervals or size. They decouple the transaction ordering and weight confirmation roles of blocks and suggest using different structures for each role. Unlike Links, such solutions that fundamentally change the blockchain structure are hard to incorporate into existing, popular PoW systems to improve their security or performance. Other works [25], [39], [43], [44] use weaker blocks (blocks created at low PoW) for storing transactions aiming to partly solve the transaction throughput problem or fix mining variance issues. We show(§V) how transactions make the weak blocks heavy and increase their propagation times. Links are light and hence can be used as a fast signaling mechanism to regularly identify the heaviest chain, thus, resolving forks faster and even preventing forks caused by blocks.

Although we discussed Links with respect to classic PoW blockchain systems in this paper, we believe links can also be extended to benefit a wide range of blockchains that implement GHOST and non-linear (hierarchical, DAG based or multi chain) PoW blockchains suggested in literature as long as they can apply the HCR rule and support weaker PoW solutions. Analysis of the benefits of such implementations are left as future work.

VII. Conclusion and Future Work

We have introduced a novel scheme called Links, that allows steady growth in chain weight and better confirmation time guarantees to PoW blockchain system they are implemented on. Links provide benefits to these systems without compromising consistency guarantees.

Based on our security analysis, Links have the potential to mitigate a broad genre of advanced selfish-mining attacks such as [12], [13], [15] and not just the ones explicitly analyzed in this work. We have showed that consistency is preserved for systems like Bitcoin that have high block intervals. By tightening the lower-bound honest growth rate we believe the consistency result can be extended to all block intervals. We leave this to a future extension of this paper. The concept behind Links is not limited to single chain PoW blockchains. It can be extended to DAG-based or multi-chain blockchains as long as they can apply the HCR rule and support weaker PoW solutions. Analysis of the benefits of such implementations are left as future work.

Acknowledgment

This work was supported by the Visvesvaraya Scheme by Meity, GOI. The authors would like to thank Shadab Zafar for his valuable contributions in the early stages of this work. We also thank Subodh Sharma, Himanshu Gandhi, Aditya Ahuja, Dilpreet Kaur and Sandeep Kumar for their insightful reviews on the early draft of the paper.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[2] V. Buterin and Others, "Ethereum white paper, 2014," *URL https://github.com/ethereum/wiki/wiki/White-Paper*, 2013.
[3] "Litecoin - open source p2p digital currency," https://litecoin.org/, (Accessed on 05/16/2019).
[4] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," *July 7th*, 2013.
[5] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.

[6] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2017, pp. 643–673.

[7] P. Gaži, A. Kiayias, and A. Russell, "Tight Consistency Bounds for Bitcoin," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 819–838.

[8] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*. IEEE, 2013, pp. 1–10.

[9] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 1, p. 2, 2015.

[10] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[11] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 515–532.

[12] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 305–320.

[13] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 154–167.

[14] C. Natoli and V. Gramoli, "The balance attack against proof-of-work blockchains: The R3 testbed as an example," *arXiv preprint arXiv:1612.09426*, 2016.

[15] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse Attacks on Bitcoin's Peer-to-Peer Network." in *USENIX Security Symposium*, 2015, pp. 129–144.

[16] K. Liao and J. Katz, "Incentivizing double-spend collusion in bitcoin," in *Financial Cryptography Bitcoin Workshop*, 2017.

[17] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 507–527.

[18] A. Kiayias and G. Panagiotakos, "On Trees, Chains and Fast Transactions in the Blockchain." *IACR Cryptology ePrint Archive*, vol. 2016, p. 545, 2016.

[19] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "SPECTRE: A Fast and Scalable Cryptocurrency Protocol." *IACR Cryptology ePrint Archive*, vol. 2016, p. 1159, 2016.

[20] Y. Sompolinsky and A. Zohar, "Phantom: A scalable blockdag protocol," 2018.

[21] C. Li, P. Li, D. Zhou, Z. Yang, M. Wu, G. Yang, W. Xu, F. Long, and A. C.-C. Yao, "A decentralized blockchain with high throughput and fast confirmation," in *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, 2020, pp. 515–528.

[22] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 3–16.

[23] A. Kiayias and G. Panagiotakos, "Speed-Security Tradeoffs in Blockchain Protocols." *IACR Cryptology ePrint Archive*, vol. 2015, p. 1019, 2015.

[24] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the Blockchain to Approach Physical Limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: ACM, 2019, pp. 585–602. [Online]. Available: http://doi.acm.org/10.1145/3319535.3363213

[25] R. Pass and E. Shi, "FruitChains: A Fair Blockchain." in *Acmccs*, vol. 2016, 2017, pp. 1–26.

[26] P. Szalachowski, I. Homoliak, and S. Sun, "StrongChain: Transparent and Collaborative Proof-of-Work Consensus," *arXiv preprint arXiv:1905.09655*, 2019.

[27] 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 *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 104–121.

[28] I. Eyal, "The miner's dilemma," in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 89–103.

[29] "Bitcointalk: Mining cartel attack," 2010. [Online]. Available: https://bitcointalk.org/index.php?topic=2227.msg29606{\#}msg29606

[30] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '15. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927.

[31] L. Kiffer, R. Rajaraman, and Others, "A better method to analyze blockchain consistency," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM CCS18, 2018, pp. 729–744.

[32] R. Gallager, "6.262 Discrete stochastic processes, Spring 2011 Chapter 7: Random Walks, Large Deviations, and Martingales," *Massachusetts Institute of Technology: MIT OpenCourseWare) http://ocw. mit. edu (Accessed 23 Mar, 2013). License: Creative Commons BY-NC-SA*, 2011.

[33] "Bitcoin hashrate distribution - blockchain.info," https://www.blockchain.com/pools?, (Accessed on 11/12/2019).

[34] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 281–310.

[35] L. Ren, "Analysis of nakamoto consensus." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 943, 2019.

[36] A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and Nakamoto always wins," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 859–878.

[37] N. Awathare, S. Das, V. J. Ribeiro, and U. Bellur, "Renoir: Accelerating blockchain validation using state caching," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2021, pp. 9–20.

[38] "bitcoin project," https://github.com/bitcoin/tree/v0.16.0, 2018.

[39] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol." in *NSDI*, 2016, pp. 45–59.

[40] "Global Ping Statistics - WonderNetwork," https://wondernetwork.com/pings, 2019.

[41] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 528–547.

[42] H. Yu, I. Nikolic, R. Hou, and P. Saxena, "OHIE: blockchain scaling made simple," *arXiv preprint arXiv:1811.12628*, 2018.

[43] P. R. Rizun, "Subchains: A technique to scale bitcoin and improve the user experience," *Ledger*, vol. 1, pp. 38–52, 2016.

[44] A. Zamyatin, N. Stifter, P. Schindler, E. R. Weippl, and W. J. Knottenbelt, "Flux: Revisiting Near Blocks for Proof-of-Work Blockchains." *IACR Cryptology ePrint Archive*, vol. 2018, p. 415, 2018.

APPENDICES

VIII. CHERNOFF BOUNDS FOR CHAIN GROWTH

We now state a lemma that is proved by Lemma 6.4 in [6] to give the probability of tail bounds when dealing with intervals in the form of a random variable plus a constant. Although Lemma 6.4 of [6] proves only the lower tail of lemma VIII.1, a simple extension of the proof using the Chernoff bound in the Appendix A of the same paper gives the upper tail stated. This is used to prove growth in theorem III.2.

Lemma VIII.1. *Consider a discrete-time arrival process in time interval of size t in which the inter-arrival times are i.i.d. and the i^{th} inter-arrival time Y_i $(i = 1, 2, \ldots)$ is of the form $X_i + \Delta$, where X_i is a geometric random variable with mean $\frac{1}{\mu}$ and $\Delta > 0$ is a constant. Define $\gamma = \frac{\mu}{1+\mu\Delta}$. Denoting the total number of arrivals in the interval by Z_t, we have, for any $0 < \delta < 1$, (1)Upper Tail: $P\big(Z_t > (1+\delta)\gamma t\big) < e^{-\Omega(\delta^2\gamma t)}$; (2) Lower Tail: $P\big(Z_t < (1-\delta)\gamma t\big) < e^{-\Omega(\delta^2\gamma t)}$.*