JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Chin, Z. H., Baskaran, V. M., Abaei, G., Tan, I. K. T., & Yap, T. T. V. (2023). Attestation of improved simblock node churn simulation. *Journal of Information and Communication Technology, 22*(2), 231-254. https://doi.org/10.32890/jict2023.22.2.4

Attestation of Improved SimBlock Node Churn Simulation

*[1]Zi Hau Chin, [2]Vishnu Monn Baskaran,
[3]Golnoush Abaei, [4]Ian Kim Teck Tan &
[5]Timothy Tzen Vun Yap
[1,2]School of Information Technology,
Monash University, Malaysia
[3]School of Computing Technologies,
RMIT University, Australia
[4]School of Mathematical & Computer Sciences,
Heriot-Watt University, Malaysia
[5]Faculty of Computing and Informatics,
Multimedia University, Malaysia

*[1]zi.chin@monash.edu
[2]vishnu.monn@monash.edu
[3]golnoush.abaei@rmit.edu.au
[4]i.tan@hw.ac.uk
[5]timothy@mmu.edu.my
*Corresponding author

Received: 17/7/2022 Revised: 16/1/2023 Accepted: 29/1/2023 Published: 3/4/2023

ABSTRACT

Node churn, or the constant joining and leaving of nodes in a network, can impact the performance of a blockchain network. The difficulties of performing research on the actual blockchain network, particularly

on a live decentralized global network like Bitcoin, pose challenges that good simulators can overcome. While various tools, such as NS-3 and OMNet++, are useful for simulating network behavior, SimBlock is specifically designed to simulate the complex Bitcoin blockchain network. However, the current implementation of SimBlock has limitations when replicating actual node churn activity. In this study, the SimBlock simulator was improved to simulate node churn more accurately by removing churning nodes and dropping their connections, and increasing additional instrumentation for validation. The methodology used in the study involved modeling the Bitcoin node churn behavior based on previous studies and using the enhanced SimBlock simulator to simulate node churn. Empirical studies were then conducted to determine the suitability and limitations of the node churn simulation. This study found that the improved SimBlock could produce results similar to observed indicators in a 100-node network. However, it still had limitations in replicating node churn behavior accurately. It was discovered that SimBlock limits all nodes to operate as mining nodes and that mining is simulated in a way that does not depict churn accurately at any time but only at specific intervals or under certain conditions. Despite these limitations, the study's improvements to SimBlock and the identification of its limitations can be useful for future research on node churn in blockchain networks and the development of more effective simulation tools.

Keywords: Blockchain, Compact block relay, Distributed network, Node churn, SimBlock.

INTRODUCTION

In a peer-to-peer (P2P) network, when a user launches an application, the user joins the system as a peer, contributes some resources while utilizing the resources of others, and quits the system when the user closes the program. A session is an example of a join-participate-leave cycle. The aggregate effect known as a churn is created by the independent entrance and exit of thousands or millions of peers (Stutzbach & Rejaie, 2006).

A churn, also known as node churn, in a blockchain network is almost identical to a churn in a traditional P2P network. It is the intermittent network connectivity of nodes in the blockchain network. It happens

due to the nodes joining or leaving the network independently (Hu et al., 2019). In fact, Bitcoin's whitepaper by Satoshi Nakamoto stated that Bitcoin nodes "may quit and rejoin the network at any time" (Nakamoto, 2008).

In a work published by Imtiaz et al. (2021), the authors analyzed the snapshots of the Bitcoin network from 19 May 2018 to 17 July 2018. It was discovered that 97.58 percent of 47,702 nodes were churning within the observed 60 days. Only 1,154 nodes (2.42%) were online at all times. The churn rate, or the rate at which nodes fluctuate between online and offline sessions, was then assessed. The churn rate is defined as $R = 1/T$, where T is equivalent to the frequency of completing one session cycle going from an online state to an offline state and back to the following online state. They observed the following:

1. There is a 45 percent probability that a node churns more than once daily (i.e., $R > 1$).
2. The likelihood that a node would churn at least nine times is 10 percent (i.e., $R \geq 9$).
3. The daily churn rate is 4.16 on average (i.e., $\hat{R} = 4.16$).

The 97.58 percent churning of nodes was also concurred and reflected as 97 percent in a revised version of the blockchain network simulator, SimBlock (Aoki et al., 2019) released in the middle of 2020. However, the churn simulation in SimBlock does not simulate the node leaving the network, ensuring that the inactive node no longer participates in validating transactions or receiving new blocks.

SimBlock – Blockchain Network Simulator

SimBlock is an event-driven blockchain network simulator. The aim is to ease the difficulty of investigating the behavior of blockchain networks, as blockchain networks generally have a large number of nodes geographically distributed over a vast region (Faria & Correia, 2019). In addition, the inherent components of the blockchain networks, such as the Proof-of-Work (PoW) consensus mechanism and the peer-to-peer networking communication, add further complexities to the cost or viability of studying the blockchain networks (Fattahi et al., 2020). SimBlock allows for the behavior of nodes to be modified, thus enabling the investigation of their influence on blockchain networks with the primary focus on simulating the Bitcoin network, where all

participating nodes are responsible for generating communication messages as well as the mining events (Hanggoro & Sari, 2021).

In mid-2020, SimBlock implemented the compact block relay (CBR) mechanism, also referred to as Bitcoin Improvement Proposal 152 (BIP-152), which was an improvement in block propagation across the P2P Bitcoin network (Corallo, 2016). In the Bitcoin P2P protocol, a transaction is propagated twice. First, when the transaction has been submitted and once when the block containing the transaction is successfully mined. It was deemed as not bandwidth efficient for block propagation. CBR seeks to reduce the bandwidth usage of a node on the P2P network by eliminating the redundancy of sending a transaction twice. This process is feasible due to the mempool of nodes, which generally hold identical information (i.e., transactions) (Zhou et al., 2020). Therefore, instead of sending complete transactions during block propagation, which is redundant, a compact block contains a shortened transaction identifier (TXID) and some prefilled transactions that the sender assumes are unknown to the receiver. The receiver can then reconstruct the block with the obtained information and its mempool without needing the known transactions to be propagated twice (Nagayama et al., 2020). The node churn feature is also included with this implementation in the simulator. The simulation definition of churn in SimBlock is described in Algorithm 1, where the probability for the CBR node failing is considered.

Algorithm 1: Node Churn Definition in SimBlock

```
if new block == compact block then
 if node == churn node then
  CBR failure rate = 0.27
 else
  CBR failure rate = 0.13
 endif
 Success = random() > CBR failure rate
 if Success == true then
  Validate the block
 else
  Request missing transactions
 endif
endif
```

In SimBlock, the CBR failure rate is the probability of a node failing to reconstruct a block using a compact block. Therefore, the difference

between a non-churning node and a churning node is dictated by the CBR failure rate, where a churning node has a higher probability of failing to reconstruct a block and higher bandwidth usage due to failed reconstruction. Imtiaz et al. (2021) stated that node churn affects the performance of CBR, hence the depicted higher CBR failure rate of churning node in SimBlock. However, it is believed that the actual churn activity is not reproduced in SimBlock. The current study improved the simulation of churn in SimBlock to include removing the churning nodes from the network and dropping ongoing inbound and outbound network connections. This research then proceeded to conduct empirical studies to attest to the simulator:

- Validating the simulation vis-à-vis reported actual observations of block propagation to (50%) and (90%) of nodes and the average block time achieved.
- The enhancement of SimBlock is also bound to increase simulation complexity. The present authors ran various simulated network sizes in terms of nodes to determine the upper limits on a standard desktop machine for simulation.

The following "Related Works" section will discuss other studies that have been performed on blockchain node churn or the process of nodes repeatedly joining and leaving the network. The "Enhanced SimBlock Node Churn" section will then describe the methods used to improve the simulation of node churn in SimBlock, a tool used to simulate the behavior of a blockchain. The "Experiment Setup" section will outline the specific details of the experiment, including the parameters and variables being tested. The "Analysis and Results" section will present the findings of the experiment and any attestation work done to validate the improvement of the SimBlock node churn simulation. Finally, the "Conclusion" section will summarize the paper's contributions and provide any final thoughts or recommendations for future research in this area.

RELATED WORKS

This work has presented the augmentation of a proposed built simulator for a blockchain network to include node churns. This section will examine some other enhancements proposed for SimBlock. The studies on node churns will then be examined since they are essential,

especially in a decentralized distributed network as it affects the network communication, which has to be self-rectifying without centralized supervision. One such distributed network is a blockchain network. In a quickly expanding ad hoc distributed network, the requirement for a scalable way to disseminate transactions across the network is crucial for transaction integrity.

SimBlock could not faithfully imitate the process of computing hashes according to Mardiansyah et al. (2022). As a result, the authors added difficulty levels to SimBlock's hashing algorithm. The difficulty level measures the difficulty of locating the necessary hash target, a number that a miner must attempt to obtain to generate a new block. The authors wanted their simulation to mimic the mining process more closely and the time it took to create a new block on an existing blockchain network utilizing the PoW consensus by introducing difficulty levels. With the original SimBlock, the simulation begins by generating a certain number of regions specified by the user. Thereafter, nodes are created inside these regions and connections are made between them. The simulation continues by creating blocks, distributing them to all the nodes and calculating the propagation time for each block. The simulation ends when the specified height of the last block is achieved. Nevertheless, the authors proposed a modified SimBlock simulator, which used the PoW consensus algorithm to create blocks in the simulation. In the modified SimBlock simulator, reading the difficulty value, assigning a random nonce value, aggregating all pertinent data, and utilizing SHA-256 hashing are all steps in the block generation process. Block generation is successful, and the block is propagated out if the final hash value meets the specified target hash value. The operation is continued until the desired hash value is obtained, at which point the nonce value is reset to a random number.

Basile et al. (2021) studied and concluded that because SimBlock was using a parametrization that resembled the state of the Bitcoin blockchain in 2021, it was unsuitable for mining blocks due to its low hash rate. Additionally, SimBlock does not currently simulate the incentive mechanism, which limits its effectiveness in evaluating Bitcoin-based services. In addressing these limitations, the authors proposed a new implementation of SimBlock with ad hoc improvements. The authors made several modifications to the original SimBlock implementation to improve its performance. One of these enhancements involved switching from the primitive Java double data

type to the arbitrary-precision signed decimal data type, BigDecimal, to support more precise arithmetic and comparison operations. Using Newton's approach, the authors also provided functions to calculate the logarithm of BigDecimal numbers. The second modification was revising the threshold amount to account for the current hash rate of the Bitcoin network and allowing room for an increase in the overall number of miners and advancements in mining technology to let SimBlock carry out accurate block mining. It was necessary to adjust the threshold value to consider a hypothetical three-order increase in the network hash rate.

Motlagh et al. (2020) investigated the queuing performance of blocks and transaction traffic due to node churning that was conducted before the CBR implementation. The number of blocks that arrive at each active node (block arrival rate) in the network grows as the number of active nodes decreases. As the block arrival rate rises, the block reaction time also increases. As a result of the almost constant block arrival rate described above, the number of blocks waiting at each node grows. With more blocks in the process queue, there will be a longer waiting time, resulting in a longer block response time. Node churn affects the number of nodes propagating new blocks and consequently blocks arrival rate and response time. It also affects the time it takes to propagate a block. Due to fewer active nodes and the minor drop in the mean number of hops, block propagation time reduces as the churn duration grows for smaller networks. Block propagation time becomes less reliant on the churning duration for larger networks (node $\geq$ 4,000) since the increase in block arrival rate and block response time tends to balance each other out. On the other hand, transaction arrival rates per node are greater in networks with fewer nodes and a longer churning duration. Furthermore, transactions are given lesser priority than blocks even though transactions are processed faster. While transaction response time is nearly independent of the churning duration in large networks, it is more sensitive in smaller networks. A decrease in the number of nodes available for processing will substantially increase response time. This impact causes transaction propagation time to reduce as the network grows while increasing as the churning duration expands. More extended churning periods result in more transactions arriving at active nodes. Consequently, transactions tend to collect while waiting for blocks to be processed over more extended churning periods, resulting in a significant increase in transaction delivery time in smaller networks.

Imtiaz et al. (2021) stated that blocks received by churning nodes have a propagation delay that is more than twice as long as any block received by non-churning nodes (i.e., 105.54 seconds vs. 46.14 seconds). It is important to remember that the block propagation delay they referred to is a single-hop block propagation delay (i.e., the time it takes for a node to fully recover and rebuild a block after receiving a block announcement from a peer). If a receiver's mempool has all the transactions whose hashes are present in the received compact block, then the original block can be correctly reconstructed. However, it will fail to reconstruct the block if not all transactions are present in the node's mempool (Imtiaz et al., 2022). When the compact block protocol fails, the additional round trips will delay block propagation and raise the danger of a blockchain fork. They also observed that a churning node misses 78.08 transactions per block with a standard deviation of 288.04. In contrast, a non-churning node misses 0.87 transactions per block on average with a standard deviation of 10.78 transactions. Approximately 11 percent of blocks received by churning nodes are missing more than 100 transactions, with some blocks having as many as 2,722 missing transactions. Only around 0.3 percent of blocks received by non-churning nodes lack more than 100 transactions, with a maximum of 307 missing transactions per block.

Paulaviius et al. (2021) presented a comprehensive and contemporary evaluation of 27 blockchain simulators. SimBlock was also included in an in-depth study and feature comparison with four other simulators. The criteria for picking blockchain simulators for further investigation were source code availability, capacity to explore a wide variety of properties of a simulated blockchain network, and flexibility to change input data and settings to represent current blockchain networks. SimBlock has many capabilities, including mimicking the network layer while considering the node's latency distribution, geographical distribution, and bandwidth. However, it has a significant flaw because it does not distinguish between full nodes and miners, meaning every node acts as a miner in SimBlock. Users may consider trust in a P2P network like the Bitcoin network when deciding which peers to communicate with (Firdhous et al., 2014). Users may be more inclined to engage and trust peers who have a reputation for being trustworthy than to trust peers who have a reputation for participating in dishonest or harmful behavior. Since SimBlock assumes that all nodes are trustworthy, it cannot be used to investigate malicious nodes. Another fundamental drawback is that

SimBlock merely simulates a blockchain network at the block level with no consideration for transaction propagation.

In summary, no global simulator is suitable to be used in various situations. Existing simulators have limited capabilities since they are designed to emulate only a few key characteristics of a real blockchain while simplifying the rest. As a result, Paulaviius et al. (2021) chose three of the most promising PoW blockchain simulations: SimBlock, Bitcoin-Simulator (Gervais et al., 2016), and BlockSim: Alharby (Alharby & van Moorsel, 2019) for further testing. They found that all three simulators could correctly reproduce the Bitcoin network in 2016. Regrettably, the Bitcoin-Simulator, one of the most promising and practical simulators, is now obsolete and unable to correctly represent the current Bitcoin network as it fails to accurately simulate the Bitcoin network in 2020. The stale block rate and block propagation delay have been drastically reduced due to recent improvements to the Bitcoin protocol, such as the compact block relay. SimBlock, which supports the compact block relay, produces simulation results that indicate improvement. It successfully reproduced the Bitcoin network in 2019 and 2020 to an acceptable degree. Nevertheless, it is unclear how precise and realistic the modeling method is without simulating transaction propagation.

ENHANCED SIMBLOCK NODE CHURN

For a node to know if it is still connected to a peer, it must periodically send a "ping" message to that peer. The "ping" message determines whether a peer network address is reachable and responsive. Meanwhile, a "pong" message is a reply to the "ping" message to prove that a node is alive (Mastan & Paul, 2018). If a response is not received within a specified period, ping timeout occurs (timeout interval), representing a disconnected peer. In the Bitcoin Core implementation, this timeout interval is stored as the variable *TIMEOUT_INTERVAL* in the file "net.h". It is set to 1,200 seconds (20 minutes) by default. Historically, it was set to 5,400 seconds (90 minutes) (Hellani et al., 2019), but it was later changed to 1,200 seconds (20 minutes) during the release of Bitcoin Core v0.10 in February 2015 (Walck et al., 2019) and has been the same ever since. If a peer uses an older version of Bitcoin Core that does not support the "pong" message, the timeout interval will be 90 minutes.

Nevertheless, the checking for timeout interval of 90 minutes has been removed in Bitcoin Core v0.22 released in October 2021. This removal is because the number of nodes running the Bitcoin Core version earlier than v0.10 should be close to none. Additionally, a node running Bitcoin Core version earlier than v0.10 that has not sent any messages in 90 minutes is most likely not beneficial. These nodes will be removed due to their slow ping time and will not provide any blocks or transactions. In summary, an active node will disconnect the connection of an inactive node if (Gervais et al., 2015):

1. A newly established connection has been active for 60 seconds and has not transmitted or received any data.
2. An already connected peer did not transmit or receive data within the timeout interval of 1,200 seconds (20 minutes) since the last successful communication.

Due to the nature of SimBlock where every node acts as a miner and all nodes are connected the moment the simulation starts, it is safe to assume that the aforementioned first rule of an inactive node will never occur. Furthermore, as churning nodes in SimBlock do not disconnect from the network, they will participate in block validation and propagation without exception. Therefore, this study will only implement the second rule to determine inactive nodes.

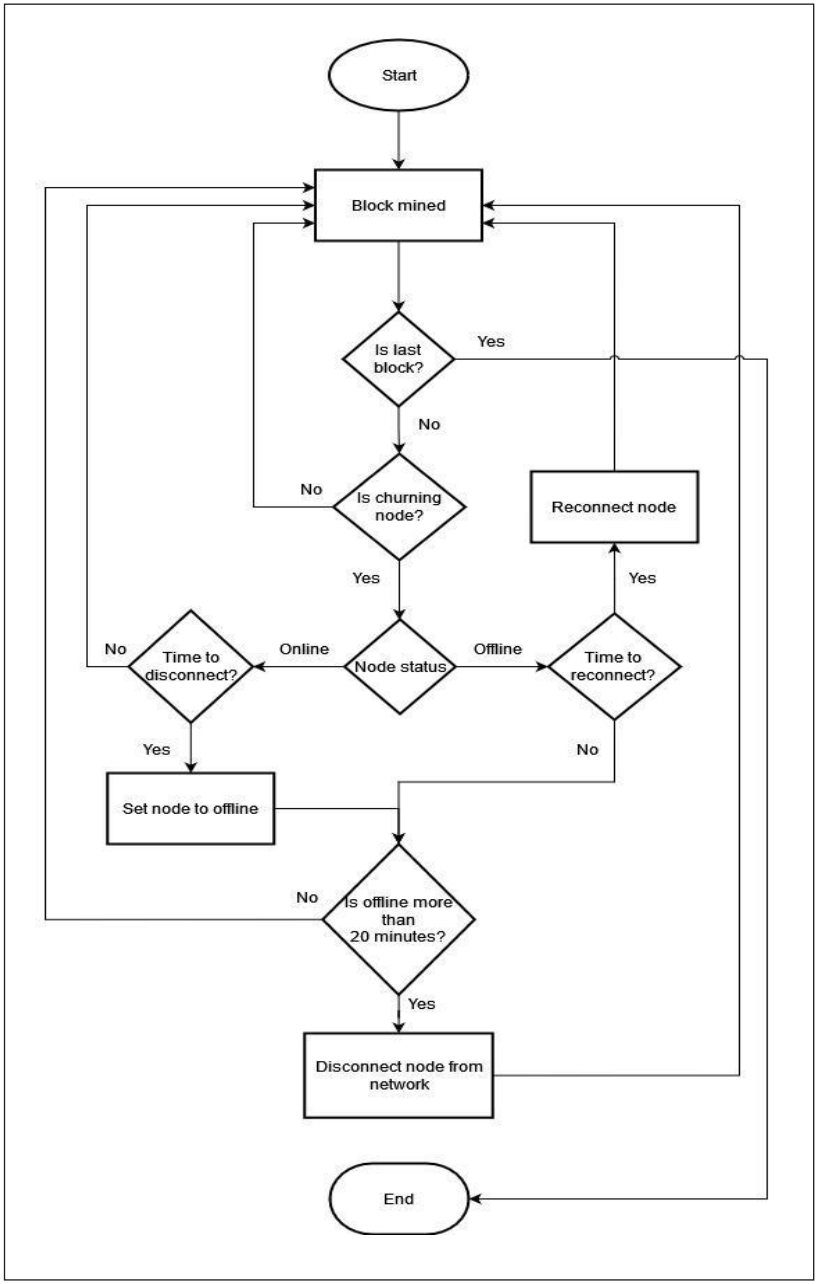
In ensuring that churning nodes are inactive (i.e., not participating in validating, propagating, and mining of blocks) in SimBlock, this study will remove churning nodes from the network. In SimBlock, a list records all the generated simulated nodes for this run. Churning nodes will be temporarily removed from the list of simulated nodes when they churn to simulate their inactivity. Therefore, they are not participating in the network. However, nodes previously connected to the churning nodes will not immediately sever the connection according to the second rule. They will only be disconnected if they are inactive for at least 20 minutes. The term "neighbor" in SimBlock indicates connected nodes and each node has its list of neighbors. A churning node is removed immediately from the simulated node list but will only be removed from a node's list of neighbors after 20 minutes. Therefore, if a churning node reconnects to the network within 20 minutes, it will reconnect to the previous nodes. Otherwise, it will connect to new nodes.

In SimBlock, an existing but unused function is known as `removeNode` (). This function removes nodes from the list of

simulated nodes but does not remove them from the network. The churning node's inbound and outbound connections need to be severed to remove a node from the network. This study will use the functions `removeNeighbor` () and `addNeighbor` (), which are existing functions in SimBlock but are unutilized. This is how the enhanced SimBlock node churn works. When a SimBlock simulation begins, N nodes are generated. and X nodes are randomly chosen depending on the churn node rate to act as churning nodes. Thereafter, the churning nodes will be subjected to a series of timings to ascertain the pattern of node churn and how long they should be online and offline. The pattern of node churn will be thoroughly explained in the next section. Instead of adopting Algorithm 1 to simulate node churn, the current study developed and put into practice the following solution as shown in Figure 1. When a block is mined and not at the end block height, it will determine the churning nodes' current status (online or offline). If a node is currently online but should be offline (based on the churning pattern), the node's status is set as offline. An offline node will not be able to contribute or participate in the network. This study removed the node from the network to prevent peers from trying to connect with a churning node that has been offline for longer than 20 minutes.

Figure 1

Flow Chart of the Enhanced Simulation of Node Churn for SimBlock



EXPERIMENT SETUP

Instrumentation

In addition to the implementation of removing a churning node from the network and dropping the connections of a churning node if it has been inactive for at least 20 minutes, the authors added more measurements to the simulator. The modifications for the instrumentation were made on multiple files, namely "Main.java" and "Simulator.java", which implemented the following measurements:

1. Average block time (actual time taken to mine a block).
2. Average block propagation delay to 50 percent of the nodes (average time taken for a block to reach 50 % of the nodes).
3. Average block propagation delay to 90 percent of the nodes (average time taken for a block to reach 90 % of the nodes).

Churning Node Behavior

Although Imtiaz et al. (2021) investigated and defined the churn rate *R*, the online and offline session durations for each cycle of the nodes were not constant and needed to be modeled. The authors used a snapshot of the Bitcoin network from 19 May 2018 to 17 July 2018 to perform an online and offline duration distribution fitting for the churning nodes. They did this with 25,000 minutes from the obtained snapshot and determined the goodness-of-fit of the different distributions to the data by utilizing the R-squared (R^2) and Root Mean Square Error (RMSE) criteria. They used MATLAB's *fitdist()* function to conduct the distribution fitting based on the maximum likelihood estimation. An exhaustive search within 10 percent of various models was conducted, where the highest R^2 and lowest RMSE in that range were chosen as the final findings of each distribution. Based on R^2 and RMSE, log-normal distribution had the best result.

1) Distribution of Node Online Session

When performing distribution fitting on the online sessions independently, which had a heavy tail distribution, the cumulative distribution function (CDF) of empirical data fit well with the log-logistic distribution. Based on the R^2 and RMSE scores and distribution fitting, they concluded that the log-logistic distribution was the best fit for the online sessions.

2) Distribution of Node Offline Session

The distribution fitting for the offline session was performed on offline sessions for up to one day (93% of the cases). A node's mempool (storage for unconfirmed transactions) would be out of synchronization with the rest of the network if it were disconnected for more than one day. For the offline session, it was concluded that the Weibull distribution model fit the data best.

3) Distribution Parameters Selection

Table 1 shows parameters obtained from the distribution fitting of online and offline sessions of the nodes independently. Using the parameters as shown in Table 1, Imtiaz et al. (2021) generated the online and offline sessions for four churning nodes with the following requirements:

1. The aggregate sum of the online and offline sessions was at least two weeks for each node.
2. A minimum of one second and up to a maximum of one day for both online and offline sessions.

Table 1

Parameters for the Best Distribution Fitting (Imtiaz et al., 2021)

Distribution	Session	Scale Parameter	Shape Parameter
Log-logistic	Online	11.00	0.771
Weibull	Offline	0.64	0.183

The present researchers used the parameters in Table 1 and adhered to the requirements stated above to generate the online and offline sessions for 100 nodes. By comparing the CDF of the generated sessions with Imtiaz et al.'s (2021) generated sessions, it was observed that the generated sessions differed from the authors' generated sessions. Nevertheless, the network snapshot (May to July 2018) that the authors used could not be obtained. This observation was due to Bitnodes only providing the latest 60 days network snapshot. Instead of fitting the same dataset as Imtiaz et al. (2021), the present researchers performed distribution fitting on Imtiaz et al.'s (2021) generated online and offline sessions that are available for download (Imtiaz, 2021). By distribution fitting the authors' generated set of online and offline sessions, the current study obtained the following parameters as recorded in Table 2. The visuals of the distribution fittings for online and offline sessions are shown in Figures 2 and 3.

Table 2

The Obtained Parameters for the Best Distribution Fitting

Distribution	Session	Scale Parameter	Shape Parameter
Log-logistic	Online	7.97945	1.07028
Weibull	Offline	1523.49	0.420073

Figure 2

Distribution Fitting for Imtiaz et al.'s Online Session (Imtiaz et al., 2021)

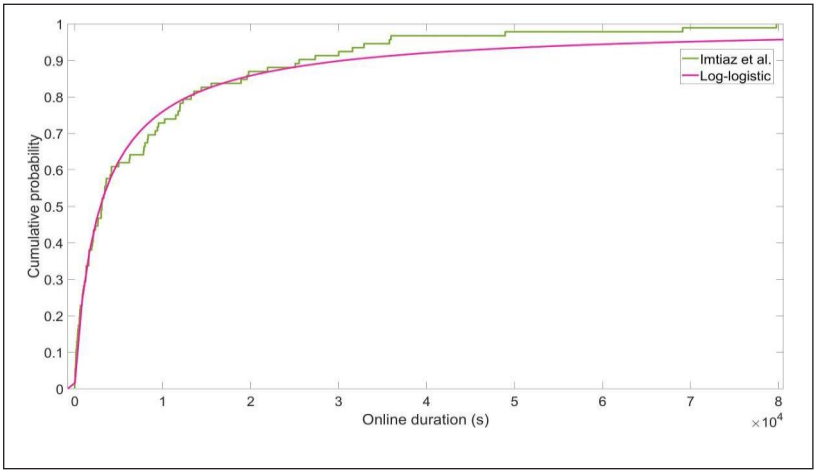
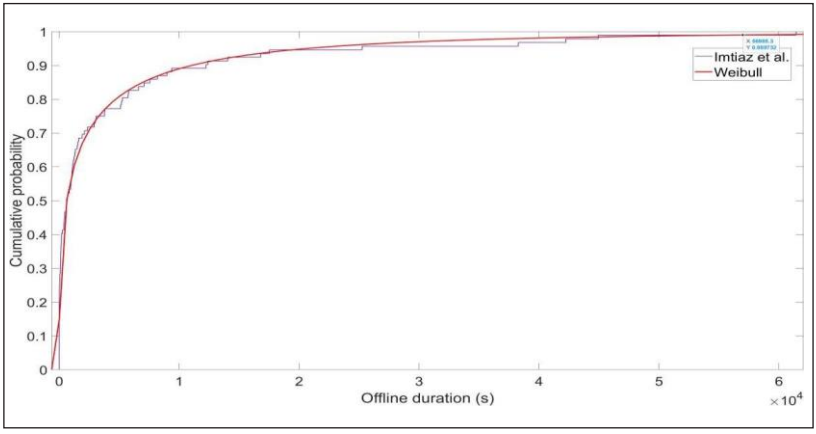


Figure 3

Distribution Fitting for Imtiaz et al.'s Offline Session (Imtiaz et al., 2021)



Based on the newly obtained parameters shown in Table 2, this study generated the online and offline sessions for 100 nodes. Figures 4 and 5 show the comparisons of generated online and offline sessions, respectively. The results were based on Imtiaz et al.'s (2021) parameters, this study's proposed parameters, and the originally generated sessions that were downloaded. The CDF of the generated sessions was then compared to Imtiaz et al.'s (2021), and a better fit overall was observed. With the improved SimBlock node churning code, the additional instrumentations, and the modeling of the node churn pattern completed, this study performed the simulations to validate the work.

Figure 4

Comparison of Generated Online Sessions

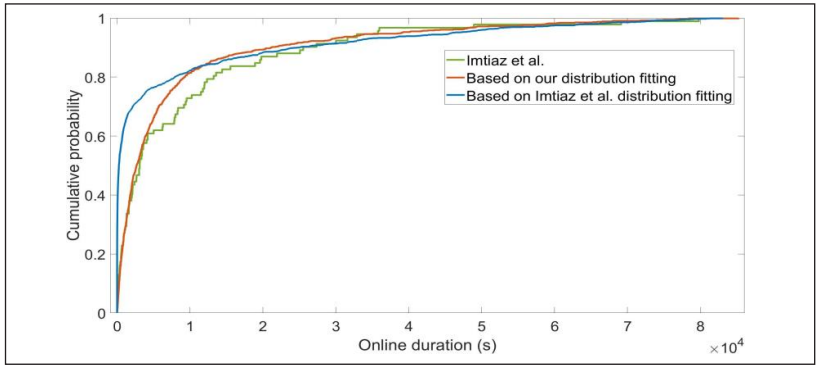
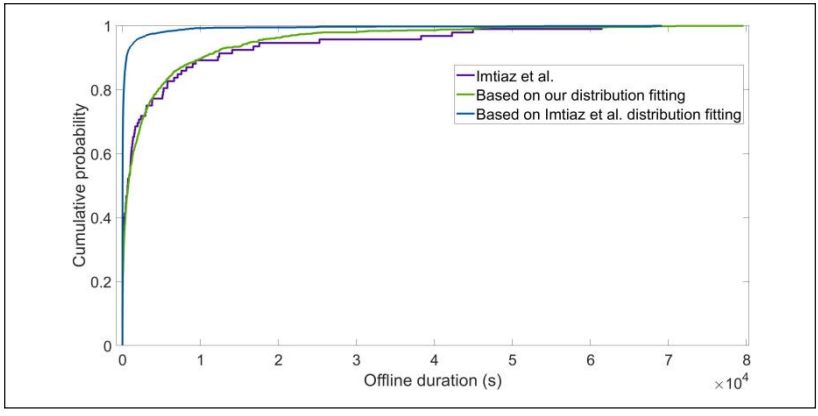


Figure 5

Comparison of Generated Offline Sessions



ANALYSIS AND RESULTS

This research used the settings as shown in Table 3 for the SimBlock simulator and ran three simulation types:

1. Simulation 1: base simulator.
2. Simulation 2: immediate churning node removal.
3. Simulation 3: churning node removal after being inactive for 20 minutes.

For each simulation type, the researchers simulated 30 iterations and the average obtained results are recorded in Table 4. The historical actual observed value of the Bitcoin network in 2019 was obtained (Yeow, n.d.). The reason was the network parameters, such as upload bandwidth, download bandwidth, and latency, which SimBlock provided were based on 2015 and 2019.

As shown in Table 4, results from Simulation 2 showed an increase in every measurement compared to Simulation 1. By comparison, the block propagation delay to reach 50 percent nodes increased by 424.43 percent, the block propagation delay to reach 90 percent nodes increased by 397.78 percent, and the average block time increased by 59.07 percent. Simulation 3 also experienced an increase in every measurement as compared to Simulation 1 but not as significant as Simulation 2. The block propagation delay to reach 50 percent nodes increased by 270.72 percent, the block propagation delay to reach 90 percent nodes increased by 317.72 percent, and the average block time increased by 5.25 percent.

Based on the observation, Simulation 3, where the churning node's inbound and outbound connections would only be dropped if the churning node had been inactive for at least 20 minutes, was more accurate than Simulation 2. This observation was especially evident when the average block time of Simulation 2 was 955.55 seconds when the churn would supposedly not affect the average block time. Nevertheless, the average block time for Simulations 2 and 3 deviated from the expected 600 seconds. This outcome could be due to the node acting as a miner (transaction validator) in SimBlock. Thus, when the churning node's inbound and outbound had been dropped, the available miner decreased and the average block time increased.

Table 3

SimBlock's Settings

Parameters	Description	Value
Network/ Consensus type	The consensus algorithm of the network	Bitcoin/Proof-of-Work
Number of nodes	The number of nodes participating in the blockchain network	100
Block generation interval	Expected time taken to mine a block	600 seconds
End block height	The block height when a simulation ends	10,000
Block size	The average block size of a block	18KB
CBR usage rate	Usage rate of compact block relay	96.4%
Churn node rate	Rate of node that churns	97.6%
CBR failure rate for control node	Chance of failing to reconstruct a block with CBR of non-churning node	13%
CBR failure rate for churning node	Chance of failing to reconstruct a block with CBR of churning node	27%

Table 4

Simulation Results

	Block propagation delay to reach 50% nodes (milliseconds)	Block propagation delay to reach 90% nodes (milliseconds)	Average block time (seconds)
Simulation 1	986.45	2611.9	600.68
Simulation 2	5173.31	13001.60	955.55
Simulation 3	3656.97	10910.46	632.26
Actual Observed (2019)	526.58	3305.40	602.91

Table 5

Simulation 3 with Varying Numbers of Network Nodes

Number of nodes	Block propagation delay to reach 50% nodes (milliseconds)	Block propagation delay to reach 90% nodes (milliseconds)	Average block time (seconds)
50	1071.41	3159.70	621.57
100	351.44	10600.78	632.89
150	2411.44	9236.68	638.73
200	2227.78	10724.37	648.71
250	5025.49	19364.71	681.60
300	6685.69	23727.15	704.62
Actual Observed (2019)	526.58	3305.40	602.91

This study further performed simulations using Simulation 3 with varying numbers of network nodes from 50 to 300. Table 5 shows that the average block time increased as the researchers increased the total number of simulated nodes. The results depicted in Table 5 were averaged from 30 simulation iterations. As observed, the average block time seemed to be affected by the node churn. As the number of nodes increased, the average block time grew. This observation was due to the limitation of SimBlock, where each node simulated was also a mining node, thus affecting the mining performance when it was churned. Furthermore, the increased complexity of the simulation affected the simulator. The actions of disconnecting churning nodes from the network and re-adding them back had taken its toll on this simulator.

It can be concluded that the node churn had been implemented correctly in SimBlock as the simulation could achieve similar ballpark figures as the approximation from the actual reported observations. This deduction considered that in the actual live Bitcoin network, not all the nodes were mining nodes, whereas in SimBlock, all the simulated nodes acted as mining nodes. Therefore, there were some penalties for achieving the average block time.

Overall, the current SimBlock implementation was unsuitable for using a realistic simulation of node churning due to two main limitations. The first restriction was that the nodes, as depicted in SimBlock, were limited to miners only, significantly impacting the average block time as seen in Table 5. Although it was theoretically feasible for miners to churn, this was less likely given their incentives. A miner's chances of solving the block's cryptographic challenge and receiving the reward would be significantly reduced if they churned (left the network) in the middle of mining it. The negative impacts were further amplified by the fewer nodes participating in the network because this study was only able to deploy 300 nodes when using the enhanced node churn simulation. Motlagh et al. (2020) stated that the network's performance was significantly impacted by churn, but a Bitcoin network with numerous nodes is impervious to churn.

The second restriction was due to how mining was depicted in SimBlock. This study could only churn a node every time after a block had been mined. The term "churn" describes the rate of a node repeatedly joining and leaving the network as desired (Rodrigues et al., 2022). A node can churn at any time and is not supposedly restricted from doing so only when a block is mined. Nevertheless, mining is not simulated in the traditional sense as the time it takes for new blocks to be added to the simulated blockchain is calculated based on specific parameters. As a result, it is impossible to churn a node at any time but only at specific intervals or under certain conditions. Therefore, this is an inaccurate depiction of how mining and node churning work in a blockchain system.

CONCLUSION

The importance of simulating node churns in a blockchain network can be summarized by the studies of Motlagh et al. (2020) and Imtiaz et al. (2021). The current work enabled this based on the SimBlock simulator by Aoki et al. (2021). In summary, this study enhanced the Simblock simulator for node churns and included instrumentation for validation metrics. Furthermore, the paper successfully modeled the Bitcoin node churn behavior based on the studies by Imtiaz et al. (2021) with the provided parameters. The parameters were used for simulating node churns in the enhanced simulator. Lastly, the study conducted validation of the node churn by comparing it vis-à-vis actual observed Bitcoin network metrics.

In conclusion, no universal simulator can be utilized in many scenarios. Existing simulators can only simulate a few essential features of an actual blockchain while the rest are simplified. Recent enhancements to the Bitcoin protocol, including the compact block relay, have significantly decreased the stale block rate and block propagation delay over time. Simulation results from SimBlock, which supports the compact block relay, have shown improvement. SimBlock has reasonably replicated the Bitcoin network in 2019 and 2020.

Furthermore, the simulation of node churn by SimBlock has been implemented correctly based on the experiments, as they yielded results that were roughly comparable to those obtained from reported observations. Nonetheless, due to the characteristic of SimBlock, where all simulated nodes were functioning as mining nodes, the performance dropped significantly with the implementation due to the increased simulation complexity. This result could be seen from the deviation of block propagation and average block time from the actual reported value. As a consequence of this limitation, this paper could not simulate more than 300 nodes during the experiment of Simulation 3 with varying numbers of network nodes as it significantly increased the time taken to run an iteration.

In improving and expanding upon the findings reported here, future studies should focus on several areas. The following are some potential directions for further research:

1. The simulator should be enhanced by adding capabilities (e.g., simulating miner, full node, and lightweight node) that let users replicate more extensive and complicated networks because it is intended to examine the scalability of various blockchain technologies. SimBlock would become a more valuable tool for academics and developers working on blockchain technology if its scalability could be improved. This would enable users to assess the performance of various blockchain protocols and applications precisely under various scenarios.
2. There is a need to enhance SimBlock's simulation and lessen its influence on the outcomes to gain more precise and reliable results. It was discovered throughout this study that the results were impacted by the increased computation brought about by using an accurate node churn simulation.

In addressing the issue of the simulator's performance affecting the simulation results, one option would be to use more efficient algorithms or optimize the code to reduce the impact of calculation workload on the simulation and produce more accurate results.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education, Malaysia, under the Fundamental Research Grant Scheme with grant number FRGS/1/2020/ICT02/MUSM/02/2.

REFERENCES

Alharby, M., & van Moorsel, A. (2019). BlockSim: A simulation framework for blockchain systems. *ACM Sigmetrics Performance Evaluation Review*, *46*(3), 135–138. https://doi.org/10.1145/3308897.3308956

Aoki, Y., Otsuki, K., Kaneko, T., Banno, R., & Shudo, K. (2019, April). SimBlock: A blockchain network simulator. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2019)* (pp. 325–329). IEEE. https://doi.org/10.1109/INFCOMW.2019.8845253

Basile, M., Nardini, G., Perazzo, P., & Dini, G. (2021, September). On improving SimBlock blockchain simulator. In *Proceedings – 2021 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1–6). https://doi.org/10.1109/ISCC53001.2021.9631470

Corallo, M. (2016, May). *BIP 152*. GitHub. https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki

Faria, C., & Correia, M. (2019, July). BlockSim: Blockchain simulator. In *Proceedings - 2019 2nd IEEE International Conference on Blockchain (Blockchain 2019)* (pp. 439–444). https://doi.org/10.1109/Blockchain.2019.00067

Fattahi, S. M., Makanju, A., & Milani Fard, A. (2020, June). SIMBA: An efficient simulator for blockchain applications. In *Proceedings – 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks: Supplemental Volume (DSN-S 2020)* (pp. 51–52). IEEE. https://doi.org/10.1109/DSN-S50200.2020.00028

Firdhous, M., Ghazali, O., & Hassan, S. (2014). Statistically controlled robust trust computing mechanism for cloud computing. *Journal of Information and Communication Technology*, *13*, 21–36. Retrieved from https://e-journal.uum.edu.my/index.php/jict/article/view/8146

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016, October). On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 3–16). https://doi.org/10.1145/2976749.2978341

Gervais, A., Ritzdorf, H., Karame, G. O., & Čapkun, S. (2015, October). Tampering with the delivery of blocks and transactions in bitcoin. In *Proceedings of the ACM Conference on Computer and Communications Security 2015* (pp. 692–705). https://doi.org/10.1145/2810103.2813655

Hanggoro, D., & Sari, R. F. (2021, May). Performance comparison of SimBlock to NS-3 blockchain simulators. In *2021 4th International Conference on Circuits, Systems and Simulation (ICCSS)*, (pp. 45–50). IEEE. https://doi.org/10.1109/ICCSS51193.2021.9464212

Hellani, H., Samhat, A. E., Chamoun, M., Ghor, H. el, & Serhrouchni, A. (2019). On BlockChain technology: Overview of bitcoin and future insights. In *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET 2018)* (pp. 1–8). https://doi.org/10.1109/IMCET.2018.8603029

Hu, Y., Manzoor, A., Ekparinya, P., Liyanage, M., Thilakarathna, K., Jourjon, G., & Seneviratne, A. (2019). A delay-tolerant payment scheme based on the ethereum blockchain. *IEEE Access*, *7*, 33159–33172. https://doi.org/10.1109/ACCESS.2019.2903271

Imtiaz, M. A., Starobinski, D., Trachtenberg, A., & Younis, N. (2021). Churn in the bitcoin network. *IEEE Transactions on Network and Service Management*, *18*(2), 1598–1615. https://doi.org/10.1109/TNSM.2021.3050428

Imtiaz, M. A., Starobinski, D., & Trachtenberg, A. (2022). Empirical comparison of block relay protocols. *IEEE Transactions on Network and Service Management*. https://doi.org/10.1109/TNSM.2022.3195976

Imtiaz, M. A. (2021). *tnsm-churn*. GitHub. https://github.com/nislab/bitcoin-logs/tree/tnsm-churn

Mardiansyah, V., & Sari, R. F. (2022). SimBlock simulator enhancement with difficulty level algorithm based on proof-of-work consensus for lightweight blockchain. *Sensors 2022*, *22*(23), 9057. https://doi.org/10.3390/S22239057

Mastan, I. D., & Paul, S. (2018). A new approach to deanonymization of unreachable bitcoin nodes. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11261 (pp. 277–298). https://doi.org/10.1007/978-3-030-02641-7_13

Motlagh, S. G., Mišić, J., & Mišić, V. B. (2020). Impact of node churn in the bitcoin network. *IEEE Transactions on Network Science and Engineering*, *7*(3), 2104–2113. https://doi.org/10.1109/TNSE.2020.2974739

Nagayama, R., Banno, R., & Shudo, K. (2020, July). Identifying impacts of protocol and internet development on the bitcoin network. In *2020 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1–6). IEEE. https://doi.org/10.1109/ISCC50000.2020.9219639

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

Paulaviius, R., Grigaitis, S., & Filatovas, E. (2021). A systematic review and empirical analysis of blockchain simulators. *IEEE Access*, *9*, 38010–38028. https://doi.org/10.1109/ACCESS.2021.3063324

Rodrigues, B., Franco, M., Killer, C., Scheid, E. J., & Stiller, B. (2022). On trust, blockchain, and reputation systems. In *Handbook on blockchain* (pp. 299–337). Cham: Springer International Publishing.https://doi.org/10.1007/978-3-031-07535-3_9/TABLES/5

Stutzbach, D., & Rejaie, R. (2006, October). Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement* (pp. 189–202). https://doi.org/10.1145/1177080.1177105

Walck, M., Wang, K., & Kim, H. S. (2019, July). TendrilStaller: Block delay attack in bitcoin. In *Proceedings - 2019 2nd IEEE International Conference on Blockchain (Blockchain 2019)* (pp. 1–9). IEEE. https://doi.org/10.1109/Blockchain.2019.00010

Yeow, A. (n.d.). *Bitcoin Network 5 Years Charts - Bitnodes*. Bitnodes.io. Retrieved 15 June 2022, from https://bitnodes.io/dashboard/?days=1825

Zhou, Q., Huang, H., Zheng, Z., & Bian, J. (2020). Solutions to scalability of blockchain: A survey. *IEEE Access*, *8*, 16440–16455. https://doi.org/10.1109/ACCESS.2020.2967218