Short Paper: On the Claims of Weak Block Synchronization in Bitcoin

Seungjin Baek[1⋆], Hocheol Nam[1⋆], Yongwoo Oh[1],
Muoi Tran[2], and Min Suk Kang[1]

[1] KAIST {seungjinb,hcnam,yongwoo95,minsukk}@kaist.ac.kr
[2] National University of Singapore muoitran@comp.nus.edu.sg

Abstract. Recent Bitcoin attacks [15,17,18] commonly exploit the phenomenon of so-called weak block synchronization in Bitcoin. The attacks use two independently-operated Bitcoin monitors — i.e., `Bitnodes` and a system of customized supernodes — to confirm that block propagation in Bitcoin is surprisingly slow. In particular, `Bitnodes` constantly reports that around 30% of nodes are 3 blocks (or more) behind the blockchain tip and the supernodes show that on average more than 60% of nodes do not receive the latest block even after waiting for 10 minutes. In this paper, we carefully re-evaluate these controversial claims with our own experiments in the live Bitcoin network and show that block propagation in Bitcoin is, in fact, fast enough (e.g., most peers we monitor receive new blocks in about 4 seconds) for its safety property. We identify several limitations and bugs of the two monitors, which have led to these inaccurate claims about the Bitcoin block synchronization. We finally ask several open-ended questions regarding the technical and ethical issues around monitoring blockchain networks.

1 Introduction

Timely propagation of blocks in Bitcoin is critical to ensure its safe consensus operations [10]. Indeed, recent partitioning [15,18] and double-spending [17] attacks against Bitcoin have exploited the phenomenon of so-called *weak block synchronization* — i.e., a large fraction of nodes (e.g., 60%) do not have the up-to-date blockchain even after an extended time (e.g., 10 minutes). This surprisingly slow block propagation is measured and confirmed by two independent sources: (1) `Bitnodes` monitor [20], a long-running and highly-cited third-party Bitcoin network crawler, and (2) `RPC-based` monitor [15], a data collector that interacts with a few Bitcoin supernodes via RPC calls. Recently, Saad et al. [16] further conjecture that weak block synchronization can be possibly caused by the increased network size and churn rate in Bitcoin. Yet, slow block propagation is a controversial claim because several anecdotal evidence from past studies and measurements from other Bitcoin monitors have suggested otherwise. In 2013, Decker et al. [5] report that a new block reaches the majority of peers in less than

⋆ Co-leading authors.

a few tens of seconds. In 2016, a technique called Compact Block Relay [4] was introduced as the default block relaying scheme in Bitcoin to further reduce the block propagation time. Besides, DSN Bitcoin Monitoring [7], a closed-source crawler developed for academic studies, independently reports that in 2021 Bitcoin blocks take only 10 seconds or less to reach 90% of nodes.

In this paper, we carefully evaluate the claims of the weak block synchronization in Bitcoin and attempt to give a more accurate account of the current state of block propagation. In particular, we challenge the accuracy of the two (i.e., `Bitnodes` and `RPC-based`) Bitcoin monitors that have been the main sources of supporting evidence for these claims. We first show that both monitors do not successfully capture the accurate block synchronization status of several live nodes that we deploy and control in the Bitcoin network. Next, we investigate their publicly available codebase and discover a number of problems that may have caused measurement errors. Some of them are architectural limitations; e.g., the polling-based block data collection in the `Bitnodes` monitor always offers outdated block information. Some are protocol-level bugs; e.g., the `RPC-based` monitor mistakenly alters the block propagation of its peers and eventually misses a significant portion of block information. We then conduct large-scale measurements of the block propagation in the Bitcoin network with our fixed `RPC-based` monitor, showing that the network is well-synchronized (e.g., 90% of peers receive new blocks in less than 4 seconds). Lastly, we reconfirm the fast block propagation in a realistic controlled network in which blocks with various sizes (e.g., 0.5–1.6 MB) are propagated through up to 10 hops of globally distributed nodes.

The paper is organized as follows: Section 2 provides the necessary background. Section 3 presents our measurements and analysis on the claims of weak block synchronization. In Section 4, we discuss several future research directions before we conclude the paper in Section 5.

2 Background

In this section, we briefly introduce the block propagation protocol logic in Bitcoin (§2.1) and then describe the high-level operations of `Bitnodes` and `RPC-based` Bitcoin monitors (§2.2).

2.1 Block propagation in Bitcoin

In Bitcoin [13], several thousands of distributed nodes independently validate and store the *blockchain*, a public ledger containing the historical transactions of all users. Transactions are written to the blockchain via a process called mining, in which specialized nodes, commonly known as miners, compete to extend the blockchain by finding a new block that includes validated transactions and the hash of the previous block. Every 10 minutes on average, a miner generates and sends a new block to all other nodes in the system so they can validate it and update their blockchain accordingly. Block data is propagated via a permissionless

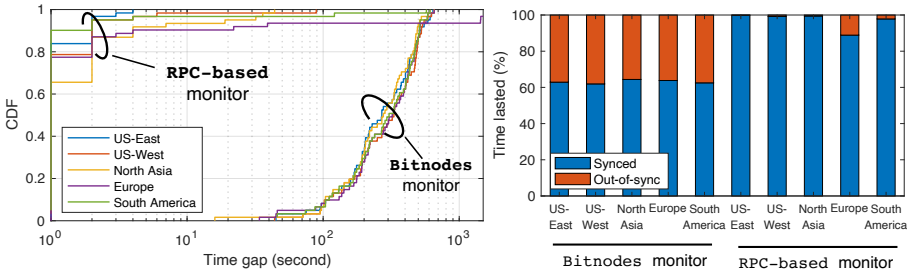
peer-to-peer network between nodes, in which each of them typically establishes up to 10 outgoing connections to *reachable* nodes that have publicly routable IP addresses and accept incoming connections. Upon receiving a new block, nodes validate and relay it immediately to their peers until the entire network is synchronized with the latest block. Since 2016, Bitcoin protocol allows compact block relaying that requires less data transmission and, hence, potentially reduces the block propagation latency [4]. Desired to receive and send block data as fast as possible, some Bitcoin miners are believed to use additional overlay techniques to accelerate their block propagation, such as using a separate block relay network (e.g., FIBRE [8], bloXRoute [2]).

2.2 Bitcoin network monitors

Since measuring the required time for all nodes to receive the latest block is crucial for evaluating the efficiency and safety of the Bitcoin network, there have existed several network monitors in Bitcoin. These network monitors connect to the reachable nodes and monitor their block update information but not unreachable nodes since they do not accept incoming connections. Here, we briefly describe two notable Bitcoin monitors, that is, `Bitnodes` [20], a popular online service, and an `RPC-based` crawler that is recently proposed in a peer-reviewed paper [15]. Among other Bitcoin network monitors (e.g., DSN Bitcoin Monitoring [7], Coin Dance [3]), the `Bitnodes` and `RPC-based` monitors are the only two monitors that have source code available and record the block propagation delay. Recent studies also use the block propagation measurements directly from these two monitors to motivate several new Bitcoin attacks [15,17,18].

`Bitnodes` monitor. `Bitnodes` is a Python-based lightweight crawler [19] designed to estimate the number of reachable Bitcoin nodes. `Bitnodes` operates continuously in rounds approximately every 4 minutes, attempting to establish connections to all reachable nodes. During the connection handshake with the reachable nodes, the `Bitnodes` monitor extracts their latest block heights from their `version` messages. After each round, `Bitnodes` monitor dumps the list of reachable nodes and their block heights into snapshots and publishes them. Recent `Bitnodes` snapshots show that there are usually around 30% of nodes that are 3 blocks (or more) behind the latest blockchain tip.

`RPC-based` monitor. The `RPC-based` monitor is particularly designed to measure the block synchronization performance of Bitcoin [15] and it consists of a data collector and a few supernodes, i.e., Bitcoin clients that increase the connection limit so that they can connect to thousands of reachable IPs concurrently. Periodically, the data collector issues RPC calls to the supernodes to retrieve the block heights of their peers. In particular, the collector uses the `getpeerinfo` RPC call that returns the list of peers connected by a supernode and their `synced_blocks` values indicating their latest block heights known by the supernode. The measurements collected by the `RPC-based` monitor show that only 40% of nodes have the latest block after about 10 minutes [15].



(a) (b)

Fig. 1: Evidence of inaccurate measurements of the two monitors. (a) Cumulative distribution of the time taken by the two monitors to publish the up-to-date blockchain of our five full nodes. (b) Percentage of lasted time of each synchronization status measured by the two monitors.

3 Our Measurements and Analysis

In this section, we evaluate the claims of weak block synchronization in four following steps. First, we present empirical evidence that both `Bitnodes` and `RPC-based` monitors fail to report the synchronization status of our own live Bitcoin nodes promptly (§3.1). Second, we report several limitations and bugs that we found in the two monitors, which have incorrectly led to the slow block propagation conclusions (§3.2). Third, we independently measure and show the fast block propagation in today's Bitcoin network (§3.3). Fourth, we conduct a controlled block propagation experiment to confirm that propagating Bitcoin blocks through multiple (e.g., 10) hops of peers only requires a few seconds of delay (§3.4). Finally, we discuss some ethical considerations of our measurements (§3.5).

3.1 Empirical Evidence of Inaccurate Measurements

To verify the block synchronization reported by the `Bitnodes` and `RPC-based` monitors, we use the ground truth data recorded at our live Bitcoin nodes. We run five Bitcoin Core clients with version 0.21.1 in five geographic regions of Amazon EC2 (i.e., US-East, US-West, South America, Europe, and North Asia) for 12 hours on September 9, 2021. Since the original `RPC-based` monitor [15] is not operating as of this writing, we download and run it too. For each of the 60 blocks our nodes receive in this experiment, we report the exact timestamps when our nodes receive it, the timestamps of the `Bitnodes` snapshots reporting our nodes with the updated height, and the timestamps when the `RPC-based` monitor observes our nodes updating their `synced_blocks` values.

We found that the `Bitnodes` monitor frequently exhibits significant delays in publishing the latest block heights of our nodes. Figure 1a shows that in 50% of cases, the `Bitnodes` monitor takes more than 4 minutes to include the up-to-date block heights of our nodes in a snapshot and the delay can be as high as 10 minutes in some worst cases. The `RPC-based` monitor reports most of the block

heights of our nodes within 10 seconds except a few outliers with one notable case in which the height update of our node in Europe is delayed for 25 minutes. As a result, the `Bitnodes` monitor incorrectly concludes that our nodes are out-of-sync for about 35% of the time while the `RPC-based` monitor incorrectly reports that our node in Europe is out-of-sync for about 10% of the time; see the orange bars in Figure 1b. These incorrect block synchronization measurements of only five nodes suggest that the large-scale measurements (e.g., covering all 10K reachable nodes) made by the `Bitnodes` and `RPC-based` monitors can be seriously misleading.

3.2 Discovered Problems in Two Monitors

We now investigate the publicly available codebase of the `Bitnodes` [19] and `RPC-based` [1] monitors to identify the root causes of their inaccurate block synchronization measurements.

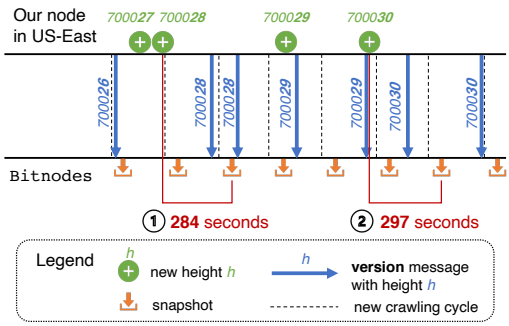


Fig. 2: Block height updates in a Bitcoin full node and the `Bitnodes` monitor.

`Bitnodes` monitor. We identify two inherent limitations of the `Bitnodes` monitor. The first limitation stems from its polling-based monitoring architecture, that is, `Bitnodes` crawls reachable IPs from the Bitcoin network in 4-minute cycles. In each crawling cycle, `Bitnodes` connects to other nodes at random timestamps, records their `version` messages, and exports them into a snapshot when the cycle ends. When a node receives a new block after sending its `version` message in a crawling cycle, it has to wait for the next cycle to update its new block height, which can be up to 8 minutes of delay. We also note an additional delay of at least 30 seconds for exporting a snapshot at the end of each cycle. In Figure 2, we illustrate how the `Bitnodes` monitor is delayed in updating the latest block height of our node in the US-East region in a 30-minute interval. For example, in ①, our node receives block **700028** after notifying `Bitnodes` with a `version` message carrying the height **700026**. Therefore, our node must wait for 284 seconds until its block height of **700028** is reflected in a snapshot. Similarly, in ②, the `Bitnodes` monitor publishes the block height **700030** of our node with 297 seconds of delay.

Another limitation stems from some buggy block height reports frequently observed in the `Bitnodes` snapshots. That is, in all the `Bitnodes` snapshots we analyze, there exist thousands (about 15% of the entire set) of reachable nodes with a zero block height. Interestingly, the vast majority (e.g., 80%) of these nodes are `.onion` addresses, accounting for about 50% of all connected Bitcoin-over-Tor nodes. We separately investigate these nodes with zero block height and

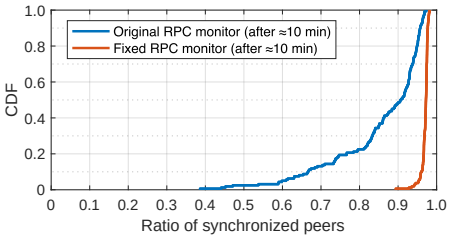


Fig. 3: Cumulative distribution of the ratio of synchronized peers measured by the original and fixed `RPC-based` monitors.

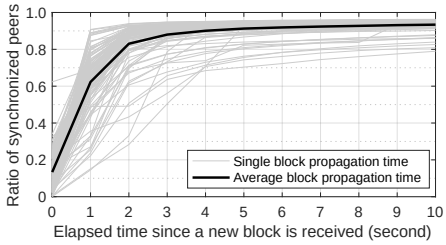


Fig. 4: Synchronization of peers in the first 10 seconds since a new block is received by our monitor. We show 100 blocks starting with height **699860**.

confirm in our experiment (see §3.3) that they are all regularly updated with the latest blockchain. According to the `Bitnodes` source code [19], the block height 0 of a node indicates that either the `version` messages sent by the node are corrupted or the internal database fails to record the actual height. From this, we conjecture that some unreliable interactions between `Bitnodes` database and `.onion` addresses might be the root cause of these nodes with zero block height. We leave further investigations for future work.

`RPC-based` monitor. Unlike the `Bitnodes` monitor, which is deployed to estimate the network size, the `RPC-based` monitor is specifically designed to monitor the block synchronization. Unfortunately, we identify one subtle yet critical problem in it that contaminates its measurement results. In particular, the `RPC-based` monitor mistakenly propagates a new block hash to all other peers that have not relayed it to the monitor. The `synced_blocks` value of a peer is, however, updated *only* when the peer sends a new block hash to the `RPC-based` monitor. When a peer receives a new block hash from the monitor before it sends the same hash to the monitor, it is considered by the `RPC-based` monitor as *unsynchronized* at least until the next block is generated.

To confirm this bug and its impact on the block synchronization measurement, we run two versions of the `RPC-based` monitor for 24 hours and compare their results. The two versions include the original open-source `RPC-based` monitor [1] and a *fixed* `RPC-based` monitor that disables block information forwarding (i.e., preventing `inv`, `headers`, and `cmpctblock` message types from being sent in the `PushMessage` function in `net.cpp` [1]). We also make our best effort to provide the same or improved experiment setup as in the original paper [15], such as issuing `getpeerinfo` calls every second, load balancing the crawling task using 10 servers. Since the exact locations and configurations of the original `RPC-based` monitors are unknown, we reasonably use 20 `t2.xlarge` instances in the US-West region of Amazon EC2 in this experiment. Our original and fixed `RPC-based` monitors connect to about 9.2K and 9.1K reachable peers, respectively, showing that our experiment successfully covers the vast majority of reachable peers in the Bitcoin network.

We show the cumulative distribution of the percentage of "synchronized" peers reported by two `RPC-based` monitors in Figure 3. The definition of being synchronized is borrowed from the original paper [15]; that is, a peer is said to be synchronized when it receives the latest block anytime before the next block is received by the monitors (e.g., after $\approx$ 10 minutes). Figure 3 shows that the original (i.e., inaccurate) `RPC-based` monitor reports that Bitcoin is weakly synchronized; that is, a significant portion (about 10% in the median case and 35% in the worst 10^{th} percentile case) of reachable peers are not synchronized even after about 10 minutes. In contrast, our fixed `RPC-based` monitor reports a drastically different result; that is, 95% or more Bitcoin reachable peers are almost always synchronized in less than 10 minutes. This comparison confirms that (1) the mistake of relaying block information to peers found in the `RPC-based` monitor is indeed a source of critical measurement errors and (2) the current Bitcoin is pretty well synchronized in practice!

3.3 Block Propagation Measured by Our Fixed `RPC-based` Monitor

We monitor how quickly a new block propagates through the network of reachable nodes using our fixed `RPC-based` monitor. In Figure 4, we highlight the network-wide synchronization status in the first 10 seconds since a new block is sent to our monitor and we show this for 100 consecutive blocks. First, it is evident that new block information is propagated to 90% of peers in the network in about 4 seconds on average. Second, once a new block propagates to the majority (e.g., about 90%) of reachable peers, its propagation quickly tapers off. Note that this result shows a stark difference from the same experiments made with the original `RPC-based` monitor [15], which shows that blocks take 76 seconds and 140 seconds to reach 90% of reachable peers in two examples.

3.4 Justification of Fast Block Propagation

Our measurements in this section so far strongly suggest that blocks propagate through the Bitcoin network with much faster speed than reported by the two monitors [20,15]. We now re-confirm that Bitcoin blocks indeed traverse multiple hops of nodes within a few seconds through a simple, fully-controlled experiment in a Bitcoin `regtest` network. We run 11 Bitcoin nodes in different cities around the world using Amazon EC2 `t2.large` instances. These 11 nodes are connected to each other to form a private network with a line topology of 10 hops. We note that the number of 10 hops is chosen conservatively since the network diameter of Bitcoin is unknown. We generate blocks with different sizes (i.e., 0.5 MB, 1.0 MB, 1.6 MB) at the first node and measure the elapsed time

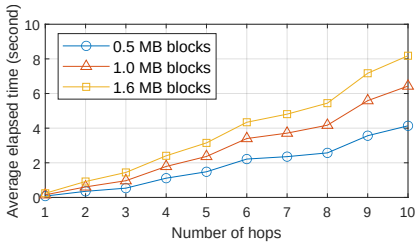


Fig. 5: Average elapsed time of multi-hop block propagation.

for the blocks to be fully received by other nodes. We repeat the same experiment 100 times and take note of the averaged propagation time. Figure 5 shows that larger blocks require more time to be propagated and all blocks need less than 10 seconds to propagate through 10 hops. We note that the application delay in Bitcoin Core nodes in the live network can be slightly higher since they would have more peers to relay blocks to (e.g., up to 125 for reachable nodes and 10 nodes for unreachable nodes). These results re-confirm the fast block propagation in Bitcoin.

3.5 Ethical considerations

Throughout our experiments, we operate a few Bitcoin nodes that differ from the default client in only some additional logging messages. Our `Bitnodes`, original, and fixed `RPC-based` monitors unavoidably occupy 1–3 out of 115 incoming slots of most reachable nodes. Hence, we run them in a very short period of time (e.g., from a few hours to one day) and minimize their disturbance to the Bitcoin network. In Section 4, we discuss the risks of allowing monitoring nodes in Bitcoin and envision a better approach for Bitcoin network monitoring with little to no ethical concerns.

4 Future Work

As we criticize the limitations and bugs found in the two monitors, we fix some of them (e.g., disabling block information forwarding) to obtain a more accurate measurement; yet, some others deserve more in-depth studies. For example, it is still unclear why `Bitnodes` frequently fails to capture the block heights of nodes with `.onion` addresses.

Another future work would be the re-evaluation of several recent Bitcoin attacks [15,17,18] that rely on the inaccurate synchronization measurements in Bitcoin. It is unclear whether the claims in these offensive security research work would still hold when Bitcoin is much better synchronized in practice.

A longer-term future work would be a clean-slate design of Bitcoin network monitors. Monitoring peer-to-peer networks has never been a designed feature of blockchain protocols and thus it always relies on running supernodes [5,14] and/or exploiting protocol side channels [6,12]. Particularly, running monitor supernodes in blockchains is a fundamentally dangerous approach because it either changes the network states (i.e., observer effect) or degrades the network performance (e.g., supernodes damage the network connectivity to some extent), creating ethical concerns. We believe that accurate yet safe network monitoring, like existing proposals for Tor performance measurements [9,11], is desired as an integrated feature of Bitcoin and other blockchains.

5 Conclusion

Network measurement is known to be tricky and error-prone when dealing with a live distributed system, comprised of heterogeneous software/hardware com-

ponents, whose states are constantly changing. This paper attempts to identify and correct some errors in recent Bitcoin network monitoring projects. Since accurate measurement of blockchain networks is evidently critical for ensuring their safety property, it is highly desirable to have more reliable and effective network monitoring primitives embedded in the blockchain protocols.

Acknowledgements This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

1. Bitcoin Lockstep Synchronous (2021), https://anonymous.4open.science/r/56e77487-0470-4e10-b634-b13e939863c0
2. bloxroute (2021), https://bloxroute.com/
3. Coin Dance: Bitcoin Nodes Summary (2021), https://coin.dance/nodes
4. Corallo, M.: BIP 152: Compact Block Relay (2016)
5. Decker, C., Wattenhofer, R.: Information propagation in the Bitcoin network. In: Proc. IEEE P2P (2013)
6. Delgado-Segura, S., Bakshi, S., Pérez-Solà, C., Litton, J., Pachulski, A., Miller, A., Bhattacharjee, B.: TxProbe: Discovering Bitcoin's Network Topology Using Orphan Transactions. In: Proc. FC (2019)
7. DSN Bitcoin Monitoring (2021), https://www.dsn.kastel.kit.edu/bitcoin/
8. FIBRE: Fast Internet Bitcoin Relay Engine (2021), http://bitcoinfibre.org/
9. Jansen, R., Johnson, A.: Safely measuring Tor. In: Proc. ACM CCS (2016)
10. Kiffer, L., Rajaraman, R., Shelat, A.: A better method to analyze blockchain consistency. In: Proc. ACM CCS (2018)
11. Mani, A., Wilson-Brown, T., Jansen, R., Johnson, A., Sherr, M.: Understanding Tor usage with privacy-preserving measurement. In: Proc. ACM IMC (2018)
12. Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N., Bhattacharjee, B.: Discovering Bitcoin's Public Topology and Influential Nodes (2015)
13. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2009)
14. Neudecker, T., Andelfinger, P., Hartenstein, H.: Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network. In: IEEE ATC (2016)
15. Saad, M., Anwar, A., Ravi, S., Mohaisen, D.: Revisiting Nakamoto Consensus in Asynchronous Networks: A Comprehensive Analysis of Bitcoin Safety and Chain Quality. In: ACM CCS (2021)
16. Saad, M., Chen, S., Mohaisen, D.: Root cause analyses for the deteriorating bitcoin network synchronization. In: Proc. IEEE ICDCS (2019)
17. Saad, M., Chen, S., Mohaisen, D.: SyncAttack: Double-spending in Bitcoin Without Mining Power. In: ACM CCS (2021)
18. Saad, M., Cook, V., Nguyen, L., Thai, M.T., Mohaisen, A.: Partitioning Attacks on Bitcoin: Colliding Space, Time, and Logic. In: Proc. IEEE ICDCS (2019)
19. Yeow, A.: Bitnodes source code (2021), https://github.com/ayeowch/bitnodes
20. Yeow, A.: Global Bitcoin nodes distribution (2021), https://bitnodes.io/