Economics of Proof-of-Stake Payment Systems [*]

Giulia Fanti[†], Leonid Kogan[‡], Pramod Viswanath[§]

First draft: November 2018

Latest draft: May 2021

Abstract

In this paper we develop a valuation framework for a Proof-of-Stake (PoS) payment system. Active network participants (proposers and validators) are required to stake tokens, and receive payments in return for their services. This property of the PoS system connects cash flows to token holdings, and allows for valuation using conventional methods. As an application of our framework, we analyze security properties of the PoS system. We show that while high token valuation relative to the flow of transactions is central to network security, valuation bubbles have adverse security implications. State-contingent monetary policies can be used to alleviate this problem. We also show how our framework can be extended to the systems with a payment channel network as the second layer.

1 Introduction

Recent years have sparked significant interest in blockchain-powered payment systems, or cryptocurrencies. In theory, cryptocurrencies offer a number of advantages over traditional payment systems, including reduced fees, improved security, and a more robust trust model. However, despite their great promise, cryptocurrencies have gained limited traction as an alternative to traditional payment systems. There are many reasons for this, including one fundamental, technical drawback. Namely, the consensus mechanism used in Bitcoin and related cryptocurrencies is poorly-suited to high-rate applications like payments processing.

[*] We are grateful to Bruno Biais, Fahad Saleh, and the seminar participants at the MIT Sloan School of Management, the 2019 NBER Conference on Blockchain, Distributed Ledgers, and Financial Contracting, the 12th Annual Paul Woolley Centre Conference at the LSE, and the 2020 Conference on The Economics of Distributed Ledger Technology at the University of Cambridge for their comments.

[†] Department of ECE, CMU, gfanti@andrew.cmu.edu

[‡] Sloan School of Management, MIT, and NBER, lkogan@mit.edu

[§] Department of ECE, UIUC, pramodv@illinois.edu

To understand this claim, notice that a blockchain is simply an immutable,[1] sequential data structure that is maintained by a distributed set of nodes. At a high level, the core consensus problem in cryptocurrencies is the following: given a set of transactions, how can a group of mutually untrusting parties agree on the ordering of transactions to form a global, immutable ledger? In practice, transactions are grouped together into so-called *blocks*, thus giving rise to the name. *Nakamoto consensus*, first proposed in the original Bitcoin paper Nakamoto (2008), solves this problem using a technique called proof-of-work (PoW). In PoW, so-called *miners* race to solve a computationally-intensive puzzle. The first miner to solve the puzzle (and provide proof of that solution) is allowed to choose the next block in the blockchain (i.e., choose the next set of transactions in the sequential ledger). The act of producing a new block and appending it to the chain is called *block proposal*, and comes with associated rewards for the miner—typically a combination of transaction fees and a freshly-minted tokens. A key property of PoW computational puzzles is their difficulty, which must be high to protect the security of the blockchain.[2] Hence, PoW exhibits an unfavorable tradeoff between security and computational complexity (see Bagaria, Kannan, Tse, Fanti, and Viswanath, 2018); indeed, Bitcoin alone is estimated to consume more energy than some developed nations (Lee, 2017). Moreover, common PoW designs like Nakamoto consensus are fundamentally limited in terms of the throughput and latency figures they can support (Bagaria et al., 2018). Although more performant PoW consensus mechanisms exist, their adoption is less widespread (Eyal, Gencer, Sirer, and Van Renesse, 2016; Pass and Shi, 2018a; Yu, Nikolic, Hou, and Saxena, 2018; Yang, Bagaria, Wang, Alizadeh, Tse, Fanti, and Viswanath, 2019; Fitzi, Gazi, Kiayias, and Russell, 2018). Because of these drawbacks, deployed PoW cryptocurrencies have struggled to support the volume of traffic flowing in industrial-grade payment systems.

The drawbacks of PoW are well-documented. To address them, an alternative called *proof-of-stake* (PoS) has been proposed. In PoS systems, the next block proposer(s) are selected randomly from the set of all users, with probability proportional to the fraction of total stake (tokens) held by each user. In principle, this selection process can be significantly

[1] Immutability of the blockchain is an idealization. In practice, it is more accurate to think of this data structure as temper-resistant.

[2] For example, in Bitcoin, the entire network solves one puzzle every 10 minutes on average.

more efficient in terms of computational effort than PoW mining. However, PoS is just an access control mechanism to prevent malicious users from generating arbitrarily many fake accounts and taking over the system. On its own, PoS does not specify a consensus protocol. For example, there are many ways to implement PoS proposer election and block management (Kiayias, Russell, David, and Oliynykov, 2017; David, Gaži, Kiayias, and Russell, 2018; Kerber, Kiayias, Kohlweiss, and Zikas, 2019; Chen and Micali, 2016). Indeed, the relative merits of PoW versus PoS is a major topic of discussion in the blockchain community, particularly regarding the economic and security tradeoffs of each approach, e.g., (Brown-Cohen, Narayanan, Psomas, and Weinberg, 2019; Wall, 2019).

In this paper, we analyze the economic properties of PoS payment systems. While analogous studies have been conducted on PoW cryptocurrencies, we find that PoS systems in particular exhibit traits that lend themselves naturally to analysis using traditional valuation tools. In particular, because individual token holdings in the PoS stake systems determine the allocation of rewards within the system, as we discuss in our PoS primer below, tokens effectively confer cash flow rights on their holders – they can therefore be valued using standard valuation methods.

We propose a dynamic equilibrium model for quantifying the *value* of a PoS payment system as a function of system parameters like transaction volume, token velocity, and token supply schedule. In PoW cryptocurrencies, token valuation is important because it gives investors and regulators a way to reason about the worth of the system. While this is also true of PoS cryptocurrencies, token valuation plays an additional role in PoS: it gives a quantitative measure of the *security* of the system. Because of the way PoS systems are architected, higher token values imply better security. Our model therefore depends critically on PoS-specific implementation choices, and allows us to approximate the effects of changing various parameters in the PoS consensus protocol.

We analyze the effect of valuation bubbles on network security within our valuation framework. This analysis highlights some important connections between network design features and properties of the network as a financial system, suggesting directions for further investigation.

The ideas proposed in this paper differ from comparable analyses for PoW systems in a few key respects. First, the primary cost in PoW systems is the physical cost of running a mining rig: electricity, hardware, and cooling, for instance. Rewards in PoW systems are designed to compensate for these costs. In contrast, the physical costs of PoS systems are comparatively low. The primary expense for PoS participants is the fact that they must deposit tokens as collateral in order to reap rewards; we discuss this point in detail in Section 2. This observation allows us to tie PoS cryptocurrency valuations to fiat quantities through the opportunity cost of holding tokens. Such a connection has not been possible in prior PoW valuations.

Another key difference is the fact that in PoS systems, it is (relatively) easy for users reallocate their resources. A user can choose to use her tokens to participate in the consensus mechanism one day, and then use those same tokens to purchase goods the next. Because of such low frictions, PoS payment systems are amenable to traditional arbitrage-based valuation techniques. At the same time, the same ease of reallocation may have undesirable implications for network security, as our analysis of valuation bubbles indicates. In contrast, under the PoW setting, a miner that invests its resources in mining rigs cannot easily convert those mining rigs into spendable Bitcoins, and vice versa. Although it is possible to rent computational resources specifically designed for PoW mining, the dominant miners in major cryptocurrencies tend to own their own mining infrastructure (Miller, 2014; Kosik, 2018).

Because of the differences between the PoS and PoW systems, the valuation framework presented here is quite different from prior work on valuing PoW cryptocurrencies. We expect our framework to provide a useful starting point for developing more sophisticated models based on specific consensus mechanisms or protocols.

Our paper is most closely related to the small but growing literature on cryptocurrency valuation. Existing papers are primarily concerned with valuation in the Proof-of-Work setting, specifically in the context of Bitcoin's consensus protocol.

Cong, Li, and Wang (2018) develop a dynamic model of cryptocurrency adoption and valuation. Their paper postulates that cryptocurrency yields a flow of benefits to the users, which increases with the degree of currency adoption. This setting is more suitable for describing the dynamics of token adoption, with token valuation described in a relatively reduced form. Related to this, are the studies by Athey, Parashkevov, Sarukkai, and Xia

(2016) and Pagnotta and Buraschi (2018). Athey et al. (2016) emphasizes the effect of learning about eventual success of the cryptocurrency on its price volatility. Pagnotta and Buraschi (2018) focuses on the link between network security and valuation, and the negative feedback from loss of security on token valuation, which may in turn lead to multiple equilibria. Sockin and Xiong (2018) also consider the possibility of multiple equilibria in a static model of utility tokens, which allow their holders to participate and benefit from membership in a digital platform. This paper is one of a few that consider PoS settings in addition to the more common analysis under the PoW setting.

In our analysis, we do not explicitly incorporate feedback from network security to token values, and focus on the reverse direction: implications of valuation dynamics for network security. What distinguishes our paper from the prior work is that we establish an explicit connection between transactions on the network, associated transaction fees, and token values – our valuation model is conventional and based on absence of arbitrage. Our results do not hinge on assumed "convenience yield" (benefits) of holding tokens, or on assumptions about token velocity.

Liu and Tsyvinski (2018) is an empirical study of the risk-return properties of cryptocurrencies. Biais, Bisiere, Bouvard, Casamatta, and Menkveld (2018) is another related study, estimating risk-return restrictions on historical Bitcoin prices. Our analysis highlights that the opportunity cost of holding tokens is central to token valuation.

We begin this paper by describing prior work analyzing the economics of cryptocurrencies, and highlight why this analysis does not apply to PoS systems. In Section 2, we explain how PoS cryptocurrencies work, in order to motivate our modeling of the system. In Section 3 we introduce our equilibrium valuation model, followed by a stochastic version in Section 3.2. Finally, we discuss security implications in Section 4.1.

2 Proof-of-Stake Primer

This section gives a brief primer on common design choices in proof-of-stake (PoS) blockchains. These design choices will inform our models in Section 3. Recall that the central question in blockchains is how to determine (and agree upon) the next block in the blockchain. As such, many existing PoS consensus mechanisms have three key architectural components:

1. **Proposer election**: One or more nodes are selected to coordinate the protocol for block production.

2. **Block proposal**: The proposers (nodes elected in the previous phase) form new blocks, add them to the blockchain data structure, and broadcast them to the rest of the network.

3. **Block confirmation**: The goal of block confirmation is to convince honest nodes of the validity of the entire blockchain sequence up to a given block. The mechanism for doing so can vary across different blockchain designs. For instance, in Algorand, confirmation occurs after every block, and the guarantee is ultimately probabilistic; the block can only be overturned with a fixed, low probability, assuming that at least 2/3 of participants are honest (Chen and Micali, 2016). In other PoS cryptocurrencies (e.g., Ethereum v2.0), confirmation is executed via a separate finalization procedure that is run by a (possibly separate) set of nodes called *validators*. Typically, once a block is finalized, it and all prior blocks cannot be reverted without the nodes responsible for the change incurring significant penalties through a process called *slashing*.

It is useful to understand common design choices for each of these architectural components in order to create meaningful economic models. Before explaining our model in greater depth, note that most cryptocurrencies – including PoS ones – make assumptions that are critical for their security guarantees. The most common assumption is that a majority or supermajority of nodes are honest, or obey protocol exactly. However, in practice, nodes are more likely to be rational than honest; this could lead to nodes disobeying protocol in ways that are not accounted for by theoretical models.

For example, in most blockchain protocols, nodes are assumed to store the full blockchain history. Without a full history (or some representation thereof), it is impossible to verify the correctness of a sequence of blocks. However, blockchain storage costs grow linearly with time, and quickly grow prohibitive. Nodes are therefore faced with a tradeoff between storing the full blockchain, which is more secure, or storing only a subset of the blockchain, which is more practical. In Bitcoin, a wide majority of nodes have chosen the latter option, operating as *light nodes* that store minimal data about the full blockchain (Donet, Pérez-Sola, and

Herrera-Joancomartí, 2014; Palai, Vora, and Shah, 2018).

We abstract from these issues in our analysis by assuming that the PoS blockchain consensus mechanisms function as designed. We acknowledge that there are issues related to the incentive-compatibility of these systems, and these are active areas of current research. However, we leave these issues aside (assuming that they will eventually be resolved), for the sake of exploring the central economic properties of PoS systems.

2.1 Proposer election

Most blockchain protocols work by specifying a designated node or set of nodes that are responsible for block generation. The process of choosing those nodes is sometimes called proposer election, and is generally executed via a distributed protocol. For example, in Bitcoin, that distributed protocol is a randomized proof-of-work protocol.

In PoS systems, a common approach to proposer election relies on verifiable random functions (VRFs). This approach is used in most PoS cryptocurrencies, including Qtum, Peercoin, BlackCoin, Particl, Cardano (Kiayias et al., 2017), and Algorand (Chen and Micali, 2016).[3] The idea is that each node will locally execute a pre-defined function that takes only public inputs, such as the blockchain contents, the node's own public key, and the current time. If this function evaluates to 1, the node is a proposer; otherwise, it is not. For example, consider the PoSv3 protocol, which is used in Particl, Peercoin, BlackCoin, and Qtum. Every time interval (e.g., 16 seconds), each node computes the following:[4]

$$\text{Hash}\ (\text{BlockchainContents} \mid\mid \text{Coin} \mid\mid \text{CurrentTime}) \overset{?}{\leq} \text{DifficultyParameter} \tag{1}$$

where Hash$(\cdot)$ denotes a hash function, 'BlockchainContents' is a condensed representation of the blockchain contents, 'Coin' denotes a user's token ID, 'CurrentTime' the current quantized timestamp, and 'DifficultyParameter' a system-wide parameter that ensures the number of proposers is not too high. We are omitting many details from this expression, but notice that the user can compute this function for each coin it possesses, with an equal

[3] See Qtum (https://qtum.org/), Peercoin (https://peercoin.net/), BlackCoin (https://blackcoin.org/), Particl (https://particl.io/), and Cardano (https://www.cardano.org/en/home/) for technical background.

[4] This expression has been simplified for clarity.

probability of succeeding each time. Hence its probability of being elected proposer on any of these coins is proportional to the number of coins the user owns.

This VRF-based approach has the advantage of being fast, distributed, and random. Note that each node can check its own status as a proposer without consulting any other nodes. In practice, more than one node may be elected proposer due to randomness in the protocol, even if the expected number of proposers per time slot is less than one. A common, critical property of many VRF proposer election mechanisms is that they require input coins to be "old enough". For example, (1) could be modified such that only coins that were last spent at least m blocks ago can be used as valid inputs. From the perspective of our economic models, this requires users to hold tokens for m blocks before they can participate in proposal. In this sense, users must decide if they want to hold their tokens (to be eligible to be a proposer) or spend them. This decision will be made explicit in our model in Section 3.

2.2 Block proposal

The purpose of block proposal is to provide a suggested ordering of transactions that can be accepted or rejected by the rest of the network. For efficiency reasons, this is typically done by the elected proposers (see Section 2.1). There are many variants of block proposal, but most commonly, for a given block, each proposer at a given time slot assembles the most recent transactions into a block, and then shares its block with the rest of the network for approval. Once a proposer has been elected, the main uncertainty in the system is regarding where to append the block. In some blockchains like Algorand, there is no forking; in other words, the likelihood of the system producing two or more concurrently-proposed blocks (for instance) is chosen to be arbitrarily low (Chen and Micali, 2016; Gilad, Hemo, Micali, Vlachos, and Zeldovich, 2017). Hence the block proposer always perceives the blockchain as a single chain, and appends its block to the end of that global blockchain. In chain-based protocols, such as those using PoSv3 (e.g, Qtum, Particl), forking is much more common; this happens both because of network delays (a proposer may not know about the most recent block) and the fact that multiple proposers may be elected at the same time. In chain-based protocols, proposers typically append their block to the longest chain known to the proposer at the time.

For our purposes, the details of block proposal matter primarily because of how they affect the incentives in the system. For instance, in chain-based protocols, proposers are rewarded for creating blocks that end up in the final longest chain. These rewards manifest in two ways: block rewards and transaction fees. **Block rewards** refer to a predetermined reward of newly-minted tokens given to the proposer of each block in the longest chain. In most cryptocurrencies, this block reward starts out large and decays (often to zero) over time. Block rewards are the primary tool for controlling monetary policy in cryptocurrencies, and their schedule is often determined upfront by system designers and encoded in client software. If a group of nodes were to disagree with the established monetary policy, they could easily modify the client software to follow a different block reward schedule and run the modified client. These blocks would not be accepted by nodes running the old client software, whereas the new client software would not accept blocks mined by nodes using the old software; effectively, the network would become partitioned (experience a fork).

The second class of rewards is **transaction fees**. Transaction fees are (typically) paid by the sender of a transaction in addition to the desired transfer amount. The transaction fee is generally paid to the proposer of the block in the main (longest) chain that includes the transaction. However, these fees can also be split between the proposer of a block and a set of *validators* that finalize transactions, as described in Section 2.3. Regardless, the higher the transaction fee of a particular transaction, the higher the reward reaped by the proposer of the block including that transaction (and possibly block validators who finalize it). In practice, many cryptocurrencies allow users to choose their own transaction fees, above a certain minimum. Another option is to set transaction fees algorithmically, e.g., by correlating them with the amount of the underlying transaction. As with block rewards, constraints like minimum transaction fees are upheld only because nodes implicitly agree to them. Nothing prevents network participants from spontaneously deciding to change the minimum transaction fee in their local software. This consensus problem of constraining transaction fees is not typically secured by any explicit incentive mechanism; the implicit incentive is that if transaction fees are set too low, the network may be overwhelmed with spam transactions. In this work, we will make assumptions only on the *aggregate* behavior

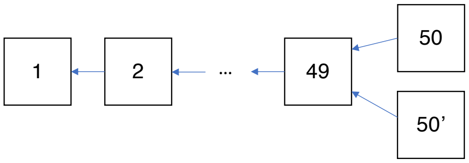

Figure 1: Example blockchain evolution. In a finalization protocol that finalizes every 50 blocks, the validators must choose between block 50 and 50'.

of transaction fees as a function of the total value of transactions processed in the system. These assumptions are discussed in Section 3.

2.3 Confirmation

Just as transactions must be confirmed in traditional financial systems (e.g., bank transactions), block confirmation rules are essential to cryptocurrencies. In traditional chain-based PoW cryptocurrencies like Bitcoin, a block (and all the transactions contained within) is typically confirmed once the block is embedded k blocks deep in the longest chain, where k is chosen to ensure a maximum tolerable probability of transaction reversal ϵ in the presence of a malicious adversary (Nakamoto, 2008). Intuitively, this rule works because as long as the adversary has less computational power than all the honest nodes put together, its *rate* of block discovery is slower than the honest nodes'. Hence, the adversary cannot overtake the honest chain with more than negligible probability in k.

In chain-based PoS systems, above intuition no longer holds. Since the computational cost of finding new blocks is negligible in PoS systems, the rate of block discovery is effectively unlimited. Hence, rational nodes are incentivized to try to propose blocks not only on the longest chain, but also on every other block in the tree, in hopes of collecting block rewards and/or transaction fees. For this reason, designing a secure confirmation rule (i.e., a rule that ensures a block is not reverted with more than some threshold probability ϵ) is known to be significantly more subtle in PoS systems (Fan and Zhou, 2017; Wang, Kamath, Bagaria, Kannan, Oh, Tse, and Viswanath, 2019).

Because of these subtleties, some cryptocurrencies are considering adding a procedure called *finalization* to the usual confirmation rules. Whereas transactions in PoW systems are typically confirmed with probabilistic guarantees, finalization is often secured by gametheoretic guarantees; participants are generally awarded tokens to incentivize participation. Hence, finalization is not inherently stronger or weaker than probabilistic confirmation; it is simply a different kind of security guarantee.

Although finalization is not yet widespread, it is being considered for adoption in a few major cryptocurrencies. Ethereum, for instance, has proposed a finalization mechanism in its roadmap (Buterin and Griffith, 2017), and major cryptocurrencies like Ripple and Stellar provide similar finality guarantees (Mazieres, 2015). Since Ethereum's finalization procedure is concrete, we describe it here. Finalization relies on a set of nodes that deposit stake in order to be recognized as *validators*. As long as their stake is locked up, validators can participate in the finalization procedure. The finalization procedure consists of periodically (e.g., every 50 blocks) running a voting protocol among the validators. The purpose is to choose one chain from the set of candidate chains that have appeared in the system. For instance, suppose the system forms a perfect blockchain until block 49, at which point two proposers independently append blocks to the same parent block. Then we have a blocktree that looks like Figure 1. The job of the validators is to choose one of the forks, and seal that choice for posterity. If the validators choose block 50', for example, then subsequent blocks will build on 50', not 50. The mechanics of the voting procedure are not important for this work; the key point is that validators receive a portion of transaction fees as a reward for participating in the finalization procedure. Hence, they are being rewarded for locking up (depositing) their stake. The longer they remain validators, the longer they reap these rewards.

Slashing Finalization protocols typically require validators to deposit stake for security reasons; the stake is used as collateral in case a validator misbehaves. At a high level, 'misbehavior' is behavior that can result in one of two outcomes: (1) a finalized block getting un-finalized, or (2) conflicting blocks getting finalized. If a validator misbehaves by, say, trying to finalize two blocks at the same height, the validator's deposit will be *slashed*, or confiscated. The actual mechanics of slashing are handled by smart contracts that are pre-programmed to confiscate funds if any node can produce evidence of a validator misbehaving. In Ethereum's finalization protocol, Casper the Friendly Finality Gadget (FFG), 4% of the slashed funds are given to the node that reports the misbehavior, and the remaining 96% is burned so that nobody can spend it. Casper FFG imposes a lower limit on the deposit to become a validator; at the time of writing, this minimum was 1500 ETH, close to $190,000. Hence, any misbehavior would result in the validator losing a substantial sum of money. Our security analysis in Section 4.1 extends the baseline model to include the effects of slashing. Notice that although slashing is described in the context of finalization, the concept can also be used to protect against other kinds of misbehavior. For example, one could require proposers to deposit stake, and slash proposers who propose multiple blocks at once.

A natural question is how to enforce slashing in blockchains, since they are inherently distributed. A key challenge is that nodes must first come to consensus on the fact that a particular validator misbehaved. Casper FFG manages enforcement by having all actions recorded in transactions: a node signals its intent to join the validator pool by sending a special deposit transaction to the network, and all of its votes for blocks are also recorded in transactions (votes can be viewed as endorsements). All transactions are signed with the validator's private key, so if an observer notices two transactions from a validator voting for conflicting blocks, the observer can prove to the rest of the network that the validator was misbehaving. The observer can then submit its own transaction with a link to the two offending transactions to claim its 4% finder's fee. Since all valid transactions are included in the blockchain, they eventually get confirmed through a combination of the block proposal mechanism and/or the finalization procedure. This highlights a very important point: because we require transaction confirmation to enforce good behavior in finalization, the blockchain formation mechanism must itself have strong security properties. In other words, it is not enough to rely on finalization alone for transaction confirmation: block proposal and finalization each improve and strengthen the security guarantees of the other. A principled economic analysis of the interplay of incentives between these systems has not been studied (to the best of our knowledge), and is an interesting direction for future work.

3 Valuation Model

In this section, we present an economic model of PoS cryptocurrencies. The main economic implication of the PoS designs discussed in Section 2 is that network participants who serve as block proposers or validators can earn rewards by committing their tokens to such activities. For example, proposers must keep their tokens unspent for some period of time (Section 2.2), and validators must deposit their stake to participate in finalization (Section 2.3). In our model, for the sake of simplicity, we do not distinguish between the two functionalities; we refer to both simply as 'validation'. Thus, as long as tokens are used for validation, each token generates a stream of cash flows for its holder in the form of additional tokens. This simplified model does not necessarily represent more complex consensus structures; however, we use it as a starting point for more nuanced models of PoS consensus.

A key property of PoS cryptocurrencies is that tokens can be re-allocated across various uses: the same token can be used for validation, for consumer and merchant transactions, and for off-chain routing of payments. Our analysis assumes, as an approximation, that tokens can be re-allocated in a frictionless manner. If a specific design deviates from this idealized frictionless model, e.g., due to constraints imposed on movements of tokens within the system, one would need to enrich the model with an explicit description of such frictions.

In a frictionless system, individual optimization by the network participants implies that expected rates of return on alternative feasible uses of the tokens must be equalized. In particular, any holder of a token may exchange it for fiat currency and invest the proceeds in a portfolio of financial assets with a return risk profile similar to that of the tokens. Thus, in a system where tokens can be easily exchanged without frictions for fiat currency, the rate of return on holding tokens must equal the opportunity cost: the expected rate of return on the risk-matched investment strategy in financial markets.

3.1 A stationary model with fee-based rewards

We first outline a parsimonious model in which rewards for validation activity are paid entirely in transaction fees generated by retail transactions rather than in newly minted tokens. This model abstracts away from many empirically-relevant details to focus on the main mechanism of token valuation.

Consider a payment network. Suppose that the network processes retail transactions with a flow rate Y_t, measured in the fiat currency, "dollars." Assume that the aggregate float of tokens is fixed, and without loss of generality equal to one. Let p_t denote the equilibrium price of the tokens in terms of the fiat currency. Assume that the network is in a stationary growth regime: the volume of transactions is expected to grow at a constant rate g_Y, so that $\mathbb{E}_t[Y_{t+s}] = Y_t \exp(g_Y s)$.

In this model we do not distinguish between block proposal and block validation functions, and refer to block proposers and block validators simply as "validators." We assume that the protocol for transactions is such that, whenever transactions take place, some tokens must be transferred to the validators as rewards for their activities, and, importantly, individual validators collect their pro-rata share of the total rewards, in proportion to their token stake. The latter assumption rules out individual strategies like selfish mining (Eyal and Sirer, 2018), which may distort the allocation of rewards among validators in relation to their individual token balances. We assume that the aggregate amount of fees generated by the network grows at the same long-run rate as the aggregate transaction volume (the ratio of the two series is a stationary process). To simplify the derivations in our model, we further restrict aggregate fees to be a constant multiple of the transaction volume – thus, validators receive fees in the collective amount of cY_t dollars per period (actual rewards are in tokens, which validators sell to consumers in exchange for dollars). We should note that this assumption applies to the total volume of fees, rather than the fee structure for individual transactions. Our assumption of aggregate fee dynamics is consistent with multiple alternative fee schedules, and, in essence, states that the total volume of fees is co-integrated with the total volume of transactions. Importantly, we do not assume that fees for individual transactions are set as a fixed fraction of the transaction value (see the Appendix for an example of a design with aggregate fees growing in proportion to the total transaction volume, and individual fees independent of the transaction value). Equilibrium fee structure depends closely on the design details of the payment systems (e.g., on the protocol for block formation and fee distribution among various actors).[5] How fees should be determined in a payment systems, and related implications for economic efficiency of the system are important questions for future research and beyond the scope of this paper. The valuation framework we develop here, however, should be useful in addressing such questions.

[5] See Lewenberg, Sompolinsky, and Zohar (2015), Pass and Shi (2017), Kiayias et al. (2017), and Huberman, Leshno, and Moallemi (2017) for analysis of fee mechanisms and alternative design suggestions.

We assume that validators have unrestricted access to financial markets and behave competitively: they take market prices, and, importantly, *the design of the payment network*, as exogenous and not affected by their individual actions. Also, we assume that the risk premium associated with a financial claim paying Y_t dollars per period is constant and equal to λ_Y (in equilibrium, this determines the opportunity cost of capital associated with validation activity). Finally, we also assume that there are no physical costs associated with block validation activities.

We look for a stationary equilibrium, in which $\phi \in [0, 1]$ tokens are held by the validators, and $1 - \phi$ are held by consumers for transaction purposes. We assume that validators have no use for tokens outside of their validation activity, and therefore stake their entire token balance. The equilibrium distribution of token holdings, ϕ, is endogenous and determined as a part of the solution.

In equilibrium, the total market value of all the tokens held by the validators is ϕp_t, which is the value of a financial claim on the perpetual stream of cash flows in the amount of cY_t. Assuming no valuation bubbles, the market value of this cash flow stream is given by the valuation formula for a perpetuity with constant growth:

$$p_t \phi = \lim_{T \to \infty} \int_0^T c\mathbb{E}_t[\exp(-\lambda_Y s) Y_{t+s}]\, ds = \frac{cY_t}{\lambda_Y - g_Y}. \tag{2}$$

To pin down the value of the tokens, we need to make an assumption about consumer's demand for tokens. In a market with infinite token velocity, consumers would hold no balances, which would imply that in equilibrium $\phi = 1$ and $p_t = cY_t/(\lambda_Y - g_Y)$. More generally, consumers hold balances of tokens for liquidity reasons. We assume that consumer balances are proportional (with a constant coefficient k) to the rate of transactions. Then,

$$p_t(1 - \phi) = kY_t \tag{3}$$

Combining (2) and (3), we find the equilibrium token value:

$$p_t = \left(k + \frac{c}{\lambda_Y - g_Y} \right) Y_t. \tag{4}$$

Note that, because consumer demand for token balances is non-negative, $cY_t/(\lambda_Y - g_Y)$ serves as a lower bound on the token value.

The equilibrium value of the tokens in (4) consists of two terms. The first term, kY_t, reflects consumer demand for holding token balances. The value of this term depends on k, which is inversely related to the equilibrium token velocity. It is important to acknowledge that token velocity is an equilibrium outcome, related to a number of properties of the payment network and the broader market. Wider adoption of the tokens could raise demand for token balances, while efficient channels for transactions between tokens and fiat currency would enable consumers to support the desired volume of transactions with lower token balances and result in higher token velocity.

The relation between token value and token velocity is commonly invoked when discussing the valuation of cryptocurrencies in PoW systems. While token velocity and equilibrium token value are certainly related in equilibrium, the relation between the two, like its analog in traditional monetary economics, is not a true structural relation and it does not provide a reliable anchor for token valuation.

The second term in the valuation equation (4), $cY_t/(\lambda_Y - g_Y)$, reflects the demand for tokens from validators. This term is proportional to the overall volume of retail transactions, Y_t, and to the rate at which fees are charged for transactions. All else equal, broader adoption and utilization of the payment system (higher Y_t) result in higher value of the tokens. Importantly, the above equation does not suggest that token value is increasing in the level of fees. Our analysis here focuses on a single stationary equilibrium and does not explicitly describe how the system responds to changes in parameters: higher fee levels would eventually lead to lower transaction volume.[6]

Our analysis in this section also relies critically on the assumption of competitive behavior by the network participants. To what extent this assumption offers a good approximation of agent behavior in this environment depends on individual opportunities and incentives to engage in strategic behavior. Ultimately, individual incentives and token valuation are closely linked in PoS systems, and must be analyzed jointly. This represents an important direction for future research.

[6] The effect of transaction fees on token value is analogous to the concept of the Laffer Curve in the theory of taxation.

3.2 An extension with increasing token supply

We now extend our valuation model to allow for increasing token supply and transitory dynamics. Specifically, we allow validators to be rewarded in newly minted tokens in addition to the fees collected from consumer transactions.

We now allow the transaction volume growth parameter to vary in time, and denote it by $g_Y(t)$. We also explicitly describe randomness in the evolution of transaction volume Y_t:

$$\frac{dY_t}{Y_t} = g_Y(t)\,dt + \sigma(t)\,dZ_t, \tag{5}$$

where $g_Y(t)$ and $\sigma(t)$ are bounded continuous functions of time, and Z_t is a Brownian motion.[7] Here Z_t is a simple model for the randomness in the system, and $\sigma(t)$ represents the instantaneous volatility of growth in transaction volume. Investors require compensation for being exposed to dZ_t shocks based on the comparable investment opportunities in financial markets – we denote the market price of risk of dZ_t shocks by η. For simplicity, we assume that η is constant. Then, under the risk-neutral valuation measure Q, the transaction volume follows

$$\frac{dY_t}{Y_t} = g_Y^Q(t)\,dt + \sigma(t)\,dZ_t^Q, \tag{6}$$

where

$$g_Y^Q(t) = g_Y(t) - \eta\sigma(t), \tag{7}$$

and Z_t^Q is a Brownian motion under measure Q.

As above, p_t denotes the total value of outstanding tokens at time t (effectively, we normalize the total supply of tokens to one) and validators hold fraction ϕ_t of all tokens. The market clearing condition requires that validators and consumers collectively hold all tokens, and thus

$$\phi_t p_t + kY_t = p_t, \tag{8}$$

where k is a consumer demand parameter, which we again assume to be constant. Thus,

$$\phi_t = 1 - k\frac{Y_t}{p_t}. \tag{9}$$

[7] We implicitly assume that all elements of the model, e.g., the growth rate of Y_t, are properly restricted so that tokens have a finite value in equilibrium.

We assume that token supply grows deterministically over time. Specifically, we assume that new tokens are issued at a bounded rate $g_S(t)$. In expectation under the risk-neutral valuation measure, validators earn the risk-free rate of return r_f, which we assume to be constant. Thus, we obtain the valuation equation:

$$\phi_t p_t = cY_t\,dt + e^{-r_f dt} E_t^Q\big[(\phi_t + (1-\phi_t) g_S(t)\,dt) p_{t+dt}\big]. \tag{10}$$

On the left-hand side, $\phi_t p_t$ is the total market value of the tokens staked by the validators at time t. On the right-hand side, we have two terms. The first term, $cY_t\,dt$, is the flow of transaction fees that accrue to the validators over the infinitesimal period $[t, t+dt)$. The second term, $e^{-r_f dt} E_t^Q\big[(\phi_t + (1-\phi_t) g_S(t)\,dt) p_{t+dt}\big]$, is the discounted expected value (under measure Q)) of the validators' token holdings at the end of the period.[8] Then,

$$E_t^Q\big[dp_t\big] = r_f p_t\,dt - \frac{p_t}{p_t - kY_t}\left(cY_t + kY_t g_S(t)\right)\,dt. \tag{12}$$

We look for an equilibrium token price process p_t of the form $p_t = p(t, Y_t)$, where $p(t, Y)$ is a sufficiently smooth function of its arguments. Applying Ito's lemma, we obtain a PDE on the token price function:

$$\frac{\partial p}{\partial t} + \frac{\partial p}{\partial Y} g_Y^Q(t) Y + \frac{1}{2}\frac{\partial^2 p}{\partial Y^2}\sigma(t)^2 Y^2 - r_f p + \frac{p}{p - kY}(c + k g_S(t))Y = 0. \tag{13}$$

Equation (13) is the valuation PDE. We look for a non-negative solution without valuation bubbles. Specifically, we look for a solution with suitably bounded growth in Y, and subject to a boundary condition

$$p(t, 0) = 0, \tag{14}$$

which requires that token value vanishes at zero transaction volume (recall that zero is an absorbing boundary for the transaction volume process). The above equation has a linear

[8] To see how this term is determined, note that all new tokens accrue to the validators. Therefore, if validators start the period $[t, t+dt)$ with a fraction ϕ_t of the tokens, they end the period with

$$\frac{\phi_t + g_S(t)\,dt}{1 + g_S(t)\,dt} = (\phi_t + g_S(t)\,dt)(1 - g_S(t)\,dt) + o(dt) = \phi_t + (1-\phi_t) g_S(t)\,dt + o(dt) \tag{11}$$

tokens.

solution,

$$p(t, Y) = A(t)Y, \tag{15}$$

where the unknown function $A(t)$ is a bounded, positive solution of the ODE:

$$\frac{dA(t)}{dt} + g_Y^Q(t)A(t) - r_f A(t) + \frac{A(t)}{A(t) - k}(c + kg_S(t)) = 0. \tag{16}$$

In the stationary case of $g_S(t) = g_S$, $\sigma(t) = \sigma$, and $g_Y(t) = g_Y$, $g_Y^Q = g_Y - \eta\sigma$, the total value of tokens p_t is a constant multiple of the transaction volume Y_t:

$$\frac{p_t}{Y_t} = k + \frac{c + kg_S}{r_f - g_Y^Q}. \tag{17}$$

This solution describes the total value of tokens under constant growth rate in the number of outstanding tokens. Note that, $r_f + \eta\sigma = \lambda_Y$, which is the expected return on the financial claim paying a cash flow stream equal to the aggregate flow of transaction fees, cY_t. We thus recover the valuation formula (4) by setting the token rewards to zero, $g_S = 0$.

The above solution highlights the valuation effect of rewarding validators in newly minted tokens. In addition to collecting transaction fees, validators also collect proceeds from seignorage. Comparing with (4), we see that this effectively raises the flow of proceeds to validators from c to $c + kg_S$. The second term, kg_S is intuitive: transfers to validators due to seignorage are proportional to the level of token balances held by consumers. If consumers hold no token balances between transactions, any benefit validators derive from collecting rewards in newly minted tokens is completely offset by the decline in the market value of tokens in their stake.

4 Network Security and Valuation Bubbles

In this section we are concerned with a particularly important type of malicious behavior: the majority, or "51%" attack. Feasibility of such attacks depends on the tradeoff between the benefit and the cost of attacks. The direct economic benefit is derived from the attackers' ability to double-spend coins on multiple transactions (the attackers may also be motivated by the objective of destroying credibility of a particular blockchain system, in which case the benefit of the attack is harder to quantify). The cost depends on the requirements for a successful attack, which in turn depend on the defensive mechanisms implemented in the network.

The list of empirical examples of successful majority attacks is growing. For example, on May 16th of 2018 attackers stole \$17.5M by executing the 51% attack on Bitcoin Gold (BTG). As recently as January 5th of 2019, Ethereum Classic (ETC) suffered a successful 51% attack, which allowed the attackers to double-spend approximately \$1,1M worth of tokens. The overall value of Ethereum Classic tokens declined by roughly 20%, or \$100M, over the week following the attack.

In a recent paper, Budish (2018) analyses the incentives behind the majority attacks in the PoW systems, like Bitcoin. In such systems, majority attacks require attackers' to gain majority of hashing power over a period of time, and hence the cost of the attack depends on the cost and liquidity of the mining equipment. In the PoS system, attackers face a very different tradeoff: to execute the majority attack, one must gain a sufficient degree of majority in the validation pool, which requires holding a significant number of tokens. [9] The cost of the attack is that ones' token holdings would be slashed (confiscated) based on the network security protocol. Majority attacks are only feasible if their economic benefits exceed the cost of executing the attack – hence, network security hinges critically on token valuation.

4.1 A security condition

Suppose there are N regularly spaced finalization points per year. We call the span of time between two consecutive finalization points an *epoch*. If Y_t is the current expected annualized rate of arrival of dollar transaction volume on the network, a successful double-spend attack would yield on average Y_t/N dollars worth of transactions. At this point, we need to recognize a simplification introduced within our model: we model the flow of transactions as smooth. This assumption is not accurate at very short time scales – it is more realistic to think of Y_t in the model as an expected rate of transaction arrival, with the actual transaction volume per epoch being a lot more random than implied by the process (5). Thus, Y_t/N is an expected

[9] Here, we are assuming that users only consider a transaction confirmed once it has been finalized. In reality, there may be confirmation rules that do not rely only on finalization, but finalization is at least a sufficient condition for confirmation, by definition.

value of transactions in the next epoch, and attacks are likely to be launched selectively to take advantage of epochs with relatively high transaction values. Our analysis thus yields an estimate of the level of revenue, $\gamma Y_t/N$, at which a double-spend attack becomes profitable, for a given set of network and market parameters. We use γ to express our answer, which effectively formulates it in units of time: a particular epoch is attractive for an attacker if it contains contains value in excess of a factor of γ of the expected epoch value.

To quantify the cost of executing an attack, suppose the network protocol requires the 2/3 majority to validate a group of blocks between finalization points. This is the case in Ethereum's Casper FFG, for instance Buterin and Griffith (2017). In that case, an attacker could execute a double-spend attack while controlling 2/3 of the total pool of tokens staked by validators. Such an attack would result in an attacker losing 1/3 of the validation pool in tokens (half of the attacker's stake) to slashing. To see this, consider a situation where the same tokens are used for two separate transactions, and both get validated. First time around, all the "honest" validators vote on the transaction, which accounts for 1/3 of the validation pool, and attacker contributes another 1/3 to the vote. Second time around, the attacker validates transactions unilaterally, using its 2/3 control of the validation pool. As a result, 1/3 of all the tokens staked by the validators ends up slashed as a penalty for validating invalid blocks. We denote the fraction of the validation pool lost due to slashing by $\mathscr{Z}$: under alternative rules it would differ from 1/3.

Note that validators earn a fair rate of return on their capital. Thus, the only other cost the attacker faces in addition to slashing is the subsequent decline in the value of its remaining stake (because of the network losing credibility following a successful attack) We do not model this value, which yields a somewhat conservative estimate of the cost of the attack (our argument for $\mathscr{Z} = 1/3$ in the previous paragraph may understate the cost of the attack by a factor of two, at the most).

Suppose the system is following a steady-state growth path, and there is no token supply. Using the valuation equation (2), the economic loss to slashing equals the benefit of double-spending if[10]

$$\mathcal{Z}\frac{cY_t}{\lambda_Y - g_Y} = \gamma\frac{Y_t}{N}, \tag{18}$$

and therefore

$$\gamma = \frac{N\mathcal{Z}c}{\lambda_Y - g_Y}. \tag{19}$$

To get a sense of the economic magnitudes involved, suppose that $N = 52,000$ (this places finalization points roughly ten minutes apart, $\lambda_Y = 7\%$, $g_Y = 2\%$, and $c = 0.10\%$. These numbers are real (inflation-adjusted). They imply that transaction volume on the network grows roughly at the same rate as GDP; the expected return on a claim on the flow of fees is roughly in line with expected real returns on the stock market (in the absence of empirical data, one could plausibly argue for a higher or a lower value); and average transaction fees are at a somewhat arbitrary level of 0.1%, which is much lower than the transaction fees charged on smaller transactions by credit cards, but higher than the cost of wiring funds through the traditional systems. With the assumed parameters, in steady state, the security threshold $\gamma \approx 350$. Thus, a double-spending attack may be worth executing if the value stolen exceeds two and a half days worth (approximately 350 ten-minute periods) of average transaction volume on the network. As long as the value of individual epochs is lower than this thresholds, double-spending attacks would not be launched by a profit-maximizing actor (this says nothing about an attacker motivated to undermine the network itself).

More generally, high token valuation relative to the economic footprint of the network serves as a deterrent against majority attacks. While there may not exist a particular level of token valuation that guarantees network security (e.g., because the benefit of an attack may be unbounded, if the objective is to undermine the network), we can nonetheless obtain useful information by observing how various design features and events affect token values.

4.1.1 Valuation bubbles

In this section we show that valuation bubbles may adversely affect network security. Consider a network with a constant growth rate and volatility of the transaction volume, zero supply of new tokens, and stochastic consumer demand for token holdings. Specifically, we

[10] we assume as an approximation that the total flow of transactions between the consequtive finalization points is Y_t/N.

assume that consumer demand follows a continuous-time Markov process, taking on one of two values: $k_H > k_L$. The transition rate from state k_i to k_j is ν_{ij}. We assume that this process is independent of the transaction volume process Y_t, and shocks to consumer demand carry no risk premium – hence, the same transition rates apply under the risk-neutral valuation measure Q.

The time-varying consumer demand for token balances captures, in reduced form, speculative trading by investors in tokens. The state with high token balances, $k_t = k_H$, can be interpreted as a state in which investors are relatively optimistic about future token price appreciation. Their belief is in fact irrational: starting in the high state, returns from holding tokens are relatively low. In contrast, we assume throughout that validators hold rational expectations about future token prices.[11] Note that while we allow for transient speculative consumer demand, we continue to rule out rational valuation bubbles, which are associated with unbounded expected asymptotic token price appreciation.

Under the above assumptions, token values in a stationary equilibrium are given by

$$p(t, Y) = A(k_t)Y, \tag{20}$$

where $A(k_t)$ satisfies the following system of equations:

$$(g_Y^Q - r_f)A(k_H) + \frac{A(k_H)}{A(k_H) - k_H}c + \nu_{HL}(A(k_L) - A(k_H)) = 0, \tag{21}$$

$$(g_Y^Q - r_f)A(k_L) + \frac{A(k_L)}{A(k_L) - k_L}c + \nu_{LH}(A(k_H) - A(k_L)) = 0. \tag{22}$$

The security thresholds γ_H and γ_L are then given by

$$\gamma_i = \mathcal{Z}N(A(k_i) - k_i), \quad i \in \{L, H\}. \tag{23}$$

The above equation reflects that $(A(k_t) - k_t)Y_t$ is the market value of the tokens stakes by the validators, at time t.

Intuitively, network security may be compromised in a high-valuation state, k_H. This

[11] We model speculative bubbles directly through time-varying consumer demand for convenience. One can, alternatively, model biased believes of investors directly. This would not change the qualitative conclusions in this section: we will make this analysis available in an Internet Appendix.

happens because, starting in this state, consumer demand for balances is expected to decline, and hence expected capital gains on holding tokens are low. To compensate for that, and allow validators to earn a fair expected rate of return on their token stakes, the overall size of the validator pool must be low in equilibrium, so that the yield from transaction fees offsets the low expected changes in the token price. This implies that the overall value of the validation pool is relatively low, and hence the network is volnurable to a majority attack.

A numerical illustration helps confirm this intuition. Assume that interest rate r_f is 1%; growth rate of transaction volume, g_Y, is 2%; the market price of risk of shocks to Y_t, η, is 0.5; volatility of transaction volume growth σ is 0.12; and the transaction fee parameter is $c = 0.1\%$. These numbers are consistent with the numerical example is Section 4.1.

To set the baseline, assume first that $k_L = k_H = 15/365$, i.e., consumer demand for balances is constant, and equal to approximately 15 days worth of spending. In this case, $A(k_H) = A(k_L) = 0.061$, and $\gamma(k_H) = \gamma(k_L) = 347$ – we obtain the same security threshold as in the previous section. Note that the security condition is determined by the value of the validators' holdings, rather than by the total value of all outstanding tokens. Hence, we obtain the same security threshold as in the previous section, independently of the assumed value of consumer demand.

Next, we let consumer demand differ across states: we assume that $k_H = 1/4$, $k_L = 15/365$, $\nu_{LH} = 0.1$, and $\nu_{HL} = 1$. Thus, consumer balances in the high (optimistic) state are equal to approximately three months worth of spending, compared to 15 days in the low state. Transition rates across the two states are highly asymmetric, so that the network spends only 9% of the time in the high state, and transitions out of that state, on average, in a year. In contrast, it takes on average ten years to enter the high state, starting from a low state.

Tokens are more valuable in a high state, due to higher consumer demand for balances:

$$A(k_H) = 0.253, \quad A(k_L) = 0.177. \tag{24}$$

The market value of the validators' holdings is highly asymmetric across the two states: in the low state, validators hold a fraction $\phi_L = 0.768$ of the entire supply of tokens. In contrast, in the high state, they hold only $\phi_H = 0.011$ of all the tokens. As we discuss above, low holdings by validators are caused by the anticipated token price decline associated with the impending "burst of the bubble." Security thresholds are, accordingly, very different across the two states: $\gamma_H = 50$, and $\gamma_L = 2,359$. Note that the possibility of a valuation bubble forming in the future makes the network more secure in the low state, by raising the value of the validators' holdings. The opposite happens in the high state. Under the assumed parameters, the security threshold declines from 347 in the baseline case (without the valuation bubble), to 50. This means that a profitable double-spend attack requires capturing roughly eight hours worth of average transaction volume in a single epoch, compared to two and a half days in the baseline case.

Certain design feature may be introduced to help reduce the negative effect of valuation bubbles on validators' willingness to stake tokens. For instance, one may lock in validators' holdings over extended time periods, making it impossible for them to respond to highly transient valuation changes. In return, validators would then require a higher expected rate of return in equilibrium, to compensate them for the loss of liquidity. Understanding the net effect of such arrangements on network security, and network dynamics in general, is an important question for future research.

4.1.2 State-contingent monetary policy

In this section, we show that a state-contingent token supply policy could be used to mitigate the effect of valuation bubbles on network security. Despite of the apparent similarity, this idea is quite different from the question of whether traditional monetary policy should be used to lean again valuation bubbles. The latter is a complex and controversial issue, primarily because valuation bubbles are difficult to identify in real time, leading to costly false positives, and because tools of traditional monetary policy may not be effective at slowing down bubbles. In the context of the PoS payment systems, the problem is much more tractable and, importantly, the mechanism through which the monetary policy interacts with the bubbles is very different in this context.

First, the objective of the state-contingent monetary policy is not to lean against the valuation bubble (we assume, for simplicity, that token holdings of 'speculators' are exogenously driven and this not affected by the monetary policy), but rather to mitigate the effect of the bubble on the size of the validator pool, with associated implications for network security.

Second, as we show below, monetary policy could be quite effective at controlling the size of the validator pool on the downside, which is its purpose. The reason is that an increase in the rate of token supply has a direct effect on the rate of return achieved by validators. The strength of this effect is inversely related to the size of the validator pool. Hence, raising the token supply rate while the size of the validator pool is deemed below the desired level, leads to a sharp increase in the validators' rate of return, thus attracting more capital to the pool.

Finally, the token supply rules we consider in this section can be implemented without taking a stand on what level of token valuation crosses in the the 'bubble' territory. Instead of connecting token supply directly to token valuation, cne could condition token supply on the size of the validator pool, or, more generally, on some function of its history. In the model of Section 4.1.1, for instance, the size of the validator pool is a one-to-one function of the state of consumer demand for balances. Hence, in this section, we assume for simplicity that the token supply schedule can be conditioned directly on such a state.

To illustrate this idea, consider the following state-contingent monetary policy. Consider the example of the valuation bubble in the previous section, and assume that the token supply rate is $g_S(k_L) = 0.01$ in the L-state, and $g_S(k_H) = M g_S(k_L)$ in the H-state, where the multiple M ranges from 1 to 10. Thus, monetary policy is contingent on the state of consumer demand for balances, or, equivalently, on the size of the validator pool. Figure 2 shows the results.

The first panel in Figure 2 shows how token valuation depends on the state of consumer demand, and on M, which is the ratio of the token supply rate in state k_H vs state k_L. As observed in the previous section, token value is high with higher consumer demand for balances. Moreover, token value is increasing in inflation rate, since higher inflation rate effectively raises the flow of fees collected by the validators. While the rate of token supply in the L-state is fixed, token value in this state is increasing in M, because this value if forward-looking and discounts the possibility of the future transition into the H-state, with a higher token supply rate.

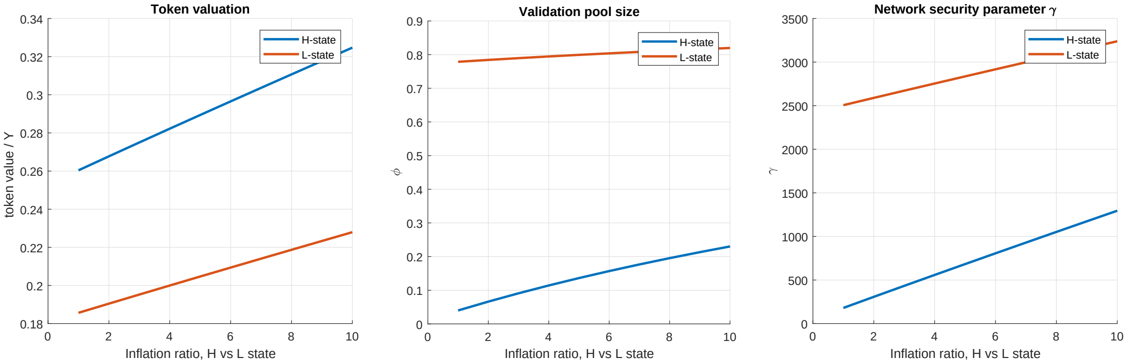

Figure 2: Effect of the state-contingent monetary policy on token valuation and network security.

The second panel shows the size of the validation pool. With the state-independent monetary policy, i.e., $M = 1$, validator pool is much smaller in state H. As we raise the token supply rate in the H-state, validator pool size rises sharply. The third panel reflects the resulting effect on network security: the network security parameter γ is significantly higher at higher values of M.

Our numerical example illustrates the general intuition: higher token supply rate in the H state helps mitigate the negative effect of the expected decline in valuations, due to the "burst" of the valuation bubble.

5 Payment Channel Networks and Endogenous Token Velocity

Increasing transaction throughput and reducing confirmation delay of blockchains has been a topic of major interest in recent years (Chen and Micali, 2016; Bagaria et al., 2018; Yu et al., 2018; Pass and Shi, 2018b; Eyal et al., 2016). An important class of approaches to this problem is known as *layer 2* solutions. The key idea behind layer 2 solutions is that instead of processing every transaction on the (slow, low-throughput) main blockchain, transactions can be processed in batches.

Payment channel networks (PCNs) are a prominent example of a layer 2 solution in which users build an overlay network, in which each node is a user and each edge (or *channel*) represents escrowed funds that can only be sent to/from the two endpoint users for a fixed time period (Poon and Dryja, 2016). PCNs enable users to send money even if they do not already share a dedicated payment channel; to achieve this, the payment is relayed over a path of channels connecting the two users in the overlay network. However, once a transaction is complete, it is only recorded by the participants (the endpoints and the relay nodes), rather than being committed to the public blockchain. Finally, these off-chain transactions can be periodically recorded in bulk on the main blockchain. Critical for our purposes, relay nodes charge a fee, which is collected once the batched on-chain transactions are processed. PCNs can be used to significantly increase transaction throughput (e.g., Lind, Eyal, Pietzuch, and Sirer, 2016; Dong, Liang, Li, and Liu, 2018; Sivaraman, Venkatakrishnan, Ruan, Negi, Yang, Mittal, Fanti, and Alizadeh, 2020), and are viewed as a promising direction for improving blockchain scalability (Cointelegraph).

The challenge to network economics is that, while PCNs rely on the main blockchain for supporting off-chain transactions, transaction fees from off-chain transactions do not accrue to the operators on the main blockchain – proposers and validators. In our setting, inflation tax resulting from the increasing token supply is an essential mechanism by which some of the rents from the off-chain transactions can be directed back to the main blockchain.

Technical properties of the payment channel layer (algorithmic details of routing protocols, network topology, physical constraints on routing, transaction arrival processes, etc.) determine throughput within the payment channel layer in relation to the volume of tokens temporarily locked within the channels. This has a significant economic implication: a system combining the main blockchain with the PCN layer imposes an endogenous limit on token velocity, which is driven by the technological properties of the second layer rather than by user preferences for liquidity.

While detailed analysis of system with the PCNs is beyond the scope of the current model, and would require a model of the payment network dynamics, some insight can be gained from the following simplified description. Suppose that all of the retail transactions in the system are processed through the PCN layer. Assume further that the network's capacity (i.e., the maximum achievable throughput Y_t/p_t) grows as a function of B_t; for instance, if the relation is linear with proportionality constant R, then we assume the PCN can support any throughput $Y_t/p_t \leq R\,B_t$. This assumption is not saying anything about the instantaneous demand; rather, it is saying that as the collateral locked in the network grows, the network is capable of processing higher transaction throughput. This assumption is clearly reasonable only up to a point. With enough collateral, the PCN will be network-bound rather than collateral-bound; that is, constraints on the communication network's throughput will dominate the PCN's inherent limits on transaction throughput. However, prior work suggests that existing PCN designs operate far from this limit (Sivaraman, Venkatakrishnan, Alizadeh, Fanti, and Viswanath, 2018).

The important distinction between this formulation and the approach specifying consumer token balances in relation to their transaction needs is that we now impose the optimality condition on the token holdings within the PCN channels. In particular, we impose the requirement that balances of tokens held within the PCN channels earn the same opportunity cost as the balances held by the validators on the main blockchain. This effectively endogenizes the velocity of tokens in the network.

We assume that, unlike the validators, operators of the payment channels do not benefit from the issue of new tokens, and are paid exclusively in transaction fees. In turn, the main blockchain does not process retail transactions directly, and instead operates as the backbone for the PCN layer. Validators on the main blockchain are paid exclusively in new tokens. As before, we assume that volume of fees (in dollars) is proportional to the volume of processed transactions, so the rate of flow of transaction fees to the PCN is cY_t. As before, the equilibrium level of the validation pool is ϕ_t, as a fraction of the total number of outstanding tokens. Then, in equilibrium, the fraction of tokens held in the PCN channels is $1 - \phi_t$.

For simplicity, assume that the system is stationary : $g_S(t) = g_S$, $\sigma(t) = \sigma$, and $g_Y(t) = g_Y$. The condition that the validators earn the arbitrage-free rate of return is now given by

$$\phi_t p_t = e^{-r_f dt}\mathbb{E}_t^Q\big[(\phi_t + (1-\phi_t)g_S\,dt)p_{t+dt}\big]. \tag{25}$$

Note that in comparison to (10), validators no longer earn transaction fees, and their return comes entirely from rewards in newly minted tokens.

Conjecture that in equilibrium, the PCN layer has sufficient capacity to process the entire volume of consumer transactions, Y_t. Holders of balances in the PCN channels must earn the fair rate of return as well, which implies that

$$(1-\phi_t)p_t = cY_t\,dt + e^{-r_f dt}\mathbb{E}_t^Q\big[(1-\phi_t)(1-g_S\,dt)p_{t+dt}\big]. \tag{26}$$

Equations (25-26) jointly determine the process for the token value p_t and the equilibrium allocation of tokens between the validators (ϕ_t) and the PCN channels $(1-\phi_t)$. Solving these two equations yields the equilibrium price process for the tokens, and the distribution of token holdings between the PCN layer and the validators on the main blockchain:

$$p_t = \frac{cY_t}{r_f - g_Y^Q}, \quad \phi_t = \frac{g_S}{g_S + r_f - g_Y^Q}. \tag{27}$$

Intuitively, the value of the total supply of tokens, p_t, equals the present value of all the fees flowing into the system, since all the actors in the network earn the fair rate of return on their token holdings. The share of the tokens held by the validators is increasing in the rate of token supply g_S, since this rate determines how heavily the PCN layer is taxed by the main blockchain, and therefore how attractive it is to use token balances for validation relative to supporting PCN transactions.

Note that the above solution is valid as long as our conjecture on the capacity of the PCN layer holds: $Y_t/p_t \geq R(1-\phi_t)$, which is equivalent to $g_S + r_f - g_Y^Q \geq cR$. If this condition is violated, the PCN layer cannot process the entire volume of transactions Y_t. To resolve this, we need to recognize that the transaction fees charged by the PCN layer are endogenous, and must be determined in equilibrium jointly with the token balances in the payment channels, and consumer transaction volume. To carry out such analysis, we would need a more detailed analytical model of the PCN, which is currently beyond the scope of this paper.

6 Conclusion

We've developed a valuation framework for tokens in Proof-of-Stake payment systems. Our analysis relies on the key observation that in a frictionless equilibrium, the rate of return earned by the validators by staking tokens must equal their alternative rate of return in financial markets, based on investments with the same risk level.

In our analysis, we assume that validators are free to reallocate their tokens instantaneously and without frictions. This is an idealization, and current designs and proposals for PoS systems include various restrictions on validators, intended to curtail various types of network attacks by allowing network participants to penalize validators for recent violations of the network protocol. Our valuation framework could be used to understand the effect of reduced liquidity of validators' holdings on token valuation dynamics and network security conditions.

Another important direction for future work is to develop a more detailed model of the payment channel network, as a part of a larger PoS payment system. While the existing work on this topic focuses on the routing algorithms, and throughput capacity of the PCN, it is also important to consider the equilibrium effect of economic incentives on token holdings within the PCN channels, and the joint determination of PCN properties and token price dynamics in equilibrium.

Finally, our analysis provides a useful starting point for thinking about economic tradeoffs between the Proof-of-Stake and the Proof-of-Work protocols.

References

Athey, Susan, Ivo Parashkevov, Vishnu Sarukkai, and Jing Xia. "Bitcoin Pricing, Adoption, and Usage: Theory and Evidence." *Working Paper*, (2016).

Bagaria, Vivek, Sreeram Kannan, David Tse, Giulia Fanti, and Pramod Viswanath. "Deconstructing the Blockchain to Approach Physical Limits." (2018). `https://arxiv.org/abs/1810.08092`.

Biais, Bruno Biais, Christophe Bisiere, Matthieu Bouvard, Catherine Casamatta, and Albert Menkveld. "Equilibrium Bitcoin Pricing." *Working Paper*, (2018).

Brown-Cohen, Jonah, Arvind Narayanan, Alexandros Psomas, and S Matthew Weinberg. "Formal barriers to longest-chain proof-of-stake protocols." In "Proceedings of the 2019 ACM Conference on Economics and Computation," 459–473.

Budish, Eric. "The Economic Limits of Bitcoin and the Blockchain." *NBER Working Paper No. 24717*, (2018).

Buterin, Vitalik and Virgil Griffith. "Casper the friendly finality gadget." *arXiv preprint arXiv:1710.09437*, (2017).

Chen, Jing and Silvio Micali. "Algorand." *arXiv preprint arXiv:1607.01341*, (2016).

Cointelegraph. "What Is Lightning Network And How It Works." `https://cointelegraph.com/lightning-network-101/what-is-lightning-network-and-how-it-works` (????). Accessed on September 8, 2020.

Cong, Lin William, Ye Li, and Neng Wang. "Tokenomics: Dynamic Adoption and Valuation." *Working Paper*, (2018).

David, Bernardo, Peter Gaži, Aggelos Kiayias, and Alexander Russell. "Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain." In "Annual International Conference on the Theory and Applications of Cryptographic Techniques," Springer, 66–98.

Donet, Joan Antoni Donet, Cristina Pérez-Sola, and Jordi Herrera-Joancomartí. "The bitcoin P2P network." In "International Conference on Financial Cryptography and Data Security," Springer, 87–102.

Dong, Mo, Qingkai Liang, Xiaozhou Li, and Junda Liu. "Celer network: Bring internet scale to every blockchain." *arXiv preprint arXiv:1810.00037*, (2018).

Eyal, Ittay, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. "Bitcoin-NG: A Scalable Blockchain Protocol." In "NSDI," 45–59.

Eyal, Ittay and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable." *Communications of the ACM*, 61(2018), 95–102.

Fan, Lei and Hong-Sheng Zhou. "A scalable proof-of-stake blockchain in the open setting (or, how to mimic Nakamoto's design via proof-of-stake)." Technical report, Cryptology ePrint Archive, Report 2017/656 (2017).

Fitzi, Matthias, Peter Gazi, Aggelos Kiayias, and Alexander Russell. "Parallel Chains: Improving Throughput and Latency of Blockchain Protocols via Parallel Composition." *IACR Cryptology ePrint Archive*, 2018(2018), 1119.

Gilad, Yossi, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. "Algorand: Scaling byzantine agreements for cryptocurrencies." In "Proceedings of the 26th Symposium on Operating Systems Principles," ACM, 51–68.

Huberman, Gur, Jacob D Leshno, and Ciamac C Moallemi. "Monopoly without a monopolist: An economic analysis of the bitcoin payment system." (2017).

Kerber, Thomas, Aggelos Kiayias, Markulf Kohlweiss, and Vassilis Zikas. "Ouroboros crypsinous: Privacy-preserving proof-of-stake." In "2019 IEEE Symposium on Security and Privacy (SP)," IEEE, 157–174.

Kiayias, Aggelos, Alexander Russell, Bernardo David, and Roman Oliynykov. "Ouroboros: A provably secure proof-of-stake blockchain protocol." In "Annual International Cryptology Conference," Springer, 357–388.

Kosik, Bill. “Data centers used for bitcoin mining.” (2018).

Lee, Timothy. “Bitcoin’s insane energy consumption, explained.” *Ars Technica*, (2017).

Lewenberg, Yoad, Yonatan Sompolinsky, and Aviv Zohar. “Inclusive block chain protocols.” In “International Conference on Financial Cryptography and Data Security,” Springer, 528–547.

Lind, Joshua, Ittay Eyal, Peter Pietzuch, and Emin Gün Sirer. “Teechan: Payment channels using trusted execution environments.” *arXiv preprint arXiv:1612.07766*, (2016).

Liu, Yukun and Aleh Tsyvinski. “Risks and Returns of Cryptocurrency.” *Working Paper*, (2018).

Mazieres, David. “The stellar consensus protocol: A federated model for internet-level consensus.” *Stellar Development Foundation*, (2015).

Miller, Rich. “As Bitcoin Infrastructure Booms, Mining Heads to the Data Center.” *DataCenter Knowledge*, (2014).

Nakamoto, Satoshi. “Bitcoin: A peer-to-peer electronic cash system.” *Working Paper*, (2008).

Pagnotta, Emiliano and Andrea Buraschi. “An Equilibrium Valuation of Bitcoin and Decentralized Network Assets.” *Working Paper*, (2018).

Palai, Asutosh, Meet Vora, and Aashaka Shah. “Empowering light nodes in blockchains with block summarization.” In “2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS),” IEEE, 1–5.

Pass, Rafael and Elaine Shi. “Fruitchains: A fair blockchain.” In “Proceedings of the ACM Symposium on Principles of Distributed Computing,” ACM, 315–324.

———. “Thunderella: Blockchains with optimistic instant confirmation.” In “Annual International Conference on the Theory and Applications of Cryptographic Techniques,” Springer, 3–33.

———. “Thunderella: Blockchains with optimistic instant confirmation.” In “Annual International Conference on the Theory and Applications of Cryptographic Techniques,” Springer, 3–33.

Poon, Joseph and Thaddeus Dryja. “The bitcoin lightning network: Scalable off-chain instant payments.” (2016).

Sivaraman, Vibhaalakshmi, Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, Giulia Fanti, and Pramod Viswanath. “Routing Cryptocurrency with the Spider Network.” In “Proceedings of the ACM Hotnets 2018,” 127–138.

Sivaraman, Vibhaalakshmi, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. “High Throughput Cryptocurrency Routing in Payment Channel Networks.” In “17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20),” 777–796.

Sockin, Michael and Wei Xiong. “A Model of Cryptocurrencies.” *Working Paper*, (2018).

Wall, Eric. “Proof-of-Stake is less wasteful.” (2019). `https://medium.com/@ercwl/proof-of-stake-is-less-wasteful-b2854a191766`.

Wang, Xuechao, Govinda Kamath, Vivek Bagaria, Sreeram Kannan, Sewoong Oh, David Tse, and Pramod Viswanath. “Proof-of-Stake Longest Chain Protocols Revisited.” *arXiv preprint arXiv:1910.02218*, (2019).

Yang, Lei, Vivek Bagaria, Gerui Wang, Mohammad Alizadeh, David Tse, Giulia Fanti, and Pramod Viswanath. “Prism: Scaling Bitcoin by 10,000 x.” *arXiv preprint arXiv:1909.11261*, (2019).

Yu, Haifeng, Ivica Nikolic, Ruomu Hou, and Prateek Saxena. “OHIE: blockchain scaling made simple.” *IEEE Security and Privacy*, (2018).

7 Appendix: aggregate fee dynamics

Suppose that each transaction costs δ_t tokens (minimal fee), where we normalize the supply of tokens to one – we assume zero supply of new tokens for simplicity. Assume a stationary economy: the number of potential transactions follows a geometric random walk.

Let there be N_t transactions per unit of time. Assume that the second best cost of executing individual potential transactions in dollars (this refers to other forms of payment – instead of using the PoS system under consideration, one could use a competing network, for example.), v, be distributed randomly with a CDF $F(v)$. For concreteness, we assume a particularly functional form for this distribution:

$$F(v) = 1 - \frac{a}{\gamma} v^{-\gamma}, \quad v \geq \left(\frac{\gamma}{a}\right)^{-1/\gamma}. \tag{28}$$

For each transaction, let's assume that the cost of the second-best option is proportional to the dollar value of the transaction:

$$v = \varrho \times \text{Dollar transaction amount} \tag{29}$$

A transaction is going to take place if

$$v > \delta_t p_t \tag{30}$$

Thus, the distribution of executed transactions depends, in turn, on token value. The above equation describes consumer demand for transactions in our model.

Anticipating the result, we assume that the transaction fee, in tokens, is inversely related to the number of transactions:

$$\delta_t = \delta_0 N_t^{-1}. \tag{31}$$

We look for the solution with the token value proportional to the number of transactions:

$$p_t = A N_t \tag{32}$$

Under the above assumptions, the total number of transactions (per period) is

$$\#\ \text{Transactions} = N_t(1 - F(\delta_t p_t)) = N_t(\delta_0 A)^{-\gamma}, \tag{33}$$

with the total dollar value of transactions being

$$Y_t = N_t \times \int_{\delta p_t}^{\infty} \varrho^{-1} v \, dF(v) = N_t \frac{a}{\varrho(\gamma - 1)} (\delta_0 A)^{-\gamma} \tag{34}$$

and the total amount of fees paid into the system being

$$\text{Fees}_t = N_t \times (1 - F(\delta p_t))\delta p_t = N_t \frac{a}{\gamma} (\delta_0 A)^{1-\gamma} \tag{35}$$

We thus find a relation between transaction volume and fees paid: total fees are a constant fraction of the total transaction volume, even though at the individual transaction level, fees do not scale with the transaction amount:

$$\text{Fees}_t = \frac{\delta p_t}{\int_{\delta p_t}^{\infty} \lambda^{-1} v \, dF(v)} Y_t = c Y_t, \tag{36}$$

where

$$c = \frac{\varrho}{\varrho(\gamma - 1)} \delta_0 A. \tag{37}$$

To complete the derivation, we use the valuation equation (4):

$$p_t = A N_t = \left(k + \frac{c}{\lambda_Y - g_Y}\right) Y_t = \left(k + \frac{1}{\lambda_Y - g_Y} \frac{\varrho}{\varrho(\gamma - 1)} \delta_0 A\right) \frac{a}{\varrho(\gamma - 1)} (\delta_0 A)^{-\gamma} N_t, \tag{38}$$

which results in an equation on A,

$$A = \left(k + \frac{1}{\lambda_Y - g_Y} \frac{\varrho}{\varrho(\gamma - 1)} \delta_0 A\right) \frac{a}{\varrho(\gamma - 1)} (\delta_0 A)^{-\gamma}, \tag{39}$$

which has a unique solution.

In summary, we find that with the flat fee structure, and fee level set in tokens, in inverse proportion to the total number of transactions, the total fees collected are proportional to the transaction volume and our basic valuation result in (4) holds: the aggregate value of tokens is proportional to the transaction volume.