HOW TO BUILD A BLOCKCHAIN: THE ASYNCHRONOUS COMPOSITION MODEL

PARTHA DEY* AND ADITYA GOPALAN†

*Department of Mathematics, University of Illinois Urbana-Champaign; psdey@illinois.edu
†Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign; gopalan6@illinois.edu

Abstract. Inspired by blockchains, in [3], we introduce a dynamically growing model of rooted Directed Acyclic Graphs (DAGs), the *asynchronous composition model*, subject to random delays with finite mean. The new block at time t is connected to blocks chosen from the graph $G_{\max(0,t-1-\xi_t)}$ according to a *blockchain rule* f. Here ξ_t is the random delay at time t and the graph is updated by taking union with the graph G_{t-1}. This process corresponds to adding new blocks to a blockchain, where delays arise due to network communication. The main question of interest is the end structure of the *asynchronous limit* of the graph sequence as time increases to infinity. We consider the *Nakamoto* rule f_{Nak}, in which a vertex is uniformly selected from those furthest from the root, and the blockchain rules f_1, f_2, where in f_k a random set of k leaves is chosen without replacement. We explain that the asynchronous limits for f_{Nak} and f_2 are one-ended, while the asynchronous limit for f_1 is not one-ended, almost surely. We also state growth properties of the longest path for the sequence of graphs for f_{Nak}.

1. Introduction

The key ingredients to implementing a blockchain protocol are the underlying peer-to-peer network and the choice of *block attachment rule*. Nodes in the network create blocks. Each block contains a hash reference to one or more previous blocks. Nodes then share these blocks amongst one another. This paper aims to motivate an asymptotic requirement called *one-endedness* for the block attachment rule. We expand on the intuition of one-endedness and give a precise characterization.

One can view the blockchain as a rooted graph. In Bitcoin, the root block is referred to as the *genesis block*. We refer to it as the root block for consistency with the scientific literature. Here blocks are vertices, and hash references are directed edges. In the most well-known blockchain protocols, such as Bitcoin and Ethereum, blocks are attached via the *Nakamoto Rule* [8]. Here the new block is attached to a single block of maximum distance from the root block. In other protocols, such as the Iota protocol, blocks are attached via a more complicated randomized rule [10]. We will show that both of these rules have the one-endedness property.

Agreement and security in blockchain systems are typically treated as limiting statements about the blockchain graph. One salient example is the common "six blocks deep" tenet for Bitcoin transactions, with the acknowledgment that for any $d > 6$, a user can have increasing *confidence* in a block which is "d blocks deep." We make the term *confidence* more precise in Section 2.2. Formally, blockchain system builders and users desire guarantees such as "the probability that a particular block will never be overturned given the current state of the blockchain is at most $\varepsilon > 0$." Such statements take a *limit into the future*.

This paper is an extended abstract of [3].

The idea of a limit into the future poses several difficulties. Even with the Nakamoto Rule, the most straightforward block attachment rule, how should one choose the value of ε? Should ε depend on the application context for a particular blockchain system? If so, how?

Furthermore, the peer-to-peer networks supporting large blockchains may suffer unbounded communication delays between network nodes [5]. There is a long line of research into the security and performance of blockchains using the Nakamoto Rule, based on [9]. However, [9], and those that follow, use bounded end-to-end delays. This is inconsistent with how peer-to-peer networks are typically analyzed and how the Bitcoin network is implemented [5]. The paper [11] studies the Nakamoto Rule's security under unbounded delays. However, the results do not generalize to other block attachment rules. These lead to difficulty in computing the probability of overtaking. Indeed, such probabilities have only been computed for network models, which are far more straightforward than what is currently implemented in Bitcoin [2]. Moreover, the supporting peer-to-peer network may change over time regarding network size and connectivity structure, rendering such a computation meaningless.

Furthermore, with growing interest in blockchain protocols that build more general block DAGs, it is not clear how to appropriately generalize the notions of security and agreement in the form of a limit into the future. Without the Nakamoto Rule, there is no notion of "overtaking," which is the critical tenet for the limits into the future for protocols like Bitcoin and Ethereum. It is a challenge to appropriately define block confirmation for this more general setting.

Protocols using block attachment rules other than the Nakamoto Rule are otherwise identical in implementation to Nakamoto Rule blockchain protocols. This suggests that the "limit into the future" approach for analyzing and designing blockchain protocols has significant shortcomings. We address these shortcomings using a different consideration of the limit as time tends to infinity.

In an appropriate sense, though finite at all times, the blockchain graph tends to be a limiting infinite graph as time tends to infinity. In this limit graph, the definition of a confirmed block can be given independently of the choice of block attachment rule. We say that a block B in the limiting graph is *confirmed* if, for all but finitely many blocks, there is a path from that block to the root block which passes through B. Thus, if a block is confirmed in the limit, a user can have complete confidence in its contents being agreed upon by all network nodes running the protocol and hence, its security.

Intuitively, this notion of confirmed blocks motivates the idea of a "unique" infinite chain of blocks resulting from an ideal trajectory of a Nakamoto Rule blockchain system. The precise sense of uniqueness is not essential to understand the results in this paper; see [3] for more details. However, the existence of such an infinite chain of confirmed blocks is not *a priori* guaranteed. Thus, we impose an additional requirement, called *one-endedness*, on the block attachment rule. One-endedness is a topological property of a graph that says that it "only grows to infinity in one direction." It ensures the existence of infinitely many confirmed blocks and, for the Nakamoto Rule, the existence of the "unique" infinite chain of confirmed blocks. This paper does not focus on the dynamics of the peer-to-peer network. We refer the interested reader to [6, 7].

1.1. **Asynchronous Composition Model.** We discuss the Asynchronous Composition Model (ACM) of [3] and its use for determining the one-endedness of limiting blockchain graphs. In particular, we highlight the use of the ACM in determining the one-endedness of limits for the Nakamoto rule and the randomized Rule used by the Iota protocol [10]. While blockchain systems consist of network dynamics arising from a peer-to-peer communication network and a growth process for the blockchain graph, the ACM isolates the blockchain graph growth process. This enables a blockchain system designer to check their block attachment rule in isolation from other complicating phenomena arising from the analysis of peer-to-peer networks.

We denote the block attachment rule by f. Here f is a function that takes as its input a blockchain graph and outputs a set of blocks (potentially random) from the input graph to which the new block connects via directed edges. The Asynchronous Composition Model evolves in discrete time, with one new block added at each time slot.

(i) We denote the blockchain graph at time t by G_t. The blockchain grows according to an iterative process, where a single block is added at each time slot. We label the root block by 0, and the block, added in the k-th time slot by k.
(ii) Due to delays, a random number of blocks, say ξ_t, may be missing at the node that creates the t-th block. Thus, the function f, rather than being applied to G_{t-1}, is applied to $G_{\max(t-1-\xi_t,0)}$.
(iii) The graph G_t is given by the graph G_{t-1}, along with the new block and edges from the new block to the set of blocks in $f(G_{\max(t-1-\xi_t,0)})$.

To simplify our analysis, we assume that the delays seen at each time step, $(\xi_t)_{t\geqslant 1}$ are independent and identically distributed. The precise mathematical model can be found in [3].

1.2. **Block Attachment Rules of Primary Interest.** We are primarily interested in the following block attachment rules.

(a) f_{Nak} is the Rule from the Nakamoto protocol, which attaches a new vertex to one vertex chosen uniformly from those at maximum distance from the root.
(b) f_1 attaches the new vertex to a leaf vertex chosen uniformly at random.
(c) f_2 is the Rule used in the Iota protocol: the new vertex is attached to a uniformly selected pair of leaf vertices if possible; otherwise, it is attached to the unique leaf vertex via a single edge.

2. One-Endedness

We say that a graph is a *blockchain graph* if it is a rooted directed acyclic graph (DAG) such that any vertex has a directed path to the root, and the root has out-degree zero. When the context is clear, we use the term *graph* to refer to a *blockchain graph*.

2.1. **Some Motivating Examples.** We begin with some motivating examples, which help to explain the concept of one-endedness and its relation to the blockchain.

First, we consider two different block attachment rules, the Nakamoto and the f_1 Rule. Recall that the Nakamoto Rule chooses one vertex at random from those at the maximum distance from the root. The f_1 Rule chooses one leaf vertex uniformly at random.

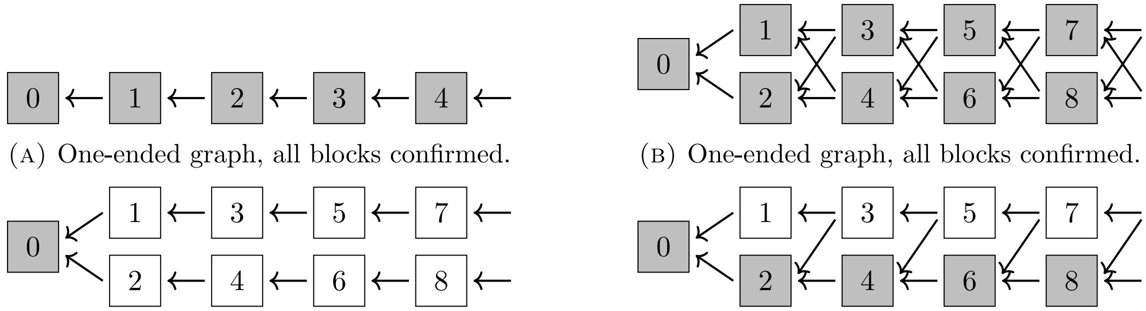
Intuitively, under non-zero delay, the Nakamoto Rule produces a "unique" infinite chain of blocks, while the f_1 Rule seems to produce a blockchain with several chains that grow independently. The fact that several chains are growing for the f_1 Rule indicates that there may not be infinitely many confirmed vertices. In contrast, the fact that there is a "unique" infinite chain for the Nakamoto rule indicates that there are infinitely many confirmed blocks. In Figure 1, we depict several example blockchain graphs under the assumption of zero network delay.

2.2. **One-Endedness and Blockchains.** In an infinite blockchain graph, a *ray* is an infinite path that ends at the root. We say that two rays r_1 and r_2 are *equivalent* if there is a third ray r so that there are infinitely many vertices that lie on both r_1 and r, and there are infinitely many vertices that lie on both r_2 and r. We say the graph is *one-ended* if any two rays are equivalent [4].

We state a result relating one-endedness to blockchains; see [3, 6].

Lemma 2.1. *Let G be an infinite blockchain graph.*

(a) *If G is one-ended, then G contains infinitely many confirmed vertices.*
(b) *If G contains infinitely many confirmed vertices, then G has a one-ended sub-blockchain graph that contains all confirmed vertices.*
(c) *If G is a tree, then G is one-ended if and only if it contains infinitely many confirmed vertices.*



(A) One-ended graph, all blocks confirmed.

(B) One-ended graph, all blocks confirmed.

(C) Two-ended graph, only block 0 confirmed.

(D) Two-ended graph, even numbered blocks confirmed.

FIGURE 1. Examples to motivate confirmed (gray) blocks and one-endedness.

Let us return to the pictorial examples in Figure 1. We see that Figures 1a and 1b are one-ended, while Figures 1c and 1d are not. Thus, by Lemma 2.1 there are infinitely many confirmed vertices in Figures 1a and 1b. However, there are also infinitely many confirmed vertices in Figure 1d. The block attachment rule here is inefficient. This inefficiency is precisely captured by the second part of Lemma 2.1. Our previous comparison of the Nakamoto Rule and the f_1 Rule is explained by the third part of Lemma 2.1, the limit for the Nakamoto Rule is one-ended, while the limit for the f_1 Rule is not. As a general principle, we desire block attachment rules to confirm every block if there is no network delay.

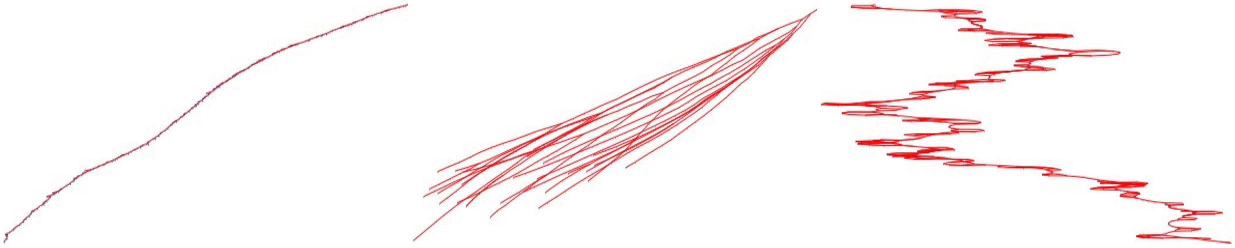
As mentioned above, given a limit graph, a user can be *confident* in a confirmed block. Recall that the existence of such blocks is guaranteed by the one-endedness property of a block attachment rule. In the limit, whether or not a block is confirmed is a deterministic property. Whereas, at any finite time, the probability of eventual confirmation is generally challenging to analyze.

3. Results on One-Endedness

We now present the main results on one-endedness for the three block-attachment rules of primary interest from Section 1.2.

Theorem 3.1 (See [3, Theorem 2.16, 2.17, and 2.18])**.** *We have the following results for i.i.d. delays with finite mean, almost surely.*

(a) *The limit graph for the Nakamoto Rule is one-ended.*
(b) *The limit graph for the f_1 Rule is not one-ended.*
(c) *The limit graph for the f_2 Rule is one-ended.*



(A) Nakamoto Rule

(B) f_1 Rule

(C) f_2 Rule

FIGURE 2. Kamada-Kawai spring layout examples for the Nakamoto, f_1, and f_2 Rules with Geometric(1/2) delays.

We give a simulated example of Theorem 3.1 in Figure 2. As mentioned before, Theorems 3.1 and Lemma 2.1 explain the critical difference in the intuition between the Nakamoto and f_1 block attachment rules. For the Nakamoto rule, the blockchain tends to grow a “single” longest chain. Thus, it has infinitely many confirmed blocks. In contrast, f_1 has multiple chains that grow to infinite length, resulting in only finitely many confirmed blocks. Our analysis of the one-endedness for the f_2 Rule in Theorem 3.1 is the first such analysis for any widely used protocol which does not use the Nakamoto Rule.

3.1. **Idea behind the Proofs of Results on One-Endedness.**

We provide some intuition to Theorem 3.1 on one-endedness. First, we note that the delay process can be decoupled from the blockchain graph process. In particular, we study the *time delay graph*, with vertices given by the non-negative integer times, and directed edges of the form $t \to \max(0, t-1-\xi_t)$. For more details, see [1, 3].

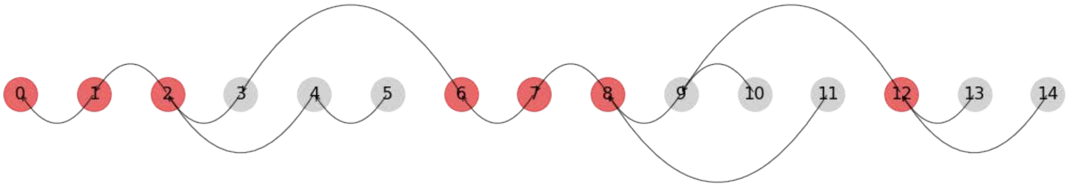


FIGURE 3. A sample time delay graph with regeneration times marked in red.

A sample delay graph is shown in Figure 3. Notice in Figure 3, that any time $T \in \{0,1,2,6,7,8,12\}$ have the property that for all times $t > T$, we have that $t - \xi_t > T$. We refer to such times as *regeneration times.* Regeneration times correspond to moments when the underlying peer-to-peer network is synchronized, *i.e.,* when all nodes have the entire blockchain history up to the current time. At such times, it is clear that the blockchain growth process “restarts,” albeit with a *genesis graph* instead of a *genesis block*.

It turns out that regeneration times frequently occur because the expected inter-regeneration time is finite, and the probability of no delay is positive. Furthermore, the random blockchain graphs at the regeneration times form a Markov process on an appropriate space of graphs. Our results follow by studying a related Markov chain induced by a specific functional of the graph-valued Markov chain. The functional depends on the block attachment rules, but it represents the number of blocks that are candidates to receive a reference by the (fixed) block attachment rule. The result then follows from the positive recurrence or transience of the induced Markov chain. When the probability of no delay is zero, one can define a notion of “regeneration interval” and extend the analysis. See [3] for more details.

4. GROWTH RATE OF THE LONGEST CHAIN FOR THE NAKAMOTO RULE

For the Nakamoto Rule, we can also address *how* the blockchain process grows. In particular, we compute the rate at which the longest chain grows. Since one block is created at each time step, the growth rate of the longest chain is also the fraction of blocks that lie on the longest chain.

Theorem 4.1 (See [3, Theorem 2.15])**.** *Denote by X_t the length of the longest chain at time t. Let H be an integer-valued random variable such that $\mathbb{P}(H \geqslant k) = \prod_{i=1}^{k} \mathbb{P}(\xi_1 \geqslant i)$. Then,*

$$X_t/t \xrightarrow{a.s.} 1/\mathbb{E}\,H \quad as\ t \to \infty.$$

4.1. **Idea behind the Proof of Theorem 4.1.**

Notice that the length of an interval in which X_t stays constant is independent of any other such interval. It depends only on those i.i.d. delays between the times of the increments. Thus, intuitively, the long-term growth rate is the inverse of the expected length of an interval in which X_t is constant.

Suppose that X_t is incremented at some time T. The next increment time can be easily computed using the independence of the delays, since the length of any such interval is s if and only if $t-\xi_t < s$

for the first $s-1$ times following time T and $(T+s)-\xi_{T+s} \geqslant s$. This is shown pictorially in Figure 4. Using renewal Central Limit Theorem, this analysis can be used to prove a Brownian Motion convergence for the length of the longest chain process [3, Theorem 2.15].

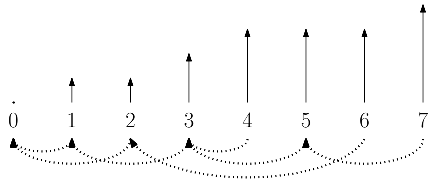


FIGURE 4. Proof idea for Theorem 4.1. Here, the vertical arrows represent the length of the longest chain, while the dotted arrows are the time delay graph.

5. New Directions

This paper identifies the relationship between confirmed blocks in blockchain graphs and one-endedness. We establish that the one-endedness property holds for protocols using the Nakamoto and f_2 rules, such as BITCOIN and IOTA. Moreover, this relationship is universal and does not depend on the choice of block attachment rule. Thus, it is natural to expect that, in addition to one-endedness, there may be a universal notion of blockchain security independent of the block attachment rule. Identifying this connection is a crucial area of future work.

In addition, this paper only analyzes block attachment rules used in popular blockchain protocols. We leave it to future work to determine the appropriate block attachment rules and when to use a certain kind of block attachment rule. These are salient problems in understanding the design principles for blockchain protocols. We hope that, as with one-endedness, they can be addressed in a general fashion instead of a rule-by-rule fashion.

References

[1] François Baccelli and Antonio Sodre. Renewal processes, population dynamics, and unimodular trees. *J. Appl. Probab.*, 56(2):339–357, 2019.

[2] Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Everything is a race and nakamoto always wins. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 859–878, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Partha S. Dey and Aditya Gopalan. On an asymptotic criterion for blockchain design: The asynchronous composition model. *arXiv preprints arXiv:2202.05080*, 2022.

[4] Reinhard Diestel. *Graph theory*, volume 173. Springer, Berlin, fifth edition, 2018.

[5] Giulia Fanti and Pramod Viswanath. Deanonymization in the bitcoin p2p network. *Advances in Neural Information Processing Systems*, 30, 2017.

[6] Aditya Gopalan, Abishek Sankararaman, Anwar Walid, and Sriram Vishwanath. Stability and scalability of blockchain systems. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(2), 2020.

[7] Aditya Gopalan and Alexander Stolyar. Data flow dissemination in a network. *arXiv preprint arXiv:2110.09648*, 2021.

[8] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, 2008.

[9] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.

[10] Serguei Popov. The tangle. *cit. on*, page 131, 2016.

[11] Suryanarayana Sankagiri, Shreyas Gandlur, and Bruce Hajek. The longest-chain protocol under random delays. *arXiv preprint arXiv:2102.00973*, 2021.