








Simulating Blockchain Consensus Protocols in Julia: Proof of Work vs Proof of Stake

Georgios Drakopoulos¹(✉) , Eleanna Kafeza² , Ioanna Giannoukou³ ,
Phivos Mylonas¹ , and Spyros Sioutas³ 

¹ Humanistic and Social Informatics Lab, Ionian University, Corfu, Hellas
`{c16drak,fmylonas}@ionio.gr`

² College of Technological Innovation, Zayed University, Dubai, UAE
`eleana.kafeza@zu.ac.ae`

³ University of Patras, Patras, Hellas
`igian@upatras.gr, sioutas@ceid.upatras.gr`

Abstract. Consensus protocols constitute an important part in virtually any blockchain stack as they safeguard transaction validity and uniqueness. This task is achieved in a distributed manner by delegating it to certain nodes which, depending on the protocol, may further utilize the computational resources of other nodes. As a tangible incentive for nodes to verify transactions many protocols contain special reward mechanisms. They are typically inducement prizes aiming at increasing node engagement towards blockchain stability. This work presents the fundamentals of a probabilistic blockchain simulation tool for studying large transaction volumes over time. Two consensus protocols, the proof of work and the delegate proof of stake, are compared on the basis of the reward distribution and the probability bound of the reward exceeding its expected value. Also, the reward probability as a function of the network distance from the node initiating the transaction is studied.

Keywords: Blockchain simulation · Consensus protocols · Proof of work · Proof of state · Stakeholder delegate · Behavioral economics

1 Introduction

After the introduction of Bitcoin research interest focused not only on cryptocurrencies but also on the consensus protocols used to verify transactions. The latter are essential in achieving reward fairness, even approximately, and trust in the respective cryptocurrency by actively engaging nodes. A blockchain with reinforced trust in addition to the ability of global secure payments independent of the control of external parties is more attractive to potential stakeholders.

Since blockchain relies on massive peer-to-peer (p2p) network technology, it is difficult to predict the exact action course during a transaction sequence

as well as the resulting blockchain state. One way to overcome this limitation is to probabilistically simulate the blockchain including the consensus protocol, network rewards, and the nodes themselves in terms of computing power.

The primary research objective of this conference paper is a highly parameterized node-level probabilistic blockchain simulation tool. As a concrete example, it has been applied to two common blockchain consensus protocols, namely proof of work (PoW) and proof of stake (PoS), and the results are analyzed.

The remainder of this conference paper is structured as follows. In Sect. 2 the recent scientific literature is briefly reviewed. Simulation is described in Sect. 3. The results are outlined in Sect. 4, while in Sect. 5 possible future research directions are given. Capital italic letters represent random variables and capital boldface letters matrices. In function definitions parameters follow arguments after a semi-colon. Finally, the notation is summarized in Table 1.

Table 1. Notation of this work.

Symbol	Meaning	First in
\triangleq	Definition or equality by definition	Eq. (1)
$E[\mathcal{X}]$	Mean value of random variable \mathcal{X}	Eq. (8)
$\text{Var}[\mathcal{X}]$	Variance of random variable \mathcal{X}	Eq. (9)
$\text{prob}\{\Omega\}$	Probability of event Ω occurring	Eq. (3)
$\langle f g \rangle$	Kullback-Leibler divergence for f and g	Eq. (22)
$f^{(n)}(x)$	n -th derivative of function $f(x)$	Eq. (9)
$i \rightarrow j / (i \rightarrow j)^p$	Path of node i to j of any length/length p	Eq. (7)

2 Previous Work

Consensus protocols are instrumental in any blockchain [21]. Among the most widespread ones are proof of work [1] and proof of stake [19]. A recent survey is [6]. Algorithmic means for defending against rogue and powerful miners [15]. Game theoretic attacks for proof of work are analyzed in [4]. Blockchain applications include smart contracts [18], payments [13], and medical records [9]. Behavioral economics focus on the cognitive mechanisms for decision making [3] like cognitive bias [11], cognitive dissonance [12], and inducement prizes [10]. Such techniques have increased engagement in cultural content delivery [5] and prolonged the visiting times in cultural portals [8]. An important effect of consensus protocols is that they reinforce Web trust in a distributed and stateless environment where parties have no *a priori* reason to trust each other [20]. The latter is critical for Web services including e-commerce [2], database architecture selection [14], recommendation engines [16], and finding trusted candidates in LinkedIn [7]. Recently blockchains have been used in sensor networks [17].

3 Simulation of Consensus Protocols

Interested parties and stakeholders typically consider joining a blockchain in order to obtain certain rewards, whether tangible or intangible. Still, this work will deal in general with (*network*) *rewards* without further specialization.

This simulation aims to address the following fundamental questions:

- The reward distribution after a long transaction sequence, especially in terms of reward fairness and final node wealth distribution.
- How the costs of joining a blockchain and verifying transactions influence the wealth distribution and whether negate any initial incentives.
- The transaction initiation distribution after a large number of transactions. In the long run it reveals the true chances a node has for collecting rewards.

Note that the actual values of both the parameters discussed below and the internal fine tuning options are given in Table 2. The primary parameters are:

- The number of blockchain nodes N_0 . It is the number of clients participating to the p2p network, each performing an identical set of roles.
- The processing power P_i , namely the number of processors and their power. They are identical, with factors like paging and caching policy ignored.
- The link capacity $C_{i,j}$ ignoring factors such as network technology, signal to noise (SNR) ratio, stack size, number of interfaces, and routing overhead.
- The node failure probability p_0 . It is independent of local resiliency technologies like backup power sources, network drives, and RAID arrays.

The simulation consists of R_1 runs with the blockchain topology changing after each run. Each such run has R_0 rounds and each round has three steps:

- The initial transaction request and its propagation through the network.
- The transaction verification according to the consensus protocol.
- The verification propagation through the p2p network.

The number of rounds is determined as in Eq. (1). Each of the N_0 nodes is selected uniformly for a transaction. This in conjunction with R_0 implies that each node has on average γ_0 chances to collect network rewards.

$$R_0 \triangleq \lceil \gamma_0 N_0 \rceil \quad (1)$$

Blockchain topology plays a central role. In each run I_0 randomly selected links are created between the N_0 nodes. The density ρ_0 is defined as in (2):

$$I_0 \triangleq \lceil \rho_0 N_0 \rceil \quad (2)$$

Each node starts with a fixed amount of W_0 network reward units. In each round a node, called the *initiator* for brevity, requests a transaction claiming a randomly selected amount w_i . The latter is chosen uniformly in the interval between $w_l W_0$ and $w_h W_0$. The uniform distribution expresses the generic nature

of the network rewards and it is by no means fit for every case. For instance, when the rewards are rare, the Poisson distribution might be more appropriate.

Network resiliency is expressed as the node failure probability p_0 . In the context of the proposed simulation p_0 is the probability of a node failing to receive the request, process it, or send the reply. As each node operates independently, the probability that k nodes fail simultaneously is given by Eq. (3):

$$\pi_k \triangleq \text{prob}\{k \text{ failures}\} = \binom{N_0}{k} p_0^k (1-p_0)^{N_0-k} \quad (3)$$

Clearly π_k in the above equation follows a binomial distribution defined over a finite population N_0 with a success probability $1-p_0$. When p_0 is very low, as it was chosen here, then Eq. (3) can be approximated as in (4):

$$\pi_k \approx \frac{(p_0 N_0)^k}{k!} e^{-p_0 N_0} = \frac{\lambda_0^k}{k!} e^{-\lambda_0}, \quad \lambda_0 \triangleq p_0 N_0 \quad (4)$$

Equation (4) is a Poisson distribution. This approximation is derived by (5). Each of the k simultaneous failures in every simulation round is independent and local as in a real p2p network there is no global failure knowledge.

$$\begin{aligned} \binom{N_0}{k} p_0^k &\approx \frac{N_0^k}{k!} p_0^k = \frac{(p_0 N_0)^k}{k!} \\ (1-p_0)^{N_0-k} &\approx (1-p_0)^{N_0} \approx e^{-p_0 N_0} \end{aligned} \quad (5)$$

The $N_0 \times N_0$ random symmetric link capacity matrix \mathbf{C} is an instrumental parameter. Symmetry implies that the capacity in bits per second (bps) along any link is the same in both directions. The time $t_{i,i'}$ to transmit a packet of length L_0 in bits, complete with protocol headers and trailers, assuming that no failures or retransmission attempts take place is shown in Eq. (6):

$$t_{i,i'} = \frac{L_0}{\mathbf{C}_{i,i'}}, \quad \text{link } (i, i') \text{ exists} \quad (6)$$

The distribution of link capacity $\mathbf{C}_{i,j}$ is log-normal with its variance taking into account equipment technology, geographical distribution, and configuration differences among other factors. The network packets carrying the transaction verification request and the corresponding confirmation have length L_r and L_v respectively. Typically L_r is long since it contains the information necessary to verify the transaction. Additionally, both packets can be salted with cryptographic data so that neither a random or fake transaction request can be generated nor a phony verification. Capacity essentially imposes a network topology where the minimum distance between nodes i and j is the minimum weighted sum over all connecting paths $i \rightarrow j$. The minimum time $\mathbf{T}_{i,j}$ for a package is (7), which lends itself among others to dynamic programming solutions:

$$\mathbf{T}_{i,j}(L_0) \triangleq \min_{i \rightarrow j} \left[\sum_{(k,k')} t_{k,k'} \right] = \min_{i \rightarrow j} \left[\sum_{(k,k')} \frac{L_0}{\mathbf{C}_{k,k'}} \right] \approx \min_{(i \rightarrow j)^p} \left[\sum_{(k,k')} \frac{L_0}{\mathbf{C}_{k,k'}} \right] \quad (7)$$

In the simulation $\mathbf{T}_{i,j}$ is used, but in a real blockchain stack only local routing information in a neighborhood of depth p is used as in the right hand side of (7). The packet transmission times $\mathbf{T}_{i,j}$ depends on the link capacity distribution as shown in (6). Since the latter is stochastic, so are $\mathbf{T}_{i,j}$. This leads to the question of what can be deduced about them given the probabilistic properties of \mathbf{C} .

The mean values $E[\mathcal{T}_{i,j}]$ of $\mathbf{T}_{i,j}$ can be computed as follows. If r.vs \mathcal{X} and \mathcal{Y} are connected through the measurable, not necessarily invertible function $h(\cdot)$, and if $f_X(\cdot)$ is the probability density function (pdf) of \mathcal{X} , then (8) holds:

$$E[\mathcal{Y}] \triangleq E[h(\mathcal{X})] = \int_{\Omega_f} h(x) f_X(x) dx \quad (8)$$

Concerning the variance $\text{Var}[\mathcal{T}_{i,j}]$ the answer is not straightforward as the variance is invariant in the general case only to linear transforms. Thus an estimate by the *delta method* of Eq. (9) will be used which relies on a first order Taylor approximation of $\text{Var}[\mathcal{Y}]$ around $E[\mathcal{X}]$. Specifically:

$$\text{Var}[\mathcal{Y}] \approx \text{Var}[\mathcal{X}] \left(h^{(1)}(E[\mathcal{X}]) \right)^2 \quad (9)$$

Given (8) and (9) the mean and variance of $\mathcal{T}_{i,j}$ are as in Eq. (10):

$$E[\mathcal{T}_{i,j}] \triangleq \frac{L_0}{E[\mathcal{C}_{i,j}]} \quad \text{and} \quad \text{Var}[\mathcal{T}_{i,j}] \approx \text{Var}[\mathcal{C}_{i,j}] \frac{L_0^2}{E[\mathcal{C}_{i,j}]^4} \quad (10)$$

Once a packet arrives at the destination node, it will be processed. Again time is a critical factor, but it is computed in a different way. The processing power P_i for each node is determined by the number and type of processors. The memory is assumed to be sufficiently high so that it does not interfere with thread or processor parallelism. Specifically, the processing power is given by Amdahl's law where each of the p_i processors is assumed to have s_i cores for a total of $P_i = p_i s_i$ cores. In this case Amdahl's speedup becomes:

$$\zeta_i \triangleq \frac{1}{(1 - \epsilon_0) + \frac{\epsilon_0}{P_i}} = \frac{1}{(1 - \epsilon_0) + \frac{\epsilon_0}{p_i s_i}} \quad (11)$$

In Eq. (11) $1 - \epsilon_0$ is the part of the verification transaction which cannot be parallelized and it is the same across nodes and runs. It has been determined based on observations and literature recommendations [19, 21]. Also, p_i and c_i are uniformly selected among the respective number of possible choices.

The round trip time r_i for node i from initiator i^* is computed as in (12):

$$r_i \triangleq \begin{cases} \mathbf{T}_{i,i^*} (L_r + L_v) + T_b / \zeta_i, & \text{PoW/PoS I} \\ \mathbf{T}_{i,i^*} (L_r + L_v) + \max[\mathbf{T}_{i,j} (L_r + L_v) + T_b / \zeta_j], & \text{PoS II} \end{cases} \quad (12)$$

The first branch in (12) represents the time required for the request packet to reach i , to process it, and return the verification to i^* for PoW. This is mechanism

also works for the original version PoS of PoS I, although the verifiers are selected in a specific probabilistic way. The second branch is the time under the delegate version of PoS or PoS II required for the request packet to reach i and be relayed to the witnesses, processing by the latter, return of the verifications from the witnesses to i , and the subsequent retransmission to i^* . T_b is the base task execution time which remains constant for every node and across runs.

3.1 Proof of Work

Under PoW the initiator is required to transmit a transaction verification request. The latter is approved only when a sufficient number of responder nodes approves the information the initiator has included in the request.

Algorithm 1. Consensus protocol simulation.

Require: The parameters of Table 2.

Ensure: The simulation objectives are achieved.

```

1: for all runs in the simulation do
2:   select topology, capacities, processors, and cores
3:   for all rounds in the current run do
4:     select initiator node and reward
5:     select responders or verifiers [and witness nodes]
6:     compute times and rank nodes based on time
7:   end for
8: end for
9: return

```

3.2 Proof of Stake

PoS relies on the principle that nodes which have accumulated more rewards are also more eager to contribute to the blockchain stability. In the original version (PoS I) η_0 verifier nodes are selected with a probability proportional to their rewards. To prevent rewards from being collected by a small group of nodes, in the delegate version (PoS II) each verifier contracts a witness, selected with a probability proportional to its computing power as shown in Eq. (13):

$$\delta_i \triangleq \text{prob} \{i \text{ is witness}\} = \frac{P_i}{\sum_{k=1}^{N_0} P_k} \quad (13)$$

Verifiers and witnesses each get a fraction τ_0 of the reward. The selection mechanism gives a chance to high power nodes in addition to the wealthier ones. Advanced PoS versions of take into account network connections or memory size.

4 Results

4.1 Rewards

In Table 2 the simulation parameters and their actual values are shown.

Table 2. Experimental setup.

Parameter	Value	Parameter	Value
Number of nodes N_0	16384	Delegate selection prob. δ_i	Eq. (13)
Node failure probability p_0	0.05	Rounds coefficient γ_0	16
Network density ρ_0	0.6	Percentage of parallelism ϵ_0	0.85
Number of links I_0	Eq. (2)	Model fit method	ML
Initial reward distribution	Fixed	Verification time T_b	Eq. (12)
Initial balance W_0	1000	Witness reward fraction τ_0	0.005
Reward limits w_l/w_h	0.05/0.1	Link capacity distribution	Lognormal
Simulation rounds R_0	Eq. (1)	Request packet length L_r	2048
Node selection distribution	Uniform	Verification packet length L_v	512
Number of delegates η_0	51	Distribution of processors p_i	Uniform
Number of runs R_1	10000	Distribution of cores s_i	Uniform

An important description of a blockchain is its reward distribution. If it is balanced enough, it may appeal to potential stakeholders seeking security. If not, it may attract high risk takers. Thus, each distribution is compatible with different behavioral stakeholder profiles, which is a key design factor.

For each node the rewards over each run and round are averaged, eliminating thus the effect of topology and keeping that of consensus protocol. To construct the empirical rewards distribution B_0 bins as in Eq. (14) will be used. This allows a large amount of bins each with a statistically safe numbers of samples.

$$B_0 \triangleq 0.25 \left\lceil \sqrt{N_0} \right\rceil \quad (14)$$

The resulting empirical mean reward distribution in logarithmic scale is shown in Fig. 1, which suggests a power law and that PoS seems to distribute network rewards more evenly than PoW.

A second way to decide whether a particular blockchain is worth joining is the probability of deviating from the expected reward. If the latter is high, then the payoff for the initial cost may be significant. The Chebyshev inequality of (15) provides upper bounds for this probability in the scale of standard deviations:

$$\text{prob} \left\{ |\mathcal{R} - \mathbb{E}[\mathcal{R}]| \geq \xi_0 \sqrt{\text{Var}[\mathcal{R}]} \right\} \leq \frac{1}{\xi_0^2} \quad (15)$$

Table 3. Chebyshev upper bounds.

$\xi_0 \sqrt{\text{Var} [\mathcal{R}]}$	$0.2 E [\mathcal{R}]$	$0.3 E [\mathcal{R}]$	$0.5 E [\mathcal{R}]$	$0.75 E [\mathcal{R}]$	$E [\mathcal{R}]$
PoW	$2.83 \cdot 10^{-1}$	$1.45 \cdot 10^{-1}$	$7.09 \cdot 10^{-2}$	$1.08 \cdot 10^{-2}$	$9.81 \cdot 10^{-3}$
PoS I (original)	$1.83 \cdot 10^{-1}$	$1.17 \cdot 10^{-1}$	$4.18 \cdot 10^{-2}$	$8.14 \cdot 10^{-3}$	$7.79 \cdot 10^{-3}$
PoS II (delegate)	$1.22 \cdot 10^{-1}$	$9.62 \cdot 10^{-2}$	$3.26 \cdot 10^{-2}$	$7.66 \cdot 10^{-3}$	$6.33 \cdot 10^{-3}$

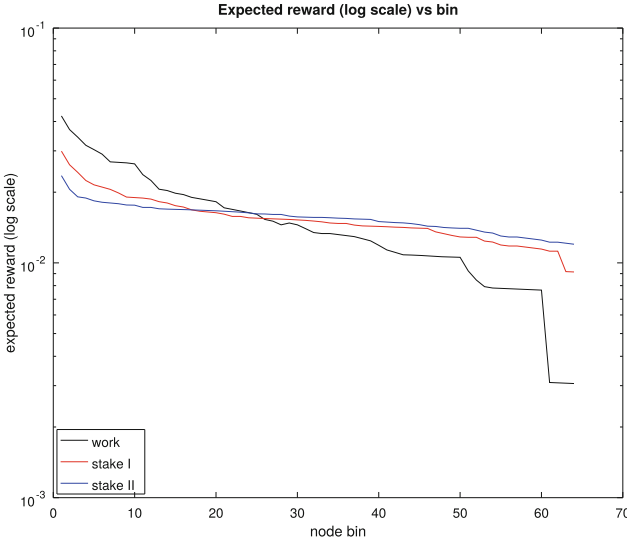


Fig. 1. Mean reward distribution for the three scenario.

From Table 3 it can be seen that PoW attains higher upper bounds, which is consistent with the less balanced reward distribution compared to the PoS variants. Therefore, a potential stakeholder may be motivated by the prospect of gaining additional rewards compared to the expected ones.

A third way to gain insight into the way the consensus protocols work is to fit a distribution to the expected network rewards. In this case more protocol properties can be derived, assuming the chosen distribution has a considerable degree of accuracy. Since from Fig. 1 the empirical distribution of the average reward appears to be a power law, three such models will be fit. Additionally, the models were selected based on the number and type of scenaria they explain.

The log-normal distribution of (16) models long scale mobile signal scatter, digital post length, and quantities made from the product of independent factors.

$$f_l(x; \sigma_l, \mu_l) \triangleq \frac{1}{x \sigma_l \sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu_l)^2}{2\sigma_l^2}\right) \tag{16}$$

The maximum likelihood (ML) estimators for μ_l and σ_l^2 are given in (17):

$$\hat{\mu}_l = \frac{1}{B_0} \sum_{k=1}^{B_0} \ln x_k \quad \text{and} \quad \hat{\sigma}_l^2 = \frac{1}{B_0} \sum_{k=1}^{B_0} (\ln x_k - \hat{\mu}_l)^2 \quad (17)$$

The Weibull distribution of (18) describes the time spent reading an Internet post, measuring therefore indirectly reader engagement as well.

$$f_w(x; k_0, \lambda_0) \triangleq \frac{k_0}{\lambda_0} \left(\frac{x}{\lambda_0}\right)^{k_0} \exp\left(-\frac{x}{\lambda_0}\right)^{k_0} \quad (18)$$

The ML estimators for $\hat{\lambda}_0$ and \hat{k}_0 are given in Eq. (19):

$$\hat{\lambda}_0 = \left(\frac{1}{B_0} \sum_{k=1}^{B_0} x_k^{\hat{k}_0}\right)^{\hat{k}_0^{-1}} \quad \text{and} \quad \frac{\sum_{k=1}^{B_0} x_k^{\hat{k}_0} \ln x_k}{\sum_{k=1}^{B_0} x_k^{\hat{k}_0}} - \frac{1}{\hat{k}_0} = \frac{1}{B_0} \sum_{k=1}^{B_0} \ln x_k \quad (19)$$

The Pareto type I distribution is frequently used to model physical and social phenomena including income distributions. It is defined for $x \geq x_0$ as in (20):

$$f_p(x; \beta_0, x_0) \triangleq \frac{\beta_0 x_0^{\beta_0}}{x^{1+\beta_0}} = \frac{\beta_0}{x_0} \left(\frac{x}{x_0}\right)^{-(1+\beta_0)} \quad (20)$$

The maximum likelihood (ML) estimators \hat{x}_0 and $\hat{\beta}_0$ are shown in (21):

$$\hat{x}_0 = \min \{x_k\} \quad \text{and} \quad \hat{\beta}_0 = \frac{B_0}{\sum_{k=1}^{B_0} \ln\left(\frac{x_k}{\hat{x}_0}\right)} \quad (21)$$

The Kullback-Leibler divergence $\langle f || g \rangle$ between two continuous distributions $f(x)$ and $g(x)$ is shown in (22) defined over the union Ω of the their domains.

$$\langle f || g \rangle \triangleq \int_{\Omega} f(x) \log_b\left(\frac{f(x)}{g(x)}\right) dx \quad (22)$$

In Table 4 the normalized Kullback-Leibler divergence between the empirical and the fitted models is shown. Rows were normalized to their respective minima.

Table 4. Divergence for reward models (Normalized).

Model/Protocol	Log-normal	Weibull	Pareto
PoW	1.6426	1.4318	1
PoS I (original)	1.8665	1.6612	1
PoS II (delegate)	2.0114	1.7344	1

The results of Table 4 can be interpreted based on the dependencies underlying the node interaction, which works differently across protocols. Under PoW the verification packets of nodes closer to the initiator are more likely to reach it first. In the PoS I case only current node rewards count, while PoS II adds computing power as a factor and hence as an extra dependency layer. Therefore, given that these simulation parameters remain constant, there is dependency in the form of memory. This is better modeled by power law distributions, while memoryless interactions by exponential ones. The log-normal distribution is the closest to an exponential distribution, the Weibull distribution balances between these two cases, and the Pareto distribution is a power law. This explains the relative scores achieved by each of these three models.

4.2 Distance from the Initiator

From Fig. 2 it can be seen that under PoW nodes closer to the initiator have considerably more chances of receiving a reward. This can be attributed to the fact that packet propagation is one of the main latency factors. In sharp contrast, the verifier selection mechanism in PoS I is topology-independent. Hence the respective curves are much different despite the round trip time being computed by the same branch of (12). The PoW II relies on an additional selection process for the witnesses which introduces additional skew compared to PoS I but remains also topology-independent. The distance from a network focal point is exploited in other applications such as high frequency trading (HFT).

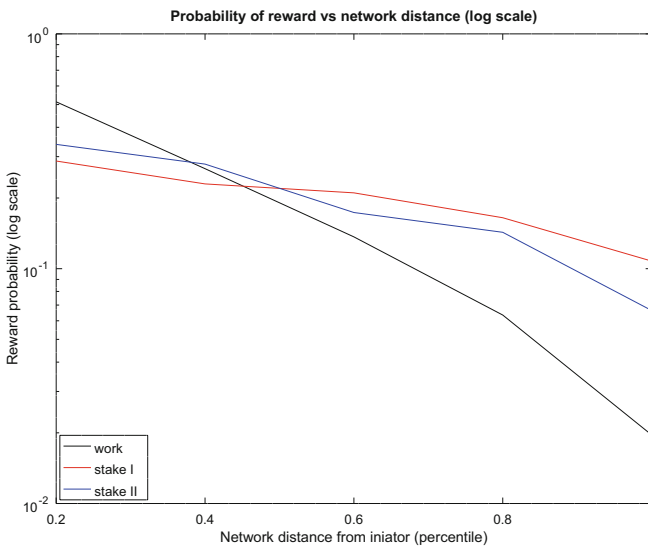


Fig. 2. Reward probability (log scale) vs network distance.

4.3 Analysis

The need for a comprehensive blockchain simulation tool is clear since to the best of the knowledge of the authors current tools focus on specific aspects. The open source SimBlock takes into consideration network parameters such as bandwidth and latency but not node reward. The very recent modular BlockSim also does not support PoS. Shadow-Bitcoin as its name suggests was created for Bitcoin simulation. Vibes as of 2017 supported only PoW. Solidity is intended only for the creation of smart contracts over the Ethereum virtual machine (EVM).

The proposed simulation has a number of limitations. A more detailed model can take into account aspects like dedicated application specific integrated circuits (ASIC) chips intended for mining rewards. Additionally, more distributions for initiator selection, rewards, and capacities can be implemented and tested.

Behavior motivation of the stakeholders and how they influence blockchain operations is vital to understanding blockchain stability. In particular, PoS can be seen as an inducement price which should be weighted against an estimate of the resources required. The reward fairness achieved by PoS may motivate stakeholders with a strong tendency for loss aversion, whereas PoW may appeal to stakeholders with powerful equipment. Furthermore, the connection of the Pareto family of distributions to the least effort principle may hint at the prospect of quick rewards as an incentive to join a blockchain.

One final note is that like any simulation the one proposed here is as accurate as the assumptions and the models allow. As real data are collected from deployed systems, their validity can be re-evaluated.

5 Conclusions and Future Work

This conference paper focuses on the probabilistic simulation of proof of work and proof of stake in blockchains. Probabilistic analysis indicates the latter achieves a more balanced reward distribution. Moreover, the probability of reward depends heavily on the distance from the initiator under the proof of work protocol.

Future research directions include more runs with a larger number of nodes and with more sophisticated consensus protocols. Moreover, failures can be extended to involve a random number of rounds, possibly relying on resiliency results from the field of temporal graphs, or neighborhoods of random radii.

Acknowledgment. This conference paper is part of Project 451, a long term research initiative whose primary objective is the development of novel, scalable, numerically stable, and interpretable tensor analytics.

References

1. Cao, B., Wang, X., Zhang, W., Song, H., Lv, Z.: A many-objective optimization model of industrial Internet of Things based on private blockchain. *IEEE Netw.* **34**(5), 78–83 (2020)

2. De Filippi, P., Mannan, M., Reijers, W.: Blockchain as a confidence machine: the problem of trust & challenges of governance. *Technol. Soc.* **62**, 101284 (2020)
3. DellaVigna, S.: Structural behavioral economics. In: *Handbook of Behavioral Economics: Applications and Foundations*, vol. 1, pp. 613–723. Elsevier (2018)
4. Dey, S.: Securing majority-attack in blockchain using machine learning and algorithmic game theory: a proof of work. In: *CEEC*, pp. 7–10. IEEE (2018)
5. Drakopoulos, G., Giannoukou, I., Mylonas, P., Sioutas, S.: The converging triangle of cultural content, cognitive science, and behavioral economics. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) *AIAI 2020. IAICT*, vol. 585, pp. 200–212. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49190-1_18
6. Drakopoulos, G., Kafeza, E., Al Katheeri, H.: Proof systems in blockchains: a survey. In: *SEEDA-CECNSM*. IEEE (2019)
7. Drakopoulos, G., Kafeza, E., Mylonas, P., Al Katheeri, H.: Building trusted startup teams from LinkedIn attributes: a higher order probabilistic analysis. In: *ICTAI*, pp. 867–874. IEEE (2020)
8. Drakopoulos, G., Voutos, Y., Mylonas, P., Sioutas, S.: Motivating item annotations in cultural portals with UI/UX based on behavioral economics. In: *IISA*. IEEE (2021). <https://doi.org/10.1109/IISA52424.2021.9555569>
9. Hasselgren, A., Krlevska, K., Gligoroski, D., Pedersen, S.A., Faxvaag, A.: Blockchain in healthcare and health sciences - a scoping review. *Int. J. Med. Informatics* **134**, 104040 (2020)
10. Khan, B.Z.: *Inventing Ideas: Patents, Prizes, and the Knowledge Economy*. Oxford University Press, New York (2020)
11. Lai, K., Oliveira, H.C., Hou, M., Yanushkevich, S.N., Shmerko, V.: Assessing risks of biases in cognitive decision support systems. In: *EUSIPCO*, pp. 840–844. IEEE (2021)
12. Li, K., Liang, H., Kou, G., Dong, Y.: Opinion dynamics model based on the cognitive dissonance: an agent-based simulation. *Inf. Fusion* **56**, 1–14 (2020)
13. Liu, Y., Ai, Z., Sun, S., Zhang, S., Liu, Z., Yu, H.: FedCoin: a peer-to-peer payment system for federated learning. In: Yang, Q., Fan, L., Yu, H. (eds.) *Federated Learning*. LNCS (LNAI), vol. 12500, pp. 125–138. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63076-8_9
14. Marountas, M., Drakopoulos, G., Mylonas, P., Sioutas, S.: Recommending database architectures for social queries: a twitter case study. In: Maglogiannis, I., Macintyre, J., Iliadis, L. (eds.) *AIAI 2021. IAICT*, vol. 627, pp. 715–728. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79150-6_56
15. Ren, W., Hu, J., Zhu, T., Ren, Y., Choo, K.K.R.: A flexible method to defend against computationally resourceful miners in blockchain proof of work. *Inf. Sci.* **507**, 161–171 (2020)
16. Saghiri, A.M., HamlAbadi, K.G., Vahdati, M.: The Internet of Things, artificial intelligence, and blockchain: implementation perspectives. In: Kim, S., Deka, G.C. (eds.) *Advanced Applications of Blockchain Technology*. SBD, vol. 60, pp. 15–54. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-8775-3_2
17. She, W., Liu, Q., Tian, Z., Chen, J.S., Wang, B., Liu, W.: Blockchain trust model for malicious node detection in wireless sensor networks. *IEEE Access* **7**, 38947–38956 (2019)
18. Voutos, Y., Drakopoulos, G., Mylonas, P.: Smart agriculture: an open field for smart contracts. In: *SEEDA-CECNSM*. IEEE (2019)
19. Wan, S., Li, M., Liu, G., Wang, C.: Recent advances in consensus protocols for blockchain: a survey. *Wireless Netw.* **26**(8), 5579–5593 (2020)

20. Werbach, K.: *The Blockchain and the New Architecture of Trust*. MIT Press, Cambridge (2018)
21. Xiao, Y., Zhang, N., Lou, W., Hou, Y.T.: A survey of distributed consensus protocols for blockchain networks. *IEEE Commun. Surv. Tutor.* **22**(2), 1432–1465 (2020)