Blockchain Networks as Adaptive Systems

Sotirios Liaskos
School of Information Technology
York University
Toronto, Canada
liaskos@yorku.ca

Bo Wang
Department of Computer Science
York University
Toronto, Canada
bowang@eecs.yorku.ca

Nahid Alimohammadi
Department of Computer Science
York University
Toronto, Canada
nalim@eecs.yorku.ca

***Abstract*—Blockchain networks have enjoyed remarkable attention the past few years in both the research community and the society at large. Such networks carry the promise of highly decentralized validation and witnessing of important social and economic events, reducing the need to rely on centralized authorities. Public proof-of-work based networks specifically, despite their limitations and challenges, continue to be popular due to their simplicity and intuitiveness. However, for such networks to perpetually meet viability, efficiency, security and environmental sustainability objectives, they need to adapt to environmental changes via continuous reconfiguration of their operating parameters. In this paper, we attempt to formulate this blockchain network adaptability problem as one of control engineering. We sketch the basic characteristics of blockchain networks as systems and identify variables available for designing a controller that allows such networks to attain macroscopic operating objectives. By means of describing and simulating a simple idealized proportional controller, we demonstrate the benefits and some of the challenges in designing such controllers.**

***Index Terms*—Blockchain, Bitcoin, Proof-of-Work, Control Theory, Complex Adaptive Systems**

I. Introduction

Blockchain networks have attracted considerable attention the past few years from both the research community and the society at large. Such networks aim at eliminating the need for centralized externally validated authorities for witnessing economic events (e.g. transactions, agreements, ownership) by instead utilizing a network of many anonymous actors to perform such witnessing tasks. By doing so, blockchain networks reduce the need to rely on single points of various kinds of failure and compromise – such as security attacks, corruption and coercion, or economic, social and environmental disasters. Through such decentralization they are believed to have the potential to revolutionize the way we arrange our social and economic affairs [1].

The decentralization objective is implemented in blockchain networks through complex mechanisms of *consensus* through which network participants agree on a unique history of events. While several protocols for attaining consensus have been proposed [2], [3], the one employed by Bitcoin [4]–[6], the pioneering blockchain network, continues to be the most studied and tested in practice. In Bitcoin, transactions submitted by users of the network are collected by participating network nodes into blocks, which then nodes compete to make part of the generally accepted history of blocks. The competition involves nodes demonstrating that they have invested a substantial amount of resources into having their version of recent transaction history (their block) accepted by the network. They are incentivized to do so via being offered – if successful – a payment in form of tokens of a currency internal to the network, which however also has external value that can potentially compensate the aforementioned participation investment.

For a blockchain network to be *sustainable*, i.e. reliably operate perpetually, several parameters affecting the above consensus process need to be within a specific range. For example, the participation resources that nodes need to invest should not be too many to disincentivize participation and slow down throughput, but not too few to bloat the network with competing versions or allow maliciously disruptive actors easily join the competition. Instead a sweet spot needs to be targeted where both performance and integrity are reliably achieved. Nevertheless, such a target is bound to be a moving one: the external price of the token, external resource availability and cost, node preferences and alternative options, as well as user fee contribution all affect the optimal configuration of the consensus mechanism. Current networks, such as Bitcoin, are rather rigid in their reconfiguration, which tends to be simplistic, infrequent and in part independent of environmental conditions, putting the networks' long-term viability at risk.

In this paper, we attempt a case for viewing and engineering blockchain networks as adaptive systems. Taking the case of the Bitcoin consensus mechanism as an example, we identify variables that affect its sustainability including those imposed by the environment and those that can be directly configured by the network's governance. Then we propose a simple model of how these variables relate and affect each other. With that model in hand we formulate the problem as a standard control engineering problem in which a controller (blockchain governance) optimizes the parameter choices so that the output of the controlled system (blockchain network) meets certain objectives. We demonstrate the benefits of such a controller through designing and simulating an idealized one that has full and accurate knowledge of the state of the environment. Using this exercise we identify the challenges for designing realistic controllers for blockchain networks focussing on peculiarities of the latter with respect to their ability to be observed and controlled.

We organize the presentation as follows. Section II offers a background on the kind of blockchain networks we consider in this research. In Section III we develop a simple quantitative model for predicting the behavior of such networks and in Section IV we sketch a basic control structure for such networks. Then in Section V we describe and simulate our simple idealized controller, discussing also its limitations. We discuss related work and conclude in Sections VII and VIII.

II. Background

The blockchain networks we consider in this research are public and based on proof-of-work consensus. Such networks consist of a number of interconnected nodes opting-in and out of the network arbitrarily. Each node receives transactions from users who desire to have these transactions witnessed by the network. When following their nominal behavior, nodes will propagate these transactions through following a flooding approach, so that all nodes maintain about the same copy of all transactions that have been submitted for validation by the network (known as the *transaction pool*).

The nodes also carry a copy of the *blockchain*, which is the set of all transactions that the network considers to be valid, organized as a list of *blocks*, each block containing a set of valid transactions. The list is linked through cryptographic hash pointers in a way that tampering with any transaction within the blockchain will cast the entire chain of blocks from that transaction and on invalid in an easily verifiable way.

Nodes collect transactions from the transaction pool, and construct blocks thereof, which they then propose to the rest of the network for addition to the agreed "official" record, i.e. the blockchain. For such a block to be successfully accepted by other nodes, it, firstly, has to contain valid transactions (e.g. payers have sufficient balance to pay, transactions properly signed etc.). However, given that the network is open and anyone can participate, to restrict participation only to agents who are invested in the good functioning of the network, an additional *proof-of-X* requirement is imposed, where 'X' is a kind of dedicated resource with real cost for the node.

The archetypal proof-of-X is *proof-of-work* which is a proof that the node has dedicated substantial amount of computing resources only to be in the position to credibly propose a block. In Bitcoin, proof-of-work is *hashcash* [7]: the node calculates the cryptographic hash (SHA256) of the block prefixed by a *nonce*, which is a string that the node can freely vary for the purpose of generating different hashes. Proof-of-work is attained if a nonce is found such that the hash meets certain rarity criteria, such as a number of leading zeros. Given the believed properties of cryptographic hash functions there is no heuristic to find such a rare hash value easily, meaning that finding it requires either extraordinary luck or substantial exhaustive computation. Nodes compete with each other in order to attain proof-of-work of their block first and propagate it as a legitimate blockchain extension proposal. The incentives' structure of the protocol is such that nodes will have no incentive not to accept other nodes' valid block, unless they are invested in the destruction of the network.

Nodes have costs and benefits for engaging in a proof-of-work based blockchain network like this. The cost is the substantial computational power they have to dedicate in order to have a reasonable chance of finding a successful hash before anyone else – computing, cooling, infrastructure support etc. The benefit is acquisition of tokens (cryptocurrencies, e.g. "bitcoins") which, due to the perceived usefulness of the network have an extrinsic value. Tokens are acquired both through fees attached to each transaction and, hence, paid by the users of the network and through granting of a reward for validating a block and having it accepted as a blockchain extension by the entire network, i.e. attaining poof-of work-first. This validation reward does not come at the expense of any participant, but is rather the method by which the network generates new tokens (hence the popular term "mining").

Public blockchain networks are open-access: any node can join or depart at any point. Whether the node will participate in the validation effort in the next instance depends on a local cost-benefit analysis: does the chance of validating a block and winning the associated reward and fees justify dedicating the cost to even try? Interestingly, though, the choice of each node to opt-in or out of the network, affects macroscopic parameters of the network, which, in turn, affect the cost-benefit analysis. To analyse these we first need a simple model for node-level decisions, which we introduce in the next section.

III. Modeling Node Behavior

As we saw, a blockchain network consists of a set of nodes engaging in proof-of-work competitions. We can view the effort to attain proof-of-work as the process of searching within a space S for elements that have a property which makes them also belong to a specific success subset $S_s \subset S$. In the case of hashcash employing SHA256, S is the space of all possible hashes 2^{256} and S_s is the set of hashes that meet the condition of certain leading zeroes – say, 2^{256-z}, where z is the binary expression of the number of leading zeros. For reasons that will become apparent below, the scope of S_s changes over time to attain specific macroscopic characteristics for the network via adjusting the *difficulty*[1] $d = |S|/|S_s|$ of the proof-of-work problem.

To attain proof-of-work, nodes in the network engage in a sequence of *trials*, i.e. random draws from S, in hopes of identifying an element that is also in S_s. Such an event would offer the node a total "mining" reward of e_r intrinsic tokens (e.g. bitcoins) each of value x in conventional currency (e.g. dollars). In addition, the node would accrue reward e_f from the user fees that are pledged in the transactions included in the block, paid by those who submitted them. However, this $e = e_r + e_f$ reward will materialize subject to successfully finding an element in S_s before any other node, which happens only with probability p_{win}. To calculate p_{win} we first observe that for the node to win the proof-of-work competition in the next trial, the trial first needs to be successful. The probability of this to be true is $p_{suc} = |S_s|/|S| = 1/d$, considering that the process is uniformly random draws from space S.

[1] Conceptually the same but arithmetically slightly different from Bitcoin's difficulty.

Nevertheless, while a node n is making a single trial, the rest of the network is doing the same, albeit much faster, seen as a massive parallel computer. Thus, for every trial n performs, $N > 1$ is the number of the trials that all the other nodes can collectively perform in the meanwhile. The expression *1:*N, then, denotes the *power ratio* between the node in question and the rest of the network. For the node to win in the next trial, it must both succeed in its one trial (probability $1/d$) and the network must fail in all the N trials it performed meanwhile, which happens with probability $(1-1/d)^N$. Thus $p_{win} = (1/d)\cdot(1-1/d)^N$ is the probability by which the node will get the reward $e \cdot x$ – excluding, for simplicity, network factors such as propagation delays.

However, performing a trial comes with the opportunity cost of not dedicating the associated unit of computational resources for participating in some other more profitable network or business purpose. We use the term *cost factor* c, measured in conventional currency (e.g., \$/trial) to represent that cost.

It follows that for the node to perform the next trial within the network in question $e \cdot x \cdot p_{win} \geq c$ must hold true, which, based on the above translates to:

$$\boxed{(e_r + e_f) \cdot x \geq c \cdot d \cdot \left(\frac{d}{d-1}\right)^N} \tag{1}$$

We will call this the *sustainability condition* for the node. Unless the condition is met, a rational node will not perform trials with the specific network. The formula is a more refined and less restrictive version of the one introduced in [8].

IV. Network Behavior and Adaptation

A. Security versus environmental sustainability

Based on where it stands with regards to the sustainability condition (1), each node will decide to either join the competitive proof-of-work games or opt-out and dedicate its resources for a different purpose. Collectively these decisions affect the overall *power* w of the network, i.e. the number of trials it performs per second. When more nodes decide to perform trials, due to e.g. an increase in the token price x, the overall network power increases; and vice-versa when mining becomes less attractive with respect to condition (1).

Importantly, sufficient amount of network power safeguards the network from adversaries. It is particularly believed that when a malicious agent controls the majority of the network power, they effectively control its validation and consensus process and, as such, are able to compromize it. This is colloquially known as the "*Goldfinger attack*" [6], [9], a reference to a villain in a popular film franchise. As such, proof-of-work networks need to exhibit enough total power that no real-world agent, malicious or not, could possibly control a substantial portion of it. At the same time, however, higher network power implies higher energy consumption and environmental footprint. It follows that, ideally, the total network power should remain at the ideal target level $\tilde{w} = w_g/\gamma$, where w_g is the maximum power an attacker is believed to be able to amass, and γ is the fraction of the network power whose control is assumed to be sufficient to successfully perform the attack. Theoretically $\gamma = 0.51$ but in practice the value may be chosen to be less given both the stochastic nature of mining and also the fact that the very threat of such an attack can act as detrimentally to the network as the attack itself [6], [10].

B. Difficulty and Block time

An additional safeguard needed in such networks is maintenance of a safe distance between validation events. In proof-of-work networks that distance is known as the *block time* t_b. When t_b is too low, many validated blocks may compete for a place in the blockchain at the same time. In such a case, depending on the structure, speed and connectedness of the network, consensus may fail, leading to blocks that are never appended to the blockchain despite being validated, or even to the introduction of two or more versions of the blockchain co-existing in the network.

To maintain a safe block time, proof-of-work networks perform adjustments to the difficulty parameter d. Note, first, that t_b relates directly to the power of the network. To show this, consider that in a process of repeated trials each with success probability $p_{suc} = |S_s|/|S|$, on average $1/p_{suc} = d$ trials will be required in order for the first success to emerge. Assume now that the network has power w trials/sec. It follows that $t_b = d/w$. Thus, given an objective to hold block time t_b constant and equal to an ideal level $\tilde{t}_b$, difficulty is periodically adjusted so that $d = \hat{w}\cdot\tilde{t}_b$ where $\hat{w}$ is an estimation of the network power based on the observed t_b's between adjustments. By adjusting the difficulty this way, the network is able to attain a stable, on average, block time.

C. Configuration and Environmental Variables

Let us now take a closer look at the variables involved in the sustainability formula (1). Two of the variables e_r and d are deterministically decided by the network's governing protocol and we can call them *configuration variables*. Taking Bitcoin as a reference network, the block reward e_r is a constant that halves every period of several years, to the effect of securing an upper bound to the total number of tokens circulating in the network in its entire lifetime. Difficulty d, on the other hand, is adjusted periodically (in Bitcoin every 2016 validation events – thus, ideally every two weeks) as we discussed above. Nodes running compliant implementations are able to adapt accordingly, thanks to the transparency and simplicity of the reconfiguration rules.

The rest of the variables of the sustainability formula (1), can be considered to be *environmental variables* and as such beyond direct control of the network's governing protocol. The quantity e_f, for example, depends on the amount that users are willing to pay in order to have their transaction processed and may depend on the cost of alternative ways to perform similar transactions, such as, e.g., sending a remittance. Further, token price x depends on public perception of its value, which may depend on a variety of factors, interestingly including, in part, the total (present or projected) number of tokens in circulation.

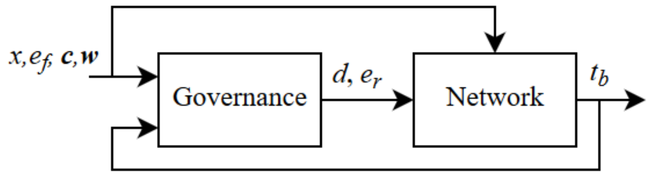

Fig. 1. Block diagram of the control problem.

Finally c and N are quantities that characterize each node. The former depends on the cost of computation (electricity etc.) augmented by the cost of foregoing use of such computational resources for other purposes. For the latter variable, each node will likely calculate an estimate $\hat{N} = (\hat{w} - w_n)/w_n$, where w_n is the perfectly known power of the node and $\hat{w}$ an estimation of the network power as above.

D. Network Configuration as a Control Problem

Given the above discussion we can focus on the problem of choosing the configuration variables d and e_r such that the network macroscopically meets two objectives:

- **O1. Constant Block Time.** The distance between the target and observed block time is minimized.
- **O2. Constant Network Power.** The distance between the target and observed total power is minimized.

It is, thus, fitting to represent the above as a control engineering problem. In Figure 1 we formulate the operation of the network and its governance as such. Specifically the network is a system with inputs the environmental parameters: token price x, cost factors $\boldsymbol{c} = c_1, c_2, \ldots, c_n$ and powers $\boldsymbol{w} = w_1, w_2, \ldots, w_n$ for all nodes n, an average fee reward per block e_f, as well as the configuration parameters: d and e_r. The latter are defined by the network governance which plays the role of the controller. The output of the system is the block time t_b. The controller uses the observed t_b as well as the environmental variables to decide the updated values for d and e_r for the next cycle, so that **O1** and **O2** are met. The problem is discrete in that adaptation decisions take place every time a new block is validated, which happens in real-time intervals of t_b.

Given the above control structure, in the next section we introduce a proportional controller that implements it, featuring, for demonstrative purposes, full knowledge of node-level decision parameters and formula. While, as we will see, the proposed controller is unrealistic, it will help us appreciate the benefit of adaptiveness and identify challenges in enabling it.

V. Demonstration

A. A Simple Controller

The example controller we describe in this section regulates d and e_r in a simple proportional fashion. At every block validation event, the controller decides whether it first needs to adapt the difficulty to a new level based on the assumed power of the network w_a. Assuming, then, that the power will stay constant for another time window, increasing the difficulty to levels $d_{target} = w_a \cdot \tilde{t}_b$ would have the effect of bringing the block time closer to the ideal $\tilde{t}_b$. As we saw, in existing networks such as Bitcoin, this is a decision that is made only every fixed amount of validation events. As opposed to Bitcoin, our controller makes the adaptation at every validation event by calculating the distance between current and proposed difficulty and accordingly increasing or decreasing the current difficulty by a proportion of that distance as in:

$$d_{next} = d_{current} + (d_{target} - d_{current}) \cdot \alpha_d$$

where $\alpha_d \in [0, 1]$ represents the size of that fraction and effectively the difficulty adaptation *rate* – a notion akin to proportional gain in control engineering terms.

Having set the difficulty, the controller goes on to decide what reward adaptation is necessary to maintain objective **O2**, i.e. maintain total network power close to a specific level. Recall that the total network power depends on whether each node finds it profitable to continue competing, effectively contributing to the overall power of the network. Let us assume a given price x of the rewarded token, the difficulty d decided above, and an estimation $\hat{e}_f$ of the total user fees that will be collected in the event of successful validation. Then, given its cost factor c_i and power ratio *1:*N_i, each node i will participate only if the reward level is at least such that the two sides to the equation (1) are equal. Let us, then, define $\mathcal{A}(e_r)$ be the set of nodes which at reward level e_r will not have a loss and will consequently decide to compete, as follows:

$$\mathcal{A}(e_r) = \{i : e_r \geq \frac{c_i \cdot d}{x} \cdot \left(\frac{d}{d-1}\right)^{N_i} - \hat{e}_f\}$$

Then, the overall network power as a function of e_r is:

$$w(e_r) = \sum_{i \in \mathcal{A}(e_r)} w_i$$

To identify the target value for e_r such that an ideal power $\tilde{w}$ is maintained, the controller needs to simply calculate the root $e_{r_{target}}$ of $\tilde{w} - w(e_r) = 0$. The resulting $e_{r_{target}}$ is used to calculate the next reward level $e_{r_{next}}$ given the current level $e_{r_{current}}$ in a way similar to that applied for difficulty:

$$e_{r_{next}} = e_{r_{current}} + (e_{r_{target}} - e_{r_{current}}) \cdot \alpha_{e_r}$$

where α_{e_r} is likewise the adaptation rate for the reward. As with difficulty, the controller will consider adaptation of the reward in every validation event.

B. Simulation

To examine the behavior of the above controller in practice we perform simple simulations based on a small hypothetical network. Specifically, the network we study includes a total of 100 nodes with randomly sampled normally distributed powers with average and standard deviation 100 and 20 (trials/sec) and cost factor averages and standard deviations 0.001 and 0.0002 ($/trial), respectively. The network starts with a difficulty $d = 6E6$, reward $e_r = 12.5$ and initial token price x of $168. The ideal block time is set to $\tilde{t}_b = 600$ sec and the ideal network power is set to $\tilde{w} = 6000$. The scenario we

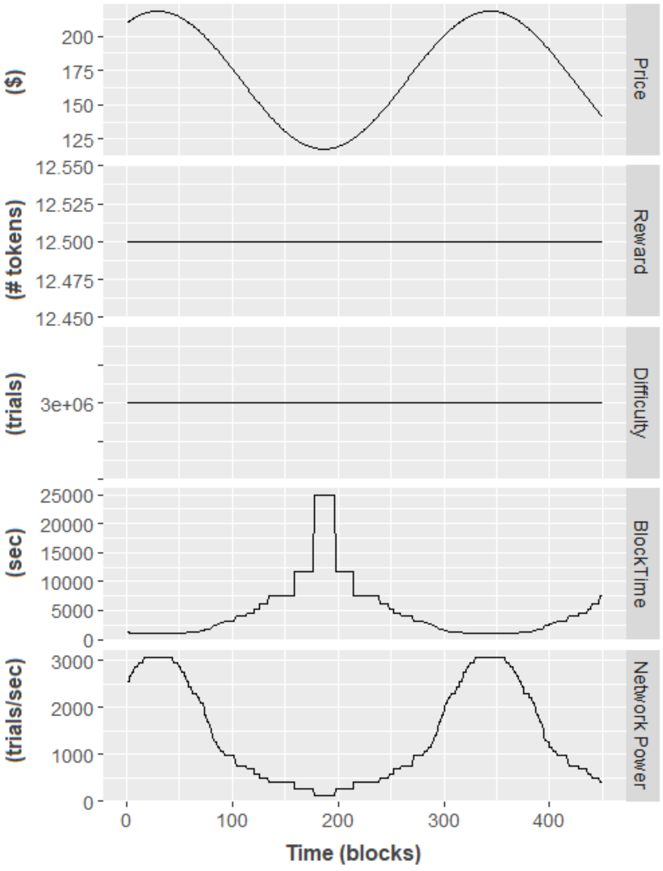

Fig. 2. Newtowk Behavior without Adaptation.

study is the network's behavior in response to a sinusoidal fluctuation of its token price x. The price drops and raises are sufficient to motivate a large number of nodes to opt-out and subsequently opt-in the competition, substantially affecting block time and power. The initial parameters and price behaviour are heuristically chosen so that the effect of the controller is better demonstrated. A total of 500 validation events (adaptation decisions) are simulated each time.

Figure 2 shows the network's behavior when either no adaptation is in effect or whatever adaptation steps take place only in cycles longer than 500 validations – a condition that fits Bitcoin. As token price drops, less and less nodes are able to cover their costs and, thus, opt-out of the competition. This results in network power loss which substantially increases block time, given that difficulty is not adapted in such a short cycle. Meanwhile, the network power falls below the target becoming susceptible to power attacks.

Use of the proposed controller offers a more promising picture, seen in Figure 3. As price drops the controller first adjusts difficulty on the assumption that power will remain stable and then adjusts reward to attain such power stability. The obvious effect is that price drops and increases synchronize with reward increases and drops respectively, though the speed of synchronization depends on the adaptation rate. Assuming the right level of the latter, we can observe that both block time and network power remain relatively stable and close to the targeted values.

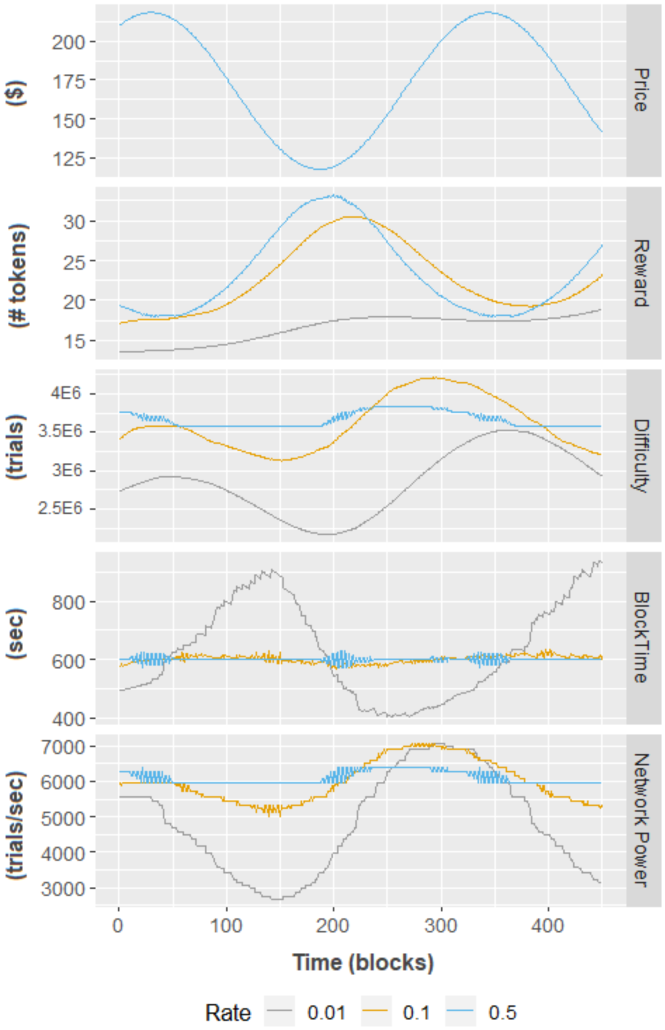

Fig. 3. Newtowk Behavior under Various Adaptation Rates.

VI. Research Challenges: Observability and Transparency

The above demonstration is meant to show how a systematic approach for designing the re-configuration behavior of blockchain networks can help them become more resilient, secure and energy efficient. Designing realistic controllers for blockchain networks can be informed by the broad literature on control engineering [11], [12] and the numerous techniques that have been introduced in that field. Here we discuss some challenges to designing pragmatic controllers that are peculiar to blockchain networks, using the controller we described earlier as a starting point.

The first challenge is observability of node characteristics. While our controller has full knowledge of the powers and cost factors of all potential participants, in practice these cannot be known directly, at least in public open-access networks. At best, a realistic controller needs to work with assumptions about the distributions of these parameters. An interesting research question is, thus, whether and how such assumptions can be made and based on what data.

Node-level decisions are also oversimplified in our controller. Firstly, nodes cannot be assumed to be rational, i.e., perfectly tuned to immediately opt-out of the competition when it is not profitable. For instance, nodes may run at a loss to contribute to the survivability of the network/token they are invested in. Further, switching to a different network may come at a cost or may not even be available in which case opting out means that the node literally switches off its computing equipment. Moreover, even nodes with the intent to be rational and instantly capable of opting-in and out may not have access to accurate information for making their decision such as their power share in the network (e.g., N above).

It follows from the above that an attempt to solve the control problem may be difficult without incorporating, e.g., stochastic and adaptive-control elements. However, a requirement that is peculiar to blockchain networks is the need for *transparency* of the adaptation calculations. Specifically, nodes need to be able to agree at any point in time on what the exact configuration parameters of the network are. The problem cannot be solved by, e.g., a central omni-observable governance node that owns the authoritative parameters, as this contravenes the decentralization premise of such networks. Thus, adaptation decisions need to be unambiguously reproducible by any participating node. In Bitcoin, for example, difficulty is adjusted at a fixed frequency and following a very simple linear formula that nodes can comply with independent of their implementation. Compared to this, approaches such as the application of, e.g., advanced machine learning or system identification techniques can, in practice at least, be susceptible to deviations due to parameterisation and implementation biases and miscommunications. The problem intensifies when more complex formulations of the objectives and their priority is necessary, i.e., when the problem is viewed as a nontrivial multi-objective optimization one. With these issues in mind, a potentially useful research direction is the study of parameterization as part of the consensus process, in which nodes reach an agreement on effective parameters in the same way they do for transactions (*"on-chain"*). Techniques for decentralized learning and decision making may also prove relevant [13], as long as the competitive nature of consensus is kept in view.

VII. Related Work

While the current inflexibility of many dominant public blockchains, such as Bitcoin, receive frequent criticism [14], [15], there is no consensus as to how proof-of-work blockchain networks can configure themselves to be more resilient in the face of external disturbances. In this context, several attempts to view blockchain networks as adaptive systems have recently emerged. Most of the work has focussed on improving difficulty adaptation [16]–[19] without regarding reward as a possible variable. One exception is work by Saito and Iwamura [20] where reward adaptation is considered for the purpose of price stability: the network inflates or deflates in response to demand increases and decreases, respectively. However the effect on power and, thus, security is not within the focus of that work. In an attempt to offer a theoretical framework, Thomas and Mantri [21] offer a characterization of blockchain networks as complex adaptive systems that operate on stigmergy, which describes agent activity determined by local signs left by the same or other agents. The work is a possible starting point for understanding the acquisition, communication and enforcement of configuration decisions within blockchain networks. Elsewhere, Zargham et al. [22] propose a comprehensive linear time-expanding representation of blockchain networks inspired directly from the domain of control engineering. Although the model is not specifically focused on the configuration variables we discussed here, it can potentially be used for the purpose. Lin et al. [23] on the other hand, focus on a reward adaptation mechanism that relies on a transaction fee rate determination, in turn determined by past transaction volume. The objective of that work, which is the sustainability of proof-of-work in the absence of mining reward – as will eventually be the case in Bitcoin – is somewhat different than that of ours, which is power stability combined with stable block time.

Further, efforts to apply blockchain to adaptive systems in IoT settings are also worth mentioning. In one case [24] the consensus mechanisms of blockchain work to isolate suspicious elements, while in another [25] control-based techniques are applied to optimize queue length in IoT networks to ensure that the queues are not saturated slowing down the response of the IoT network; the idea is implemented with the support of a blockchain platform.

Finally, it is worth adding that proof-of-work is only one of the possible ways consensus can be safeguarded in blockchain networks [2], [26] with *proof-of-stake* increasing in popularity often combined with delegation/voting schemes. It is important to note that independent of the choice of mechanism, the problem of control of its parameters will likely continue to exist in varying forms.

VIII. Conclusions

We proposed a simple model for quantitatively reasoning about the parameters that affect and are affected by the consensus process of public open-access blockchain networks and attempted a first-cut formulation of such model as an adaptive system. To motivate the approach we experimented with an idealized controller, only to subsequently deconstruct it in order to demonstrate the challenges in designing real-world controllers. Such challenges include limited observability, complex and heterogeneous local decision making methods and adaptation consensus issues. We are hoping that our work motivates further research for developing mechanisms and protocols for making blockchain networks more adaptive and, as such, more resilient and sustainable.

References

[1] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. Portfolio, 2016.

[2] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *Proceedings of the 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, jan 2017, pp. 1–5.

[3] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," www.bitcoin.org, Tech. Rep., 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[5] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Advances in Cryptology - EUROCRYPT 2015. Lecture Notes in Computer Science, vol 9057*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer, 2015, pp. 281–310.

[6] J. A. Kroll, I. C. Davey, and E. W. Felten, "The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries," in *Proceedigns of the 12th Workshop on the Economics of Information Security (WEIS 2013)*, Georgetown University, Washington, D.C., 2013, pp. 1–21.

[7] A. Back, "Hashcash - A Denial of Service Counter-Measure," Technical Report, Tech. Rep., 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf

[8] S. Liaskos and B. Wang, "Towards a Model for Comprehending and Reasoning About PoW-based Blockchain Network Sustainability," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18)*. Pau, France: ACM, 2018, pp. 383–387. [Online]. Available: http://doi.acm.org/10.1145/3167132.3167175

[9] Joseph Bonneau, "Hostile blockchain takeovers," in *Proceedings of the 5th Workshop on Bitcoin and Blockchain Research*, Santa Barbara Beach Resort, Curacao, 2018.

[10] N. Houy, "It Will Cost You Nothing to 'Kill' a Proof-of-Stake Crypto-Currency," *SSRN Electronic Journal*, vol. 34, 2014.

[11] T. A. Weber, *Optimal Control Theory with Applications in Economics*. MIT Press, 2011. [Online]. Available: http://www.jstor.org/stable/j.ctt5hhgc4

[12] K. Ogata, *Modern Control Engineering*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[13] E. Pournaras, P. Pilgerstorfer, and T. Asikis, "Decentralized Collective Learning for Self-managed Sharing Economies," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 13, no. 2, pp. 10:1—-10:33, nov 2018. [Online]. Available: http://doi.acm.org/10.1145/3277668

[14] M. Iwamura, Y. Kitamura, T. Matsumoto, and K. Saito, "Can We Stabilize the Price of a Cryptocurrency?: Understanding the Design of Bitcoin and Its Potential to Compete with Central Bank Money," Institute of Economic Research, Hitotsubashi University, Discussion Paper Series 617, 2014. [Online]. Available: https://econpapers.repec.org/RePEc:hit:hituec:617

[15] N. T. Courtois, M. Grajek, and R. Naik, "The Unreasonable Fundamental Incertitudes Behind Bitcoin Mining," *arXiv e-prints*, p. arXiv:1310.7935, oct 2013. [Online]. Available: http://arxiv.org/abs/1310.7935

[16] D. Fullmer and A. S. Morse, "Analysis of Difficulty Control in Bitcoin and Proof-of-Work Blockchains," *arXiv preprint arXiv:1812.10792*, 2018. [Online]. Available: https://arxiv.org/abs/1812.10792

[17] G. Hovland and J. Kucera, "Nonlinear Feedback Control and Stability Analysis of a Proof-of-Work Blockchain," *Modeling, Identification and Control*, vol. 38, no. 4, pp. 157–168, 2017.

[18] D. Kraft, "Difficulty control for blockchain-based consensus systems," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 397–413, mar 2016. [Online]. Available: https://doi.org/10.1007/s12083-015-0347-x

[19] D. Meshkov, A. Chepurnoy, and M. Jansen, "Short Paper: Revisiting Difficulty Control for Blockchain Systems," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology. DPM 2017, CBT 2017. Lecture Notes in Computer Science, vol 10436*, J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomartí, Eds. Cham: Springer, 2017, pp. 429–436.

[20] K. Saito and M. Iwamura, "How to Make a Digital Currency on a Blockchain Stable," *arXiv e-prints*, p. arXiv:1801.06771, jan 2018. [Online]. Available: https://arxiv.org/abs/1801.06771

[21] J. Thomas and P. Mantri, "Complex Adaptive Blockchain Governance," *MATEC Web Conference*, vol. 223, p. 1010, 2018. [Online]. Available: https://doi.org/10.1051/matecconf/201822301010

[22] M. Zargham, Z. Zhang, and V. Preciado, "A State-Space Modeling Framework for Engineering Blockchain-Enabled Economic Systems," *arXiv e-prints*, p. arXiv:1807.00955, jul 2018. [Online]. Available: http://arxiv.org/abs/1807.00955

[23] F. Lin, Z. Zheng, Z. Huang, C. Tang, H. Peng, and Z. Chen, "A Sustainable Reward Mechanism for Block Mining in PoW-Based Blockchain," in *Proceedings of the 5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, Hangzhou, Zhejiang, China, aug 2018, pp. 156–161.

[24] P. E. Sedgewick and R. de Lemos, "Self-Adaptation Made Easy with Blockchains," in *Proceedings of the 13th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Gothenburg, Sweden, may 2018, pp. 192–193.

[25] R. Casado-Vara, P. Chamoso, F. De la Prieta, J. Prieto, and J. M. Corchado, "Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management," *Information Fusion*, vol. 49, pp. 227–239, jan 2019. [Online]. Available: https://doi.org/10.1016/j.inffus.2018.12.007

[26] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, may 2018, pp. 1545–1550.