On Improving SimBlock Blockchain Simulator

Mariano Basile
Dept. of Ingegneria dell'Informazione
University of Pisa
Largo Lucio Lazzarino 1, Pisa, Italy
mariano.basile@ing.unipi.it

Giovanni Nardini
Dept. of Ingegneria dell'Informazione
University of Pisa
Largo Lucio Lazzarino 1, Pisa, Italy
giovanni.nardini@unipi.it

Pericle Perazzo
Dept. of Ingegneria dell'Informazione
University of Pisa
Largo Lucio Lazzarino 1, Pisa, Italy
pericle.perazzo@unipi.it

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Largo Lucio Lazzarino 1, Pisa, Italy
gianluca.dini@unipi.it

***Abstract*—Bitcoin-based smart city services are an ever increasing up-and-coming reality. For these services, simulating the Bitcoin blockchain is important to parametrize the system and tailor the costs and the economic incentives. In this regard, SimBlock simulator is the current state-of-the-art tool for blockchain simulations. Unfortunately, based on an up-to-date parametrization, SimBlock turns out not to simulate the mining of blocks. Furthermore, it does not simulate the incentive mechanism. These limitations strongly confine SimBlock's effective usage towards evaluating Bitcoin-based services relevant to many application contexts, including smart cities. To overcome these limitations, we propose an improved SimBlock's implementation. Upon it, we assess whether SimBlock can abstract the current Bitcoin blockchain. The experimental analysis shows that the proposed implementation can effectively simulate the current Bitcoin blockchain. Though, introducing relay network modelling in SimBlock should even improve the accuracy of the simulation.**

***Index Terms*—smart city, SimBlock, Bitcoin, blockchain, simulator, experimental analysis, validation**

I. Introduction

According to the United Nations, 68% of the world population will be living in cities by 2050 [1]. The significant increase in population and subsequent building development and urbanization raise a variety of economic and social problems. These problems affect to a large degree citizens' living conditions. In this respect, the *"smart city"* paradigm [2], [3] aims to optimize the use and exploitation of resources in a city, to provide high quality services and facilities for the citizens as well as to improve citizens' quality of life.

Citizens engage with smart city services in various ways, typically by using smartphones and mobile devices. Strengthening the interaction between citizens and smart city services is an important point for a smart city's development and evolution. In effect, engaging citizens with services and allowing inclusivity bring a wide range of benefits such as stronger communities, more empowered citizenry and better services themselves [4], [5]. On the contrary, failing to do so, leads to cities facing different negative effects, from an alienated population to the implementation of unpopular policies. Unfortunately, empowering citizens' engagement and enhancing citizens' interaction with smart cities services is a massive challenge. For this reason, administrations are looking at Information and Communication Technologies (ICT) with increasing interest, expecting them to play an important role. This is especially true for new and emerging technologies, such as: Internet of Things, big data, blockchain, artificial intelligence, data analytic and machine learning.

In this paper, we focus on the blockchain technology. Because of its properties, namely decentralization, transparency, pseudonymity and security, the blockchain offers a great potential to promote and increase the interaction between citizens and smart cities services. Specifically, we focus on the *Bitcoin blockchain* [6]. Nowadays, there already exist administrations embracing the Bitcoin blockchain, i.e., allowing bitcoin payments for smart city services [7]. Moreover, Bitcoin-based smart city services are an ever increasing up-and-coming reality. As of the first half of 2021, the plethora of Bitcoin-based smart city services concern the commercial and transportation domain [8].

For these services, simulating the Bitcoin blockchain is important to parametrize the system as well as to tailor the costs and the economic incentives (i.e., mining rewards). Costs include hardware, electricity, Internet connectivity and others related to mining [9]. As to the economic incentive, it is related to the probability to successfully mine a block. This, in turn, depends on mining *hash rate*, i.e., mining power. Though, it is important to highlight that successfully mining a block is only a necessary but not sufficient condition to get a mining reward. Indeed, a successfully mined block must also become part of the valid consensus chain, i.e., the block must not turn into an orphan block. This actually depends on different factors, primarily block size and network latency [10]. All in all, investigating the chance of getting a mining reward could be accomplished by accurately evaluating the performance of the blockchain protocol.

However, blockchain's intrinsic nature makes this performance evaluation task rather complex. Just to name but a few, from an architectural viewpoint, the blockchain consists of a decentralized, *large-scale* network of nodes. As to the *Bitcoin network*, large-scale means a network of the order of some tens of thousands of nodes scattered all over the world. Moreover, each node in the network is obliged to maintain its own copy of the blockchain. As to mid-April 2021, the space complexity of the Bitcoin blockchain is about 340 GigaByte (GB). To deepen on complexity of blockchain systems, and in particular of the Bitcoin blockchain, one can refer to [11], [12].

Conversely, a simulation-based approach becomes a promising evaluation means. In this regard, based on the recent comparative analysis about state-of-the-art blockchain simulators [13], we consider SimBlock by Aoki et al. [14]. Due to SimBlock's comprehensive set of input and output parameters, it ranks among the three (3) most advanced blockchain simulators over a set of twenty-seven (27) candidates [13]. Regarding Bitcoin-Simulator [15] and BlockSim [16], the other two state-of-the-art blockchain simulators, we do not take them into account for the following different reasons. As demonstrated in [13], Bitcoin-Simulator cannot accurately simulate the status quo due to recent extensions to the Bitcoin protocol, such as Compact Block Relay (CBR). BlockSim's main limitation is its inability to simulate the block propagation time, meaning that a *fixed input parameter* have to be instead specified.

Coming back to SimBlock, we properly tested the Java implementation (commit 06bd263) [17] according to a parametrization resembling the state of the Bitcoin blockchain in 2021. Because of the current *hash rate*, in the order of some hundred of Exahash per second (EH/s), it turned out that SimBlock is unsuitable for mining blocks. Furthermore, it must be noted that, at present, SimBlock does not simulate the incentive mechanism. These limitations significantly constrain SimBlock's effective usage towards evaluating Bitcoin-based services pertaining to a multitude of application domains, including the services related to the smart city's domain. To overcome these limitations, we propose a number of ad-hoc improvements. These are in the form of a new SimBlock's implementation. Specifically, we make the following contributions:

1) We propose and make publicly available an improved version of SimBlock [18]. The implementation allows to effectively simulating the mining of blocks based on current Bitcoin network hash rate.
2) We present an up-to-date experimental SimBlock's parametrization. Based on related research works, the latest comprehensive SimBlock's setup date back to the state of the Bitcoin blockchain in 2015.
3) Upon the new implementation, we experimentally validate whether SimBlock can realistically simulate the current Bitcoin blockchain. Specifically, we compare experimental results with Bitcoin real network statistics as well as with experimental results presented in previous research works [13], [19].
4) Based on key findings emerged from the comparisons, we investigate how to further improve SimBlock. By analyzing SimBlock's model of the blockchain network, we point out the lack of relay networks modelling.

The remaining of this article is organized as follow. We discuss related works in Section II. In Section III, we provide a concise technical background about SimBlock's mining operational mode. Based upon such information, in Section IV, first, we pinpoint the issues why SimBlock's original implementation is not suitable for mining of blocks. Then, we present the improvements we made to address those issues. In Section V, we present the up-to-date SimBlock's experimental parametrization and we comment on the methodology applied. In Section VI, we discuss our experimental results. Finally, Section VII concludes this article.

II. Related Works

Nagayama et al. [19] examined the impacts of CBR and Internet improvement on the Bitcoin block propagation time and fork rate for the period 2015 to 2019. In their study, they used SimBlock. As to Bitcoin blockchain in 2015, they used the same parametrization provided in SimBlock's original paper [14]. Regarding 2019, they computed and presented a SimBlock's updated setup. Though, regarding hash rate, they only specified *"Gaussian distribution"* which is the standard way SimBlock allocates the hash rate to each node.

Kanda and Shudo [20] proposed a method for adjusting Bitcoin block generation interval based on fork rate. To assess the technique, authors employed SimBlock. In this case, SimBlock's parametrization simulated a mix between the Bitcoin blockchain in 2015 and 2019. Values for input parameters related to year 2015 are the ones in [14]. Values for input parameters for the year 2019 are the ones in [19].

Otsuki et al. [21] highlighted the need for increasing Bitcoin's throughput by reducing the block propagation time and orphan rate via using relay networks. Since the considered relay networks were quite different from each other, but their ideal operation were basically common, authors decided to experimentally investigate the effects of an *ideal* relay network. Authors considered the state of the Bitcoin blockchain in 2019. Examining SimBlock's input parameter settings, the parametrization is that in [19]. Nevertheless, regarding the values related to block size and hash rate input parameters, they are instead those included in [14], therefore pertaining to the Bitcoin environment in 2015.

Paulavičius et al. [13] provided a systematic review of blockchain simulators. Subsequently, they presented an experimental analysis of state-of-the-art blockchain simulators, including SimBlock, and assessed the quality of the simulation results based on a parametrization reflecting the state of the Bitcoin blockchain in 2020. In practice, values for SimBlock's input parameters such as (download and upload) bandwidth, latency and hash rate are those shown in [19].

Summarizing, latest *comprehensive and unmixed* SimBlock's parametrization is that from 2015, i.e., the one included in in SimBlock's original manuscript [14]. As to individual input parameters characterizing the parametrization, the values related to the *hash rate* are the only ones which have not been updated for six years now.

After all, it is urgent to solve SimBlock's unsuitability to mine blocks, to provide an up-to-date SimBlock's parametrization and, based on it, to experimentally validate whether it can realistically simulate the current Bitcoin blockchain. In this work, we specifically aim to fulfill these goals.

III. Technical background on SimBlock's mining

To understand the motivations why SimBlock is unsuitable for mining blocks, we need to first discuss on its mining-related operational mode.

As first step (step #1), SimBlock assigns to each node *i*, *i=1,2,..N* where N is the number of nodes defined in the parametrization, a hash rate according to (1):

$$hashrate_i = max((r_i \times \sigma + \mu), 1) \tag{1}$$

where r_i is a pseudo-random Gaussian distributed value, with a mean equal to 0 and standard deviation equal to 1, μ is the average hash rate of each node and σ is the standard deviation of the hash rate of each node. When, in Section II, we discussed about values for the hash rate input parameter, we specifically referred to μ and σ.

Based on each node's hash rate, as second step (step #2), SimBlock computes the *network hash rate* according to (2):

$$hashrate = \sum_{i=1}^{N} hashrate_i \tag{2}$$

Upon (2), as third step (step #3), SimBlock calculates the difficulty of mining a block. Difficulty is a measure of how difficult it is to find a hash below a given target. Specifically, SimBlock computes the difficulty according to (3):

$$difficulty = hashrate \times interval \tag{3}$$

where *interval* is the block generation interval in milliseconds (ms). Then, SimBlock assigns this same difficulty to all nodes and it does not adjust it throughout the duration of the simulation (i.e., static difficulty). At this point, SimBlock designates a random node to create the *genesis* block. The genesis block is never mined hence it is always announced to the network.

Upon receiving a block, each node checks whether the block is valid and it is not orphan. If this is the case, the node adds the received block to its own chain.

Finally, based on its own hash rate, each node simulates the mining of a new block. Nodes perform this task if and only if the (static) difficulty is lower than a given *threshold*. The rationale behind this condition is that it must be indeed difficult to mine a block. If the condition is not met, it means the network has too high hash rate that it is trivial to find a hash below the given target. Stated in SimBlock's terminology, fourth step (step #4), mining a block is accomplished if and only if equation (4) holds:

$$\frac{1}{difficulty} > \frac{1}{threshold} \tag{4}$$

which is identical to what we stated before as soon as we consider the inverse of the two quantities. The threshold is set based on the real Bitcoin network hash rate. In SimBlock's original implementation, it resembles the Bitcoin network hash rate in 2015. Specifically, the threshold is set according to (5):

$$threshold = 2^{53} \cong 9 \times 10^{15} \tag{5}$$

IV. Improving SimBlock

In this section, first we outline the issues why SimBlock's original implementation is unsuitable for mining blocks. Initially, in Section IV-A, we provide a high-level description of the issues, then we go into the details. Afterwards, in Section IV-B, we discuss about the corresponding improvements we made.

A. Mining Issues

There exist two main issues for which SimBlock is not suitable for mining blocks. Both issues are caused by the Bitcoin network hash rate as compared to that in 2015. In Fig. 1, we show, on a logarithm scale, the trend of the Bitcoin network hash rate, in Petahash per second (PH/s), as a function of time from January 1st, 2015 to April 13th, 2021. As of April, 2021 Bitcoin network hash rate has increased by almost three orders of magnitude compared to early 2015, oscillating between 1.2×10^5 PH/s and 1.8×10^5 PH/s (i.e., between 120 and 180 EH/s).

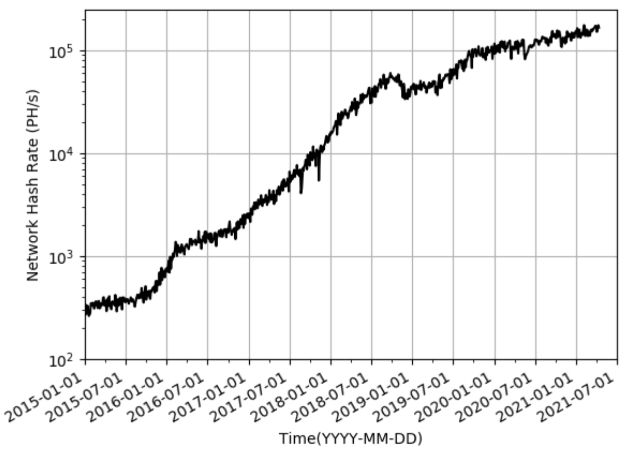

Fig. 1. Bitcoin Network Hash Rate in Petahash per second (PH/s) from January 1st, 2015 to April 13th, 2021.

The two issues concern step #4 discussed in Section III, namely computing the inverse of the difficulty and comparing it with the inverse of the threshold.

As to the first issue, given current Bitcoin network hash rate, computing the inverse of the difficulty equals 0. This is because the resulting difficulty is so high that calculating the inverse and saving into a Java *double* datatype, as required by the original implementation, makes it impossible to store as many significant decimal digits as required.

Regarding the second issue, it is because inequality (4) does not hold anymore. Particularly, the inverse of the current difficulty is less than the inverse of the threshold included in SimBlock's original implementation. We now argue on the two issue more in the detail.

About the first issue, Java double datatype allows 52 bits mantissa, hence about 16 decimal digits to be stored. However more decimal digits (19) need to be stored. Indeed, based on up-to-date values for node's hash rate (Section V - Table I) and relation (1), the *current maximum hash rate per node* is around 2×10^8 hash per milliseconds (H/ms)[1]. Considering a total of 9,853 current nodes (Section V - Table II) and equation (2) that equals to a *current maximum network hash rate* of about 2×10^{12} H/ms. Multiplying the previous value by interval, equation (3), equals to a *current maximum difficulty* of 1.2×10^{18} H. Computing the inverse of such a number requires (at least) 19 decimal digits in order for the result to be nonzero. However, there is only space for 16 decimal digits.

Regarding the second issue, considering the *current maximum difficulty*, namely 1.2×10^{18} H, and *threshold* value included in SimBlock's original implementation, equation (5), it turns out that inequality (4) indeed no longer applies.

B. Improvements

To address the first issue, we modified the original implementation switching from primitive Java double datatype to arbitrary-precision signed decimal, i.e., *BigDecimal*. Given that arithmetic and comparison operations on BigDecimal (and BigInteger) require operands of the same type, such a modification implied reworking a large portion of the code. Finally, to compute the logarithm of BigDecimal numbers, we introduced functions on numerical computing [22] based on Newton's method.

To tackle the second issue, thereby allowing SimBlock performing the mining of blocks, we updated the *threshold* value accordingly. Specifically, we considered as a baseline the current Bitcoin network hash rate. As per Fig. 1, this means setting a threshold of about 2^{63} H. Then, we left some room in the light of: i) an increase of the total number of miners; and ii) improvements in mining equipment. Envisioning a new network hash rate grow of about three orders of magnitude, we changed the threshold value according to (6):

$$threshold = 2^{70} \cong 1.2 \times 10^{21} \tag{6}$$

Finally, we extended SimBlock so as to introduce the functionality to save simulation results (i.e., block propagation time). Although, the original implementation is supposed to offer such a feature, it is actually not the case as per corresponding open issue [23]. Moreover, we make our solution publicly available [18] to both researchers and developers community.

[1]The minimum hash rate per node should actually be considered in order for the inverse of difficulty to be maximum. However, since it leads to the same final consideration, we consider the maximum hash rate per node so as to reuse the computations later on.

V. SimBlock's Experimental Parametrization

We validate SimBlock according to the new implementation. Validation dictates SimBlock's input parameters to reflect the present conditions of the Bitcoin blockchain. In this section, we present the up-to-date SimBlock's experimental parametrization, i.e., up-to-date values for SimBlock's input parameters. Furthermore, we describe the methodology applied in order to compute those values.

As far as the parametrization is concerned, blockchain simulators typically distinguish between network-layer, consensus-layer and incentives-layer parameters [13]. As already mentioned, SimBlock does not simulate the incentive mechanism, hence it only consists of a network-layer and a consensus-layer. In Section V-A, we present consensus-layer parameters, whereas in Section V-B we detail on network-layer parameters. Values for those parameters are statistics computed on data related to the real Bitcoin blockchain in the period from March 1st, 2021 to April 13th, 2021 (i.e., 43 days in total). The reader may refer to [14], [17] to deepen on the rationale of those parameters.

A. Consensus-Layer Parameters

In Table I we show (only) the updated statistics for consensus-layer parameters with respect to SimBlock's original implementation. The average hash rate of a node (i.e., *AVERAGE_MINING_POWER*) and the standard deviation of the hash rate of a node (i.e., *STDEV_MINING_POWER*), respectively correspond to μ and σ presented in equation (1).

TABLE I
Consensus-Layer parameter for SimBlock

AVERAGE_MINING_POWER Unit: Hash per millisecond	190,539,325
STDEV_OF_MINING_POWER Unit: Hash per millisecond	11,380,327
END_BLOCK_HEIGHT Unit: Dimensionless	6,479

As to *average/stdev_mining_power* they are computed as follows: i) Get Bitcoin network hash rate distribution among mining pools; ii) Compute average Bitcoin mining pool's hash rate; iii) Compute average Bitcoin mining pool's nodes; iv) Get Bitcoin network hash rate per day; v) From iv), compute average/stdev. Bitcoin network hash rate per day; vi) From v), compute average/stdev. Bitcoin network hash rate per second; vii) From vi), ii) and iii) compute average/stdev. Bitcoin network hash rate per second per node; and viii) Convert vii) from EH/s to H/ms. Let us now verify that the average network hash rate indeed results in 162 EH/s per day, that is the average network hash rate over the considered time-frame. The *average_mining_power* can be written as 190×10^9 H/s. Multiplied that by 86,400 (i.e., the number of seconds in a day) equals to 1.65×10^{16} H/s per day per node. Multiplied that by 9,853 nodes equals to 162 EH/s per day.

Regarding *end_block_height*, it is calculated as the number of blocks mined between March 1st, 2021 (block's height: 672,621) and April 13th, 2021 (block's height: 679,100).

B. Network-Layer Parameters

In order for the simulated network to resemble the real Bitcoin network, we exploited the *Bitnodes's* Application Programming Interface (API) endpoint named *"List Snapshots"* [24]. The API provides the list of all Bitcoin reachable nodes, together with their associated information (e.g., IP address, version of Bitcoin protocol, country of the node, etc). The list consists of 10,000 network *snapshots*, i.e., 36GB of data. We parsed data and compute relevant statistics. In Table II, we present (only) updated statistics as compared to the original.

TABLE II
NETWORK-LAYER PARAMETER FOR SIMBLOCK

REGION_DISTRIBUTION Unit: Dimensionless	{18.8616%, 59.7868%, 4.33491%, 13.6794%, 1.58874%, 1.74855%}
DOWNLOAD_BANDWIDTH Unit: bit per second	$\{41 \times 10^6, 39 \times 10^6, 21 \times 10^6, 20 \times 10^6, 29 \times 10^6, 46 \times 10^6, 6 \times 10^6\}$
UPLOAD_BANDWIDTH Unit: bit per second	$\{13 \times 10^6, 17 \times 10^6, 6 \times 10^6, 13 \times 10^6, 18 \times 10^6, 17 \times 10^6, 6 \times 10^6\}$
LATENCY Unit: milliseconds	{30, 86, 187, 230, 158, 152, 86, 12, 240, 236, 227, 253, 187, 240, 41, 327, 317, 327, 230, 236, 327, 87, 51, 151, 158, 227, 317, 51, 15, 113, 152, 253, 327, 151, 113, 14}
NUM_OF_NODES Unit: Dimensionless	9,853
CBR_USAGE_RATE Unit: Dimensionless	0.89
CHURN_NODE_RATE Unit: Dimensionless	0.975
BLOCK_SIZE Unit: Byte	1,326,097

As to *region distribution*, it is computed as the ratio of nodes in each SimBlock's region to unique nodes in the 10,000 snapshots. Since 14.90% of the unique nodes turned out operating under the Tor network or in South-Africa, we exploited Bitcoin-Simulator to split such percentage between available regions based on actual proportions.

About *download and upload bandwidth*, they are computed based on nodes' countries. Specifically, first we retrieved download and upload bandwidth of each country from Testmy.net [25], so we computed the average of the (download and upload) bandwidths of the countries belonging to the same SimBlock's region.

Regarding *latency*, we retrieved it between pairs of SimBlock's regions from Verizon [26] and Wondernetwork [27].

On *num_of_nodes*, first we computed the total nodes per snapshot, then we took the mean value.

As regards *cbr_usage_rate*, we calculated the ratio of nodes running the Bitcoin protocol implementing CBR to total nodes per snapshot. Thus, we computed the average value.

With respect to *churn_node_rate*, it is the difference between one (1) and the ratio of nodes always connected to the Bitcoin network to unique nodes in the 10,000 snapshots.

Finally, regarding *block_size* we gathered the average Bitcoin block size per day and then we computed the mean value.

VI. VALIDATION RESULTS AND DISCUSSION

Upon the provided parametrization, we assess to what extent the proposed implementation is able to abstract the current Bitcoin blockchain. Specifically, we experimentally computed the: i) 50-th percentile of the block propagation time t_b^{50} (in seconds); ii) 90-th percentile of the block propagation time t_b^{90} (in seconds); and iii) average orphan block rate r_o (in percent) considering a 99% confidence level.

Then, we compared experimental results with Bitcoin real network statistics in 2021 as well as with results presented in previous research works [13], [19]. For a fair comparison, we also present the average block size s_b in MegaByte (MB).

Regarding the experimental part, we performed 40 independent simulations, each under the same setup on a machine with 2 x AMD EPYC 7702 64-Core @3.35GHz Processors, 1 TB RAM DDR4-3200, and running Ubuntu 20.04.2 LTS.

As to real network statistics, they are computed from data coming from Bitcoin monitoring websites, respectively [28] and [29]. As to the 50-th and 90-th block propagation time, it is computed as the *median* of the data. As to orphan rate, it is the fraction of orphan blocks to the total number of blocks in the considered time-frame. Table III summarizes the key findings.

TABLE III
COMPARISON OF THE 50-TH AND 90-TH PERCENTILE OF THE BLOCK PROPAGATION TIME (t_b)(IN SECONDS), AND THE ORPHAN BLOCK RATE (r_o) (IN PERCENT) AS A FUNCTION OF THE BLOCK SIZE (s_b)(IN MEGABYTE), IN THE BITCOIN NETWORK AND IN SIMBLOCK SIMULATOR.

Context	**Year**	s_b	t_b^{50}	t_b^{90}	r_o
Bitcoin (real network)	2019	1.00	0.40	2.35	N/R
Nagayama et al. [19]	2019	1.00	1.34	2.36	N/R
Bitcoin (real network)	2020	1.22	0.50	3.3	0.06
Paulavičius et al. [13]	2020	1.22	1.30	2.10	0.19
Bitcoin (real network)	2021	1.33	0.63	2.83	0.09
Our Work	2021	1.33	1.68	3.50	0.31 ± 0.03

About t_b^{50}, our experimental result differs only by 1.05 seconds with respect to real network statistic. In [13] the delta is 0.8 seconds. In [19] it is 0.94 seconds. As to t_b^{90}, our experimental value is 0.67 seconds larger than the real one. Whereas in [19], the experimental value is 0.01 second higher. In [13], instead, it is 1.2 seconds lower than the real one. As to the latter case, this is likely due to authors computing the *mean* value for real network statistics, i.e., values are affected by outliers. Regarding r_o, in our simulations it differs from the real value by $0.22 \pm 0.03\%$, in contrast to the 0.13% of [13]. In general, the simulated results turned out to be larger than the real values throughout the last three years.

The motivation is very likely due to SimBlock's lack of relay networks modelling. In fact, by checking the model of the simulated peer-to-peer network, we figured out that each node simply chooses its own peers at random until a given number of connections are established. Eventually, a random-topology network with no relay nodes gets created.

However, relay networks allow to quickly spread blocks among nodes, improving the block propagation time and reducing the orphan block rate. Even if the participation rate is as low as 3%, simulation results in [21] show that the propagation time is reduced to approximately 77% of the original value, whereas the orphan block rate is shortened to around 85% of the original value. These effects become stronger as relay network participation rate increases [21].

We claim that it is indeed a matter of participation rate when considering the increased delta between the experimental and the real value for t_b^{50} and t_b^{90} in 2021 than in the past. In effect, in 2021 most miners participate in Fast Internet Bitcoin Relay Engine (FIBRE) relay network [30]. On the other hand, Falcon relay network participation rate in 2019 was as low as 2.65% [19]. Conversely, we state that if participation rate is not high, t_b^{90} is not affected as much as t_b^{50}. Evidence of this is shown by the greater difference between the experimental and the real value of t_b^{90} in 2021 (i.e., 0.68 seconds) compared to 2019 (i.e., 0.01 seconds). Finally, given that current participation rate is higher than ever before, it is reasonable that the experimental value for r_o in 2021 is larger than the one in [13].

Thus, we can conclude that our model correctly simulates the current Bitcoin blockchain. Nonetheless, relay networks modelling should improve the accuracy of the simulation.

VII. Conclusions

To tailor the costs and the economic incentives regarding Bitcoin-based smart city's services, but not only, simulating the Bitcoin blockchain is of vital importance. By testing the state-of-the-art blockchain simulator called SimBlock (commit 06bd263), based on Bitcoin network hash rate in 2021, we spot that it is unsuitable to mine blocks. In this work, we proposed an improved SimBlock's version alongside an up-to-date parametrization. In this regard, the experimental analysis showed that the proposed solution can effectively simulate the current Bitcoin blockchain. Though, introducing relay network modelling in SimBlock should actually increase the accuracy of the simulation. All in all, complementing the proposed solution with relay networks modelling concretely lay the foundations to SimBlock's future extensions, especially simulating the incentive mechanism which, at present, is missing.

References

[1] United Nations. "World urbanization prospects: the 2018 revision;". [Online]. Available: https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf [accessed: June 18, 2021].

[2] E. Tabane, S. M. Ngwira, and T. Zuva, "Survey of smart city initiatives towards urbanization," in Proc. IEEE ICACCE, Durban, South Africa, Nov. 2016, pp. 437–440.

[3] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in Proc. ACM dg.o, College Park, MD, USA, 2011, pp. 282–291.

[4] C. Manville, R. Europe, J. Millard, D. T. Institute, and A. Liebe, "Mapping Smart cities in the EU". [Online]. Available: https://op.europa.eu/it/publication-detail/-/publication/78882e80-fc4a-4a86-9c39-2ad88ab89f9b [accessed: June 18, 2021].

[5] The Young Foundation. "Understanding barriers to engagement". [Online]. Available: https://youngfoundation.org/wp-content/uploads/2013/06/Understanding_barriers_to_engagement.pdf [accessed: June 18, 2021].

[6] S. Nakamoto "Bitcoin: A peer-to-peer electronic cash system". [Online]. Available: https://bitcoin.org/bitcoin.pdf [accessed: June 11, 2021].

[7] "Top 10 cities where you can pay the bill in Bitcoin". [Online]. Available: https://www.itworldcanada.com/slideshow/top-10-cities-where-you-can-pay-the-bill-in-bitcoin [accessed: June 21, 2021].

[8] "7 Uses of Blockchain and Crypto in Smart Cities.". [Online]. Available: https://smartcityhub.com/technology-innnovation/7-uses-of-blockchain-and-crypto-in-smart-cities/. [accessed: June 17, 2021].

[9] G. Poxleitner, "Reducing Mining Costs and Value Optimization", [Online]. Available: https://www.srk.com/en/publications/operating-cost-forminers [accessed: June 21, 2021].

[10] C. Qiu, H. Yao, C. Jiang, S. Guo and F. Xu, "Cloud Computing Assisted Blockchain-Enabled Internet of Things," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2019.2930259.

[11] F. Tschorsch e B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies", IEEE Commun. Surv. Tutor., vol. 18, n. 3, pagg. 2084–2123, thirdquarter 2016, doi: 10.1109/COMST.2016.2535718.

[12] M. Conti, E. Sandeep Kumar, C. Lal, e S. Ruj, "A Survey on Security and Privacy Issues of Bitcoin", IEEE Commun. Surv. Tutor., vol. 20, n. 4, pagg. 3416–3452, Fourthquarter 2018, doi: 10.1109/COMST.2018.2842460.

[13] Paulavičius, S. Grigaitis, and E. Filatovas, 'A Systematic Review and Empirical Analysis of Blockchain Simulators', IEEE Access, vol. 9, pp. 38010–38028, 2021, doi:10.1109/ACCESS.2021.3063324.

[14] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, 'SimBlock: A Blockchain Network Simulator', in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Apr. 2019, pp. 325–329. doi: 10.1109/INFCOMW.2019.8845253

[15] A. Gervais and V. Glykantzis. (2016). "Bitcoin-Simulator". [Online]. Available: https://github.com/arthurgervais/Bitcoin-Simulator [accessed: June 25, 2021].

[16] M. Alharby and A. van Moorsel, 'BlockSim: A Simulation Framework for Blockchain Systems', ACM SIGMETRICS Perform. Eval. Rev., vol. 46, no. 3, pp. 135–138, Jan. 2019, doi: 10.1145/3308897.3308956.

[17] "dsg-titech/simblock". [Online]. Available: https://github.com/dsg-titech/simblock [accessed: June 21, 2021].

[18] "SimBlock Blockchain Simulator". [Online]. Available: https://github.com/Unipisa/SimBlock [accessed: July 22, 2021].

[19] R. Nagayama, K. Shudo, and R. Banno, 'Identifying Impacts of Protocol and Internet Development on the Bitcoin Network', ArXiv191205208 Cs, Jun. 2020, Accessed: May 10, 2021. [Online]. Available: http://arxiv.org/abs/1912.05208

[20] R. Kanda and K. Shudo, "Block Interval Adjustment Toward Fair Proof-of-Work Blockchains," 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW), 2020, pp. 1-6, doi: 10.1109/ICDEW49219.2020.00-15.

[21] K. Otsuki, R. Banno and K. Shudo, "Quantitatively Analyzing Relay Networks in Bitcoin," 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 214-220, doi: 10.1109/Blockchain50366.2020.00034.

[22] "Compute the natural logarithm of BigDecimal", Wikipedia. [Online]. Available: http://www.java2s.com/example/java/java.math/compute-the-natural-logarithm-of-bigdecimal-x-to-a-given-scale-x-0.html [accessed: June 25, 2021].

[23] "dsg-titech/simblock - Standard output and out.txt". [Online]. Available: https://github.com/dsg-titech/simblock/issues/16 [accessed: June 25, 2021].

[24] "Global Bitcoin nodes distribution". [Online]. Available: https://bitnodes.io/api/ [accessed: June 25, 2021].

[25] "Internet Speed Test". [Online]. Available: https://testmy.net/ [accessed: June 25, 2021].

[26] "Verizon Network Performance: Weekly Latency Data". [Online]. Available: https://www.verizon.com/business/solutions/business-continuity/weekly-latency-statistics/ [accessed: June 25, 2021].

[27] "Global Ping Statistics". [Online]. Available: https://wondernetwork.com/pings [accessed: June 25, 2021].

[28] "Bitcoin Network Monitor". [Online]. Available: https://www.dsn.kastel.kit.edu/bitcoin/data/invstat.gpd [accessed: June 26, 2021].

[29] "Fork Monitor—BTC Stale Block Candidates". [Online]. Available: https://forkmonitor.info/feeds/stale_candidates/btc.rss [accessed: June 22, 2021].

[30] Lehar, Alfred and Parlour, Christine A., Miner Collusion and the BitCoin Protocol (March 22, 2020). [Online]. Available: http://dx.doi.org/10.2139/ssrn.3559894 [accessed: June 28, 2021].