RELATIONSHIPS BETWEEN BLOCKCHAIN CONSENSUS PARAMETERS AND NETWORK EXTERNALITIES

by

JOSEPH V. ANDERSEN

A THESIS

Presented to the Department of Economics
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

February 2023

An Abstract of the Thesis of

Joseph V. Andersen for the degree of Bachelor of Science
in the Department of Economics to be taken June 2023

Title: Relationships Between Blockchain Consensus Parameters and Network Externalities

Approved: *Professor David Evans, PhD.*
Primary Thesis Advisor

Large-scale decentralized networks have many advantageous fault tolerance properties over their centralized counterparts. These properties enable resiliency in the face of faulty, and even adversarial behavior. Consensus protocols are a way to coordinate state across these networks and are what makes them extremely robust to adversaries. This paper investigates the effectiveness of different consensus protocol designs by examining various questions about the relationships between blockchain consensus parameters and network externalities. In this research we develop a linear regression model to estimate which characteristics of blockchains are associated with the highest levels of byzantine fault tolerance. This quantitative research provides evidence for whether there are statistically significant relationships between aspects of blockchain design and voting power centralization. Finally, we explore what key network metrics are used by node operators in the decision making process to participate in the mining process.

Acknowledgements

I would like to thank Professor David Evans and Professor Lindsay Hinkle for helping me thoroughly examine my topic and consider the various perspectives and contexts related to this subject matter. I would like to extend my deepest gratitude to Professor Evans for supporting my decision to write my thesis on a topic I knew almost nothing about before starting this process, and encouraging me every step of the way. I also want to thank Professor Hinkle for having such a positive impact on my education within the Clark Honors College and at the University of Oregon.

I would also like to thank the countless academics, researchers, and developers who open source their work and make it available to all. This project would not have been possible without the extensive resources made available by the free and open-source software community.

Table of Contents

List of Accompanying Materials

1. Thesis.Rmd (R file used to generate graphs and regression models)
2. PartialSynchrony_BFT.csv
3. Export-AddressCount.csv
4. Export-BlockDifficulty.csv
5. Export-MarketCap.csv
6. Export-NetworkUtilization.csv
7. Export-TxGrowth.csv

List of Figures

Introduction

Distributed Computing

Distributed computing is a forty year old field of science that primarily aims to improve scalability and redundancy in computer systems. Scalability has received the bulk of the attention in this field because of the applications to computationally intensive projects. This has led scalability to be extensively explored in both academic and practical settings. On the other hand, until recently, much of the focus of redundancy has been academic. The goal of redundancy is to eliminate single points of failure by replicating the same computation across multiple different machines. In (Lamport et al. 1982), the authors introduce the allegory of a group of Byzantine Generals' attempt to achieve consensus on when to attack a city. In doing so, the authors outlined the fundamental problem faced by redundancy and formalized the study of consensus: the ability for distributed systems to maintain reliability in the face of arbitrary failures and adversarial behavior. For many years the study of consensus stayed within the context of closed, “permissioned”, networks. The number of nodes, their identities, and communications are known by all participants in these networks. To join the network, nodes must first ask for permission before they are allowed to participate in consensus. Permissioned networks have many practical applications such as businesses that want redundancy for sensitive information. They are also relatively easy to set up, maintain, and easy to coordinate because there is a high degree of certainty about the participants in the network. The type of consensus protocols that developed for these networks are commonly known as classical or Byzantine Fault Tolerant (BFT) protocols, characterized by quorum based consensus. BFT consensus protocols have since been adopted to work in permissionless settings (Kwon, 2014) but still operate under many of the same assumptions and limitations. One such limitation is the quadratic communication overhead. This overhead makes BFT protocols unfavorable for large-scale consensus as linear node participation scales with superlinear performance degradation.

In 2008 a pseudonymous entity by the name of Satoshi Nakamoto introduced a new type of consensus, now known as chain-based consensus. As opposed to BFT consensus, chain-based consensus allows node participation to scale linearly with sublinear performance degradation. This breakthrough was made possible by relaxing assumptions BFT consensus protocols made about voting finality and network communication between nodes. While BFT consensus has historically been extremely limited in scope due to scaling issues, chain-based consensus opened the door for large-scale, permissionless, peer to peer redundancy in computer networks.

This allowed for the creation of extremely large and fault tolerant networks known as blockchains. Equally important to Nakamoto's new type of protocol, was the idea of using consensus to distribute parts of the *ownership* of a database instead of simply distributing parts of the computer processes. The idea of a community owned data system has various implications, including the creation of a trillion-dollar cryptocurrency industry.

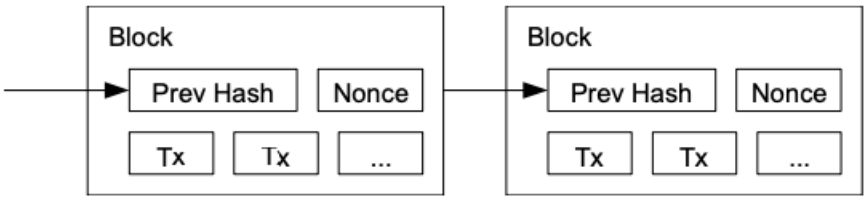
Blockchains

Blockchain is an all-encompassing term that refers to a specific type of distributed system and data structure. The first blockchain is largely agreed to be Bitcoin.

From a distributed systems perspective, a blockchain is a type of protocol that allows multiple parties to form consensus on the ordering and inclusion of transactions. A major part of the protocol is the type of consensus protocol used to form agreement between nodes, such as the one invented by Satoshi Nakamoto. However there are many other important parts of a blockchain's protocol including resource pricing and the peer-to-peer (p2p) networking layer. A blockchain network consists of many different people running clients of the same protocol on their computers. There are often many different types of clients as there are advantages and disadvantages for implementing a protocol in one programming language over another. The strength of the network comes from the fact that if a node violates the protocol, other nodes will kick them off the network by dropping them as a peer. The reason for protocolizing a blockchain as opposed to everyone running the same client is that it helps decentralize the development of the network.

The term "blockchain" originates from the type of data structure used to maintain consensus of transactions. When transactions are submitted to the network, nodes bundle them together and form consensus on the ordering in batches. These batches are referred to as blocks, and since the history of the network is essentially a long chain of blocks, the network became known as a "blockchain".

Figure 1: Blockchain Diagram



Source: https://bitcoin.org/bitcoin.pdf

In this image are two blocks, each containing a list of transactions (tx), the cryptographic hash of the data in the previous block (prev hash), and a number nodes must solve for when forming consensus (nonce). It is important to note that the blockchain data structure only gives nodes the history of transactions. The current state of the blockchain (ie: the current balance of each account), is maintained in a different data structure known as a Merkle tree, and is computed locally by replaying the ordering of transactions from the very first block. This is why it is important to distinguish that blockchain protocols are only used for the ordering and inclusion of transactions.

Although blockchain is an all-encompassing term, it is important to recognize that it originated from the data structure used to keep track of consensus. The Bitcoin whitepaper does not call the network a “blockchain”, but rather a “distributed timestamp server”. This definition is more in line with the type of distributed system a blockchain is, however the data structure definition has become more popular despite the fact that it reveals very little about the actual network.

Economics & Distributed Computing

Economics plays many roles in the design of blockchains and permissionless distributed systems in general, which is why it is a good topic for my major oriented thesis. Game theory is central to the incentive structures of blockchains. The primary goal of any blockchain is to incentivize nodes to act honestly under any circumstance. Any collusion or deviation from consensus rules could be disastrous for the network and the applications built on top of it. These incentive structures are commonly known as cryptoeconomics. By joining the best security properties of cryptography with the best economic incentives, it is believed that nodes can be prevented from acting maliciously. Fundamental to cryptoeconomics, is how to make the incentive for consensus participation, typically a native digital currency, as attractive as possible. This generally boils down to balancing the issuance of the currency against a

mechanism that removes it from circulation. Although out of scope for this paper, it is an inherently economic question and is a central theme to the sustainability of a blockchain.

Majority Attacks

Blockchains have a very large attack surface. Outside of attacks related to users accounts or implementation bugs, there are a number of ways to compromise a blockchain by directly attacking the consensus protocol. The most well known of these is the majority attack. Although there are various types of nodes in a blockchain network, only two are first-class citizens and have authority over the network.

The first is the blockproducer, sometimes referred to as a miner or validator. These nodes act as the "server" of the network and are responsible and rewarded for publishing updates to the network. However, there is a high economic cost to operating this type of node. To prevent spam and other sybil attacks, blockproducers only receive voting power over what updates are published if they have committed resources towards the network. Ideally each human would get one vote, but because we can't authenticate whether a single person is using a single computer, consensus must happen in the resource paradigm. In proof-of-work blockchains, the original version of the resource paradigm, voting power is determined by your proportion of allocated computing power relative to the rest of the blockproducers. In proof-of-stake blockchains, a newer version of the resource paradigm, voting power is determined by your proportion of stake relative to the rest of the blockproducers. Stake is essentially collateral that a blockproducer escrows to the network.

The second type of node is the full node. These nodes act as the "clients" of the network and have lower hardware requirements so that the average person can operate them. Although operators are not explicitly rewarded for running a full node, it is the only way to access the network as a first-class citizen without running a block producing node. The reason that this is important, is that if you access the network through a trusted third party's node, you lose the security benefits that make blockchains different from other networks in the first place. Once the blockproducers come to consensus and publish an update, every full node downloads the block and makes sure that the update is a valid state transition. Validity is proven by re-executing the transactions in the order they were given and checking that the protocol rules were followed. If the block does not follow protocol rules and is an invalid state transition, full

nodes will reject it and revert to the last valid block. In proof-of-stake protocols, blockproducers are explicitly punished for publishing an invalid block by slashing their stake. In proof-of-work protocols blockproducers are implicitly punished by incurring high computational costs without any reward. Protocol rules enforce many various things including the payment of fees per unit of compute, that transactions were signed correctly, and the issuance schedule of the native currency was followed correctly. In doing so, full nodes verify that blockproducers are following the protocol that the community has agreed upon. This gives clients of blockchain networks much more power than in traditional client server models.

A majority attack is an attempt to gain control over a byzantine threshold of the total blockproducers' voting power to disrupt the safety or liveness of a blockchain. Safety is the guarantee that no malicious update has been committed to the network and liveness is the guarantee that the network will not stop. In blockchains with a synchrony assumption, this threshold is 50%. In blockchains with a partial synchrony assumption, this threshold is 33% for liveness and 66% for safety. Majority attacks can be executed by colluding with other blockproducers or purchasing more of the resource that grants you voting power. The effect of a majority attack depends on the type of consensus protocol being used, however full nodes will always prevent consensus rules from being changed. Nevertheless, majority attacks can still result in censorship of specific transactions or users, the blockchain halting on chains with partially synchronous BFT consensus, and double spends on chains with chain-based consensus protocols. This is due to the fact that these protocols rely on probabilistic transaction finality.

Research Exploration

In this thesis I explore how certain design choices impact a blockchain's resistance to a majority attack. To do so, it is important to gain a deep understanding of how blockchains, and their consensus protocols work. The literature review is a summary of my research into the history of blockchain development.

Definitions and Abbreviations

1. **Asynchrony:** A communication model that assumes nothing about network delays or time bounds. In this model consensus rounds don't progress until nodes have responses from all other nodes
2. **Blockchain:** A type of distributed network that forms consensus on the ordering and inclusion of transactions
3. **Blockproducers:** The nodes that form consensus on what updates to push to the network. Sometimes referred to as "miners" or "validators" depending on whether the network is proof-of-work or proof-of-stake based
4. **Blocks:** A bundle of transactions submitted to the network
5. **Blocksize:** The size of updates made by blockproducers
6. **Blocktime:** The amount of time that passes between updates made by blockproducers
7. **Byzantine faults:** a computer fault that presents different symptoms to different observers
8. **Byzantine threshold:** The proportion of voting power in a consensus protocol that once passed, allows an adversary to execute a majority attack
9. **Consensus:** the study of achieving overall system reliability in the presence of faulty processes
10. **Full nodes:** The type of node that a typical person will use to access the network as a client
11. **Liveness:** The guarantee that a distributed system will not halt
12. **Majority attack:** An attack in which adversaries target the consensus protocol to disrupt the safety or liveness of a blockchain
13. **Nakamoto Coefficient:** The number of block producing nodes necessary to collude and execute a Majority attack
14. **Nodes:** The actors within a blockchain network
15. **Partial synchrony:** A communication model that operates under an asynchrony assumption until a network timeout and then operates under the assumption that a non-response is a byzantine fault
16. **Permissioned network:** A network in which nodes must ask and receive permission from existing nodes to join the network
17. **Permissionless network:** A network in which nodes can freely join and leave a network at anytime
18. **Safety:** The guarantee that nothing bad has happened in a distributed system
19. **Synchrony:** A communication model that assumes all nodes will have complete information about the network by Δ. In this model consensus rounds will always progress once Δ has passed as a non-response results in the assumption that the node has dropped from consensus participation

Literature Review

The Origins of Blockchains

Blockchains spun out of a political effort known as the crypto wars. Originally starting in the 1970s, the crypto wars were an effort by the public sector to democratize access to cryptography. Before this effort, cryptography was a technology tightly guarded by various governments and militaries around the world. This movement was largely started when (Diffie & Hellman, 1976) was published. The paper fittingly opens with the statement "We stand today on the brink of a revolution in cryptography" and closes with "We hope this will inspire others to work on this fascinating area in which participation has been discouraged in the recent past by a nearly total government monopoly". Diffie and Hellman not only kicked off an academic movement, but also a political one. Many began to realize that cryptography was a way to maintain privacy rights and individual freedoms in the information age. This realization elevated the crypto wars from a scholarly subject to a fight for democracy. The crypto wars had many successes including a series of legal battles that brought cryptography and code under the protection of the First Amendment. The crypto wars were also foundational in paving the way for online privacy rights in an era of big data and surveillance.

The Cypherpunks, the proponents of the crypto wars and the individuals that carried on the torch from Diffie and Hellman, realized that despite all of their successes they still had a huge problem. Cryptography did not guarantee privacy by itself. Traffic and metadata information could reveal almost as much information as unencrypted data. Furthermore, cryptography did not inherently have any censorship resistant properties. This problem led to the creation of platforms such as TOR, Pirate Bay, and ultimately blockchains, which aim to complement the privacy properties of cryptography with censorship resistant guarantees. Satoshi's comment that "Bitcoin is an implementation of Wei Dai's b-money proposal on Cypherpunks" (Benjyz, 2014) indicates that the original intention of blockchains was to solve the censorship problem of cryptography.

Timeline of Consensus Protocols

1980-2008

Academic BFT research laid the groundwork for consensus in the face of Byzantine behavior. Seminal papers such as (Leslie Lamport et al. 1982) and (Michael J. Fischer et al. 1986) helped formalize the area of study. Further work such as (Cynthia Dwork et al. 1988), (Miguel Castro et al. 1999), and (Hagit Attiya et al. 1994), helped tackle existing problems. Although these protocols were studied under many constraints, such as permissioned and low participation settings, they undoubtedly blazed the path for blockchains.

2008-2014

(Nakamoto, 2008) built upon much of the work done by BFT research, but adopted consensus for the permissionless setting. Although assumptions about network communication and finality had to be relaxed, Satoshi showed that consensus protocols could scale to the global level. The introduction of chain based protocols ignited a renewed interest in consensus, and led to hundreds of new attempts at consensus protocols, each with different focuses and tradeoffs. (Buterin, 2014) demonstrated that virtual machines, complete with their own Turing-complete programming language, could be built on top of consensus protocols. While state machine replication was not a new concept in consensus, previously, it was not believed possible that complex state transitions could be done on permissionless consensus. This was made possible by creating a blockchain-native virtual instruction set that operates as a common CPU for all nodes to use. This environment is known as the Ethereum Virtual Machine, and was particularly important because it introduced the concept of making instructions resource limited through the use of a gas meter. Every operation code for each instruction has its own price based on the usage of the physical resources of the actual computer running a node. Resource pricing is defined in the yellow paper, the protocol's formal specification. Resource pricing and opcodes are often updated over time to better balance node expenses from running the virtual machine's state transitions on their own physical hardware.

The synchrony assumption of chain-based consensus previously forced blockchains to have enormously slow blocktimes, such as bitcoin's 10 minutes, or else safety could be disrupted. (Sompolinsky & Zohar, 2013) demonstrated that slow blocktimes could be worked around. By counting stale and orphaned blocks as part of the chain's weight, nodes could mine asynchronously at times with only a minimal trade off in security. Many blockchains such as Ethereum have adapted this fork choice rule in order to lower blocktimes to a couple seconds.

2014-2017

(Kwon, 2014) was the next big innovation in consensus as it introduced secure proof-of-stake by using a BFT protocol. Prior to Tendermint, it was believed that proof-of-stake was not possible due to the long range attack, also known as costless simulation. This problem is inherent to the fact that proof-of-stake relies on a resource that is tracked by the very database that an adversary is trying to manipulate. Proof-of-work protocols do not have this problem as the resource is computing power which is external to the network. Although this solved the core proof-of-stake issue, many believed that it was actually a step back in permissionless consensus. Tendermint's reliance on BFT consensus meant that it faced many of the scaling issues that had originally led to Satoshi's chain based protocol. Tendermint eventually circumvented this problem by taking advantage of its consensus protocol's fast transaction finality to enable secure communication between tendermint based chains and scale horizontally. Most of the contention against this scaling approach comes from the fact that it is very hard to run full nodes for multiple chains as the network of blockchains continues to grow. The need for chain based proof-of-stake wasn't solved until two years later when (Kiayias et al. 2016) was published. Ten days later, another chain based proof-of-stake protocol (Daian et al. 2016) was published. Both have different approaches to the long range attack with Ouroboros using a preassigned slot leader and Snow White using checkpointing.

2017-Present

A number of engineers have begun combining different protocols to achieve the best properties of both. (Buterin et al. 2020) overlays a chain based protocol with a BFT based finality protocol and is being used in the current Ethereum protocol. More recently, there has been a focus on the use of directed acyclic graph (DAG) based consensus in BFT protocols in papers such as (Danezis et al. 2021). Although DAGs have been used in GHOST protocols to help calculate uncle rewards, the implementation into BFT protocols for transaction ordering is novel. This approach removes the communication overhead that plagues BFT protocols, because it allows nodes to interpret transaction order locally which separatees consensus logic from the peer-to-peer (p2p) networking layer. Rather than sending a confirmation vote for each consensus round, each node has a local view of the graph in which they can interpret proposals and votes without additional communication.

Blockchain consensus has begun to especially diverge from classical consensus within the past few years due to the emphasis on scaling. One such example is the idea of modular blockchains. Traditionally, each machine in the network participates in state machine replication. Every node receives the same input, executes the input, receives an output, and then compares the results to other nodes where consensus on the correct output is eventually formed. Modular blockchains offload parts of state machine replication to different networks. This is only possible because the network where consensus is formed and state is settled, is able to check that the other networks actually performed the work they were supposed to. (Ben-Sasson et al. 2018) demonstrates how execution of state transitions can be outsourced to untrusted third parties. This is done by the untrusted third party encoding the execution trace of transaction execution into a polynomial, and then verifying that the third party knows the polynomial. Verification of the polynomial is succinct because if the polynomial is of degree n, we only need $n + 1$ unique points to identify it. This allows the settlement network to focus on consensus networking and the availability of transaction data, while execution can be outsourced to a single powerful computer without raising hardware requirements for the rest of the nodes. A bottleneck for many blockchains is data availability, the need for transaction data to be available at all times in case a node falls out of sync with the network and needs to build the current state themselves. Traditionally, consensus requires each node to store all of the state itself. Since blockchains rely on decentralization for security, forcing all nodes to store an ever increasing amount of state can be extremely harmful. Eventually people may get priced out of the network as they are forced to buy more storage and compute to keep up with state bloat. (Mustafa Al-Bassam et al. 2018) introduced a solution to this. The data needed to reconstruct state is not held by nodes on the consensus network, but on a separate network in which it is easily verifiable that data is not being withheld or manipulated by probabilistically sampling the data. This is possible by a technique called data availability sampling which is core to many of the modular blockchains designs and is a proposal for Ethereum's scaling plans.

Evaluating Consensus Protocols

Consensus protocols are often rigorously evaluated for fault tolerance with formal security proofs. This is done by reducing a protocol into a mathematical model to prove correctness of intended properties. Although many protocols such as Satoshi's Bitcoin are introduced without any formal proof, distributed systems academics and engineers take it upon themselves to try to prove security or the absence of it. Bitcoin was not given a generic security proof until (Juan A. Gray et al. 2014). Most recently was (Francesco D'Amato et al. 2022) which argued that

Ethereum's current consensus protocol needed to be changed because of how its complexity hinders formal security analysis.

Despite the abundance of rigorous protocol evaluation, there is a lack of comparisons of fault tolerance between protocols themselves. Distinguishing between correlation and causation makes qualitative measurements of fault tolerance extremely difficult. (Bulat Nasrulin et al. 2022) compares different protocols by benchmarking transaction throughput, latency, and CPU usage however it is not comprehensive because it doesn't factor in many key components such as state bloat, minimum hardware requirements for nodes, or how fault tolerance is optimized differently in each protocol. Other papers such as (Gengrui Zhang et al. 2022) closely examine message complexity, however are not immediately applicable to permissionless blockchains because they assume nodes share equal voting power. Similarly, (Fan-Qi Ma et al. 2021) addresses in what situations stochastic performance can destabilize consensus, however does not factor in that in practice nodes have varying amounts of voting power. (Lewis-Pye & Roughgarden, 2021) explores fundamental differences in permissionless consensus protocols such as synchronicity, sybil resistance mechanisms, and impossibility theorems. However, it does not examine protocols in production and examine how stake is distributed amongst nodes. (Evan Sultanik et al. 22) explores fault tolerance extensively by not only looking at stake distribution, but also properties exogenous to the protocol itself such as network delay, network topology, and ISP reliance. Although this was a thorough dive into networking centralities, it did not examine the protocols themselves but rather the infrastructure they rely on. Factoring this into a comparison of the protocols themselves would provide the greatest indication of fault tolerance.

A couple metrics are commonly used to measure stake distribution including Nakamoto coefficients and Gini coefficients. Nakamoto coefficient was a metric introduced by former Coinbase CTO, Balaji Srinivasan. It represents the minimum number of entities needed to compromise a subsystem of blockchains. Nakamoto coefficient is typically used in the context of stake distribution, in which case the Nakamoto coefficient is equal to the minimum number of entities needed to reach a Byzantine threshold. However, Nakamoto coefficients can also be used in reference to client implementations, developers, nodes, exchanges, or others. It was loosely based off of the Gini coefficient and Lorenz curve which model the dispersion of income inequality.

Summary

The research process of this thesis demonstrated just how far blockchains have developed since the original Bitcoin whitepaper. However, many of these developments are not reducible into metrics or quantitative statements that would make comparison practical. To continue investigating the original question of whether certain design choices of blockchains are correlated with higher resistance to majority attacks, I must compare blockchains that are easily comparable. As such, the experimental portion of this thesis focuses on consensus parameters rather than different consensus protocols.

Regression Modeling

OLS Model

My primary question is "what is the relationship between blockchain consensus parameters and network resistance to majority attacks?". This question can be broken down into independent variables (blockchain consensus parameters) and a dependent variable (Nakamoto coefficient). In econometrics, Ordinary Least Squares (OLS) regressions are a common method for estimating the correlation of independent variables with a dependent variable.

OLS regression models adhere to the format:

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 \ldots + U_i$$

- Y_i = the dependent variable being measured
- β_0 = the intercept (constant)
- β_1 = the slope coefficient for variable "1"
- X_1 = independent variable "1"
- U_i = the error (disturbance) term that represents all other factors that affect Y_i

OLS models assign betas to each of the independent variables by minimizing the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n} (Y_i - \hat{y}_i)^2$$

To obtain a unique solution to this optimization problem, OLS models must have more observations than independent variables. Furthermore, OLS models must restrict the independent variables to as few as possible because of a phenomena known as overfitting. Fitting extremely complex models causes machine learning models to learn the random noise in the data instead of the relationship we are trying to estimate. This causes problems with causal inference and using the model to predict outside of the training sample. Due to these problems, it is important that my model remains simple with respect to the amount of observations I can collect.

OLS models are useful because they demonstrate that a one unit change in X_i is associated with a β_i unit increase in Y_i. Under strong assumptions about the error term, β_i is the effect of X_i on

Y_i. OLS models explicitly assume that the data modeled has a linear relationship. Although this is a strong assumption, OLS is an extremely performant and flexible model because nonlinear data can be represented through transformations and basis expansions. The flexibility of the model is shown by its widespread use in academic and industry settings of statistical learning. In prediction settings, OLS will often outperform more complex algorithms in accuracy and training time. However, my decision for using OLS is because the linear assumption makes interpretation of the betas straight forward. This is important as my research question is looking at relationships within the data with the sole purpose of causal inference.

Data Collection

Although blockchains have been around since 2009, the population sample of blockchains that I can collect data for is extremely small. This is because many blockchains fail to sufficiently decentralize protocol development, and so when projects run out of runway they are abandoned. As a result, there is a limited amount of blockchains that are running and have organic user activity. Additionally, I could not collect data on proof-of-work blockchains for the first research question, because it is extremely difficult to get reliable numbers on voting power distribution due to mining pools. I ended up collecting data on 18 blockchains primarily from the Cosmos ecosystem. Although it would have been preferable to run full nodes and query parameters myself, this requires extensive time and technical expertise. As a result, my data was drawn from Mintscan Block Explorer and Coingecko. I compiled this data into a CSV file using the software ModernCSV. The data frame is attached in the appendix at the end.

Because of the results from the first research question, I collected and investigated additional data drawn from Etherscan.io. All of this data was from the Ethereum blockchain. In addition to this, I used QuickLatex to render my exported tables.

Research Question #1: Nakamoto Coefficient and Consensus Parameters

Hypothesis

I am exploring the research question "what is the relationship between blockchain consensus parameters and network resistance to majority attacks?". In the literature review it was suggested that lowering the barrier of entry to network participants is extremely important for decentralization. In this research, I am investigating what design choices blockchain engineers should make to maximize decentralization. My hypothesis is that Nakamoto coefficients will negatively correlate with higher hardware requirements. Simply stated, blockchains with high barriers of entry will have the highest levels of voting power centralization.

H_0 (Null hypothesis): No statistically significant relationships between Consensus Parameters and Nakamoto Coefficients

H_A (Alternative hypothesis): Statistically significant relationships between Consensus Parameters and Nakamoto Coefficients

OLS Model and Rational

$$Nakamoto\_Coefficient_i = \beta_0 + \beta_1 Block\_time_1 + \beta_2 Block\_size_2 + \beta_3 num\_validators_3 + \beta_4 network\_age_4 + \beta_5 market\_cap_6 + U_i$$

Nakamoto coefficient is the number of malicious entities required to reach a Byzantine threshold. In partially synchronous protocols, this is the number of malicious entities required to reach ⅓ of total stake for liveness and ⅔ of total stake for safety. For the purpose of this research, we will only focus on the liveness threshold as all data was drawn from partially synchronous BFT protocols.

Blocktime is the amount of time that passes in between state transitions. Because nodes must check for transaction validity, and then sum the new inputs to the historical state in order to compute the current state, the faster the blocktime the more computationally expensive the state transition becomes as the node has to do the same amount of work in less time. If the blocktime is too fast, nodes can be priced out of the network which can lead to centralization.

Blocksize is the amount of data contained in each block. This places a similar constraint on nodes as the more megabytes the block is, the more bandwidth a node requires to download it. In between blocktimes, nodes must fully download the proposed block, verify the state transition, and then broadcast its vote before the consensus round ends. This places minimum hardware and internet requirements on nodes which can cause some to become priced out of the network.

Number of validators represents the protocol's hard limit of how many nodes can participate in consensus. This is also known as the “active validator set”. Although some protocols such as Ethereum do not have a limit, most BFT protocols such as those in the Cosmos ecosystem do. This is to limit the communication complexity during consensus rounds which helps keep networks performant. However, it also places a hard limit on the potential for voting power distribution.

Network age must be controlled for because the longer it has been around, the greater the chance that more people know about it and would run nodes. As such, it would be unfair to compare a brand new blockchain to one that has been around for many years.

Market cap is important to control for as a more popular coin will reach a wider audience than a lesser known coin. As such, the population that would consider running a node for it will be larger which may affect stake distribution.

Running the Model: RQ #1

Before running the model, I plotted the Nakamoto coefficient against the various independent variables to see if any trends emerged. Below are the graphs for each of the variables:

Figure 2: Nakamoto Coefficient vs. Blocktime

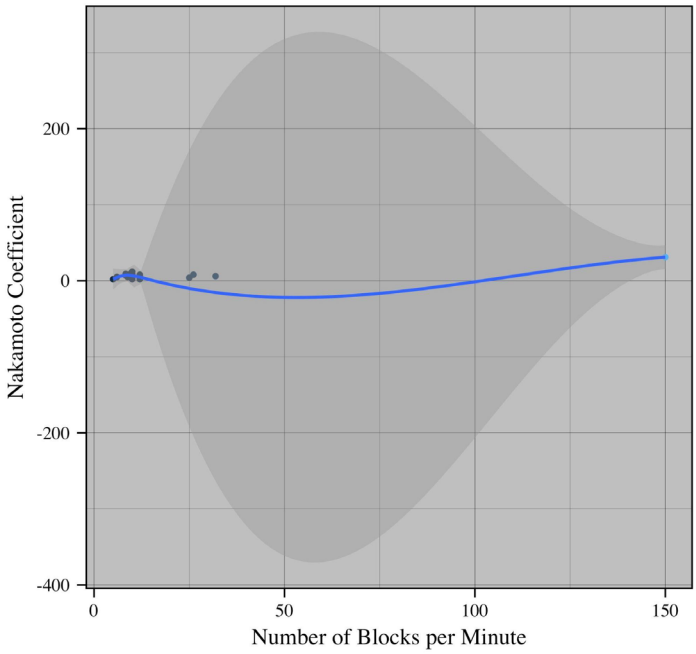


Figure 3: Nakamoto Coefficient vs. Blocksize

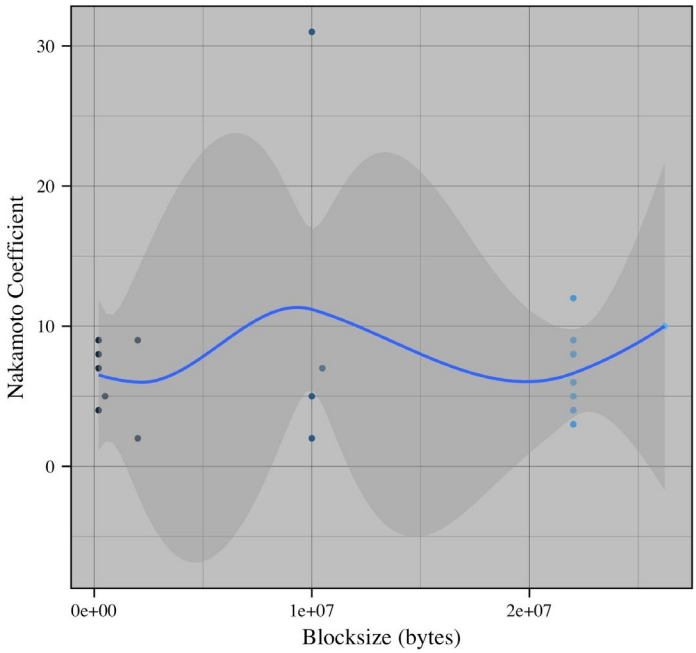


Figure 4: Nakamoto Coefficient vs. Active Validators

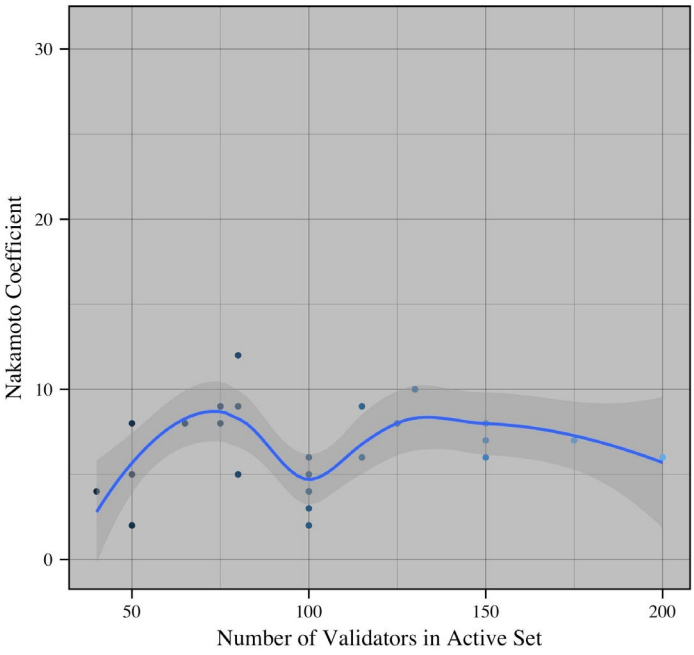


Figure 5: Nakamoto Coefficient vs. Network Age

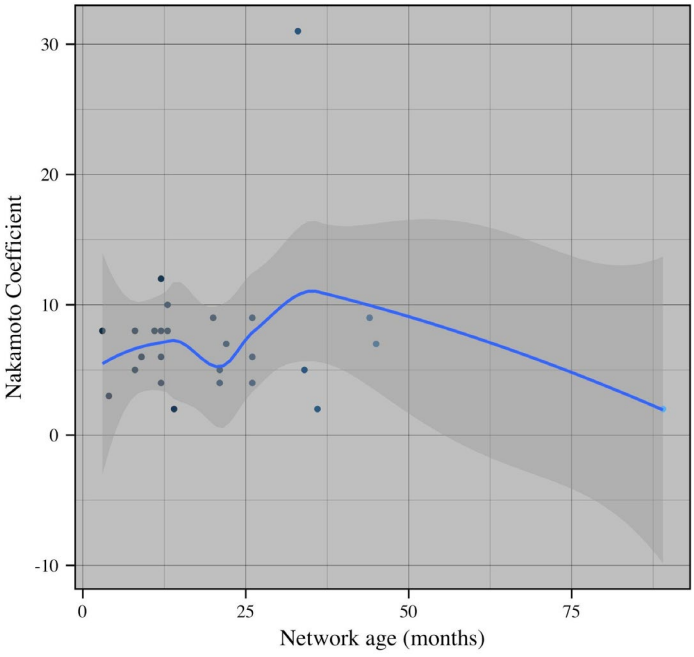


Figure 6: Nakamoto Coefficient vs. MCAP

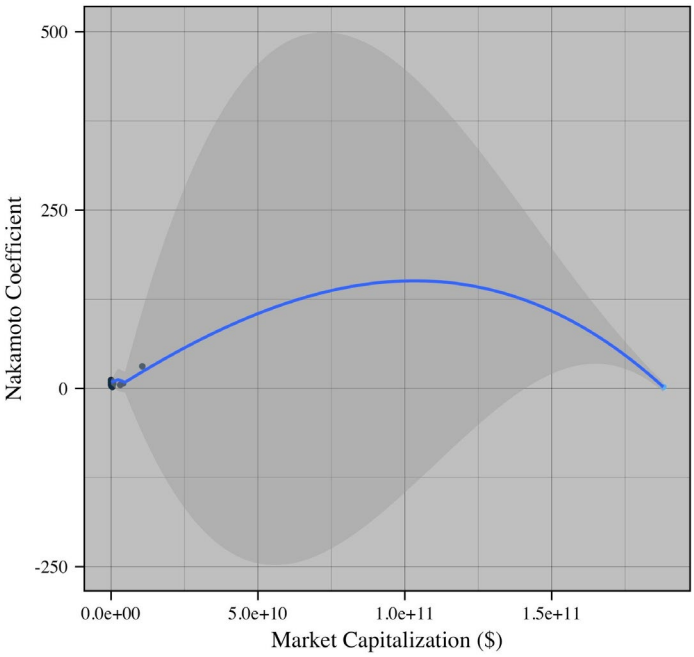
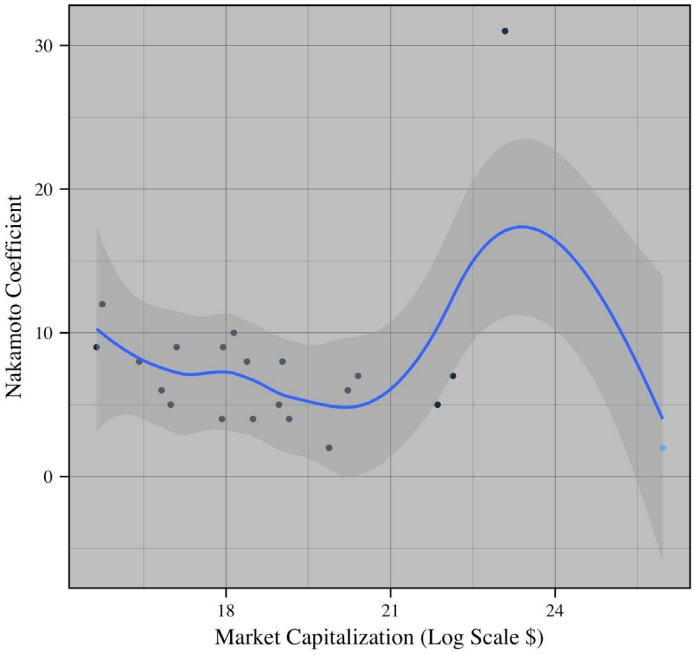


Figure 7: Nakamoto Coefficient vs. Logarithmic MCAP



The data was smoothed using the geom_smooth() function via a generalized additive model. The gray band is the 95% confidence level interval for prediction from the model. Given the presence of overplotting, this smoothing helps us identify possible trends, however does not guarantee that they exist. Because of the large differences in market capitalization of the blockchains in the dataset, an additional graph was added which does a logarithmic transformation of market capitalization. This makes it easier to examine the variation between market capitalization and Nakamoto coefficient.

These graphs show no clear, linear trend between the independent variables and Nakamoto coefficient. Part of the difficulty of this experiment is the low variation in the population sample. Drawing conclusions from 18 samples is an extremely small sample for this experiment which is further emphasized by the huge confidence level bands in the graphs. However, given the large number of factors that influence Nakamoto coefficient, it is possible that trends are not identifiable because the other variables are not controlled for.

When running the OLS model developed previously, we get the following estimates:

Figure 8: OLS Model

	Dependent variable:
	nakamoto
blocktime	−0.112
	(0.098)
blocksize	0.00000
	(0.00000)
validators	0.016
	(0.014)
age	−0.042
	(0.073)
mc	−0.000
	(0.000)
Constant	7.483***
	(2.586)
Observations	18
R^2	0.160
Adjusted R^2	−0.190
Residual Std. Error	2.510 (df = 12)
F Statistic	0.457 (df = 5; 12)
Note:	*p<0.1; **p<0.05; ***p<0.01

Figure 9: Log Transformed OLS Model

	Dependent variable:
	nakamoto
log(blocktime)	-0.741
	(1.487)
log(blocksize)	-0.103
	(0.305)
log(validators)	3.195^{***}
	(1.041)
log(age)	-1.040
	(0.789)
log(mc)	-0.763^{**}
	(0.308)
Constant	12.828^{*}
	(7.312)
Observations	18
R^2	0.416
Adjusted R^2	0.172
Residual Std. Error	2.094 (df = 12)
F Statistic	1.706 (df = 5; 12)
Note:	*p<0.1; **p<0.05; ***p<0.01

In the first OLS model none of the coefficients are statistically significant except for the constant. The R squared is 0.160 which is low, but not surprising, given the large amount of factors that go into a blockchains Nakamoto coefficient. Other variables that could play a role include how well documented the node operating process is, how much the network is subsidizing blockproducers with inflationary rewards, and how expensive it is to get into the active blockproducer set. As a result it is not surprising that this model would only explain 16% of the variance of the dependent variables. Because of the large variance in the data, I did a linear-logarithmic transformation. This lets us interpret a 1% change in the independent variable as a $\beta/100$ change in Nakamoto coefficient.

In the log transformed model the R squared is 0.416 which is much higher than the non-transformed model. This indicates that some of the relationships between the independent variables and dependent variables may be logarithmic. The variable blocktime has a coefficient of -0.741 which means that a 1% increase in blocktime results in a 0.00741 decrease in Nakamoto Coefficient. Although the estimator is almost negligible and is not statistically significant, it opposes the original hypothesis. This is because the coefficient can be interpreted as an increase in blocktime, and lowering of hardware requirements and the barrier to entry, is correlated with an increase in stake centralization. The variable blocksize has a coefficient of -0.103 and is not statistically significant. Regardless, it can be interpreted as a 1% increase in blocksize being correlated with a 0.00103 decrease in Nakamoto Coefficient. Like the previous variable, this opposes the original hypothesis. The variable network age has a coefficient of -1.04 and is also not statistically significant. Although it is negligible, it opposes our hypothesis as the longer a network has been around, the more people may know about it which increases the potential for stake decentralization. Because these variables are not statistically significant we can not reject the null hypothesis that there is no relationship between blocktime, blocksize, and network age with a blockchain's Nakamoto Coefficient.

On the other hand, the variables validators and market capitalization have statistically significant results. The variable validators has a coefficient of 3.195 and is significant to the 0.01 level. This can be interpreted as a 1% increase in the validator cap being correlated with a 0.03195 increase in Nakamoto Coefficient. Although this is a relatively small factor in the Nakamoto Coefficient, it supports the original hypothesis that lowering the barrier to entry allows stake to decentralize more. The variable market capitalization has a coefficient of -0.763 and is significant to the 0.05 level. This can be interpreted as a 1% increase in the market

capitalization being correlated with a 0.00763 decrease in Nakamoto Coefficient. Like the previous variable, this is a very small correlation. At first it seems that this coefficient opposes the original hypothesis as an increase in market capitalization should bring more attention to the blockchain and thus potential for stake decentralization. However, there is also a well known phenomenon called Maximal Extractable Value (MEV) which occurs in blockchains with high economic activity and creates stake centralization in the nodes with the most voting power. This is because blockproducers can propose blocks to the network that include their own transactions that frontrun transactions from other users. The profit from this frontrunning allows those nodes to buy more voting power, propose blocks more often, and extract more value. Over the long run this compounds stake centralization in the most powerful nodes. Given this relationship it is difficult to tell whether this variable supports or opposes the original hypothesis.

Evaluation: RQ #1

This experiment attempted to answer the question of whether there are relationships between blockchain consensus parameters and Nakamoto Coefficient based on a sample of 18 blockchains. Given that many of the results were not statistically significant, we can not answer this question in the affirmative. As a result, we fail to reject the null hypothesis that there are no statistically significant relationships between consensus parameters and Nakamoto Coefficients. The results also do not support my original hypothesis that Nakamoto Coefficients will negatively correlate with higher hardware requirements. Rather, there is no relationship at all. During the research and data collection process of this experiment I realized that there are an innumerable amount of factors that contribute to Nakamoto Coefficient and stake decentralization that I would simply not be able to account and control for. Ultimately, it was not surprising that I was not able to draw causal relationships from my results, however it was surprising that I was unable to find any strong correlations.

This experiment also provided clarity on what types of relationships OLS regressions could model. In particular, I felt that modeling financially incentivized relationships could be captured by this type of model extremely well as it would be much easier to collect and find data for. In the second research question I examine one such relationship.

Research Question #2: Network Hash Rate and Network Usage

Hypothesis

For the second research question I am exploring the question "what network metrics are most correlated with increasing the incentive to run a block producing node?". The largest reason for running a block producing node is to generate revenue. Given that the market is extremely competitive, and some blockchains have high start-up and fixed costs, the decision to run a block producing node is almost entirely financially motivated. My hypothesis is that the decision to run a block producing node will be primarily based on network activity.

H_0(Null hypothesis): No statistically significant relationships between number of block producing nodes and network activity

H_A(Alternative hypothesis): Statistically significant relationships between number of block producing nodes and network activity

This experiment will be conducted on data solely from the Ethereum blockchain. Ethereum was chosen because there is a large variety of data that can be tested that is widely documented and available. Although Ethereum switched from a proof-of-work based blockchain to a proof-of-stake blockchain in August 2022, we will be looking at historical data from when it was proof-of-work based.

OLS Model and Rational

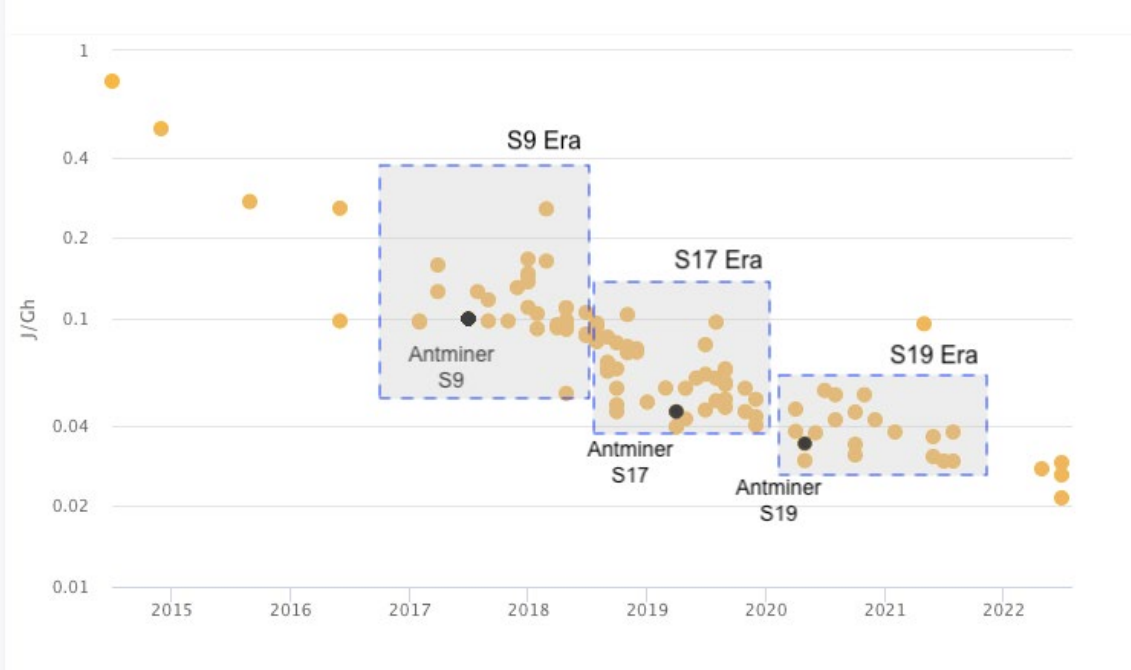
$$PoW\_Difficulty_i = \beta_0 + \beta_1 network\_utilization_1 + \beta_2 daily\_transactions_2 + \beta_3 active\_addresses_3 + \beta_4 market\_capitalization_4$$

In proof-of-work blockchains, voting power is determined by a node's proportion of allocated compute relative to the total allocated compute of all block producing nodes. Using your compute to propose blocks for consensus is known as "mining". To propose a block in the old Ethereum proof-of-work protocol, nodes must incrementally nonce a hashing function until the output is numerically less than the proof-of-work difficulty. The difficulty is a mechanism that helps keep blocktimes relatively constant. If the hash did not change in difficulty, blocktime

would become increasingly faster as more compute joined, which would cause instability and centralization in the network. Ethereum's difficulty mechanism works by making the hash harder to solve if blocks are proposed faster than 12 seconds, and making the hash easier to solve if blocks are proposed slower than 12 seconds.

Because the proof-of-work difficulty is effectively a rate limiting mechanism for how many people are mining the network at any given time, it is a good metric for how much interest there is to participate in consensus. Although we can not track the number of nodes or how much voting power a specific node may have, proof-of-work difficulty is a metric that will indicate the total number of people that think mining is profitable. It is important to note that this is not a perfect metric as there have been drastic improvements in efficiency for the mining process over the years. Originally mining was done on standard desktops, however as mining got more competitive, specialized hardware built for the sole purpose of mining emerged.

Figure 10: Evolution of Bitcoin Mining Efficiency



Source:https://dataalways.substack.com/p/the-merge-and-cryptos-electricity

This figure shows how over time bitcoin mining has gotten more efficient in terms of energy over gigahashes (one billion hashes per second). Although this research question is investigating network data about Ethereum, the evolution of bitcoin mining equipment is a

phenomenon that has also occurred in the Ethereum mining industry. As a result of this, we can not perfectly assume that an increase in difficulty was because more people joined the network. Rather, a more efficient mining rig may have been released. However, for the purposes of this experiment, we will make the assumption that an increase in difficulty represents an increase in the number of people who think mining the blockchain is profitable.

Network utilization is the average gas used over the gas limit per block. Gas is a resource cost for using a specific opcode in the Ethereum Virtual Machine. 1 gas is equivalent to 1 gwei, which itself is equivalent to 10^{-9} ether, the native currency of Ethereum. A full list of the opcodes and corresponding gas costs can be found on the Ethereum Foundation's website. Each block has a gas limit which used to be determined by the miners, however is now hard coded into the protocol itself. The gas limit has increased from 3 million gas in 2015 to 30 million in 2022. Given this, network utilization is effectively a measure of how much demand there is to use the network. Ethereum has a dynamic fee market that increases depending on network utilization.

Daily transactions is another commonly observed metric which measures the total amount of transactions that took place in a given day. A transaction is an action initiated by an externally-owned account (EOA). This is done by calling a function such as transfer(), bundling it into a transaction, signing it with your accounts private key, and then sending it to the network where it is gossiped around until it is included into a block. It is important to note that some transactions are larger and use more gas than others. As a result, daily transactions are not as good of a metric for measuring network usage as network utilization, however it is often commonly cited.

Active addresses is the number of unique addresses that used the network as a sender or receiver on a particular day. This metric is also not perfect because one person may control multiple addresses. A well documented phenomenon known as "wash-trading", where a person sends an asset back and forth between accounts they own, is sometimes done to artificially inflate active addresses or trading volume addresses. Like daily transactions, active addresses is another commonly cited metric however it has its problems.

Market capitalization is a measure of the total supply times the market price of Ethereum. This metric is also commonly used in a node operator's decision on whether to mine or not. To

collect their profits, a miner must sell their rewards over the counter or on the open market. Oftentimes, the market price fluctuates on whether the network is over or undervalued, which itself is represented by market capitalization. Because of this, market capitalization is an important factor for miners to consider.

By investigating these commonly used variables, I hope to see which is most influential on the decision making process to participate in the mining process. I also hope to answer the question, "what network metrics are most correlated with increasing the incentive to run a block producing node?".

Running the Model: RQ #2

Like the previous experiment, I plotted the data before running the model to see if any immediate relationships were apparent. Once again, the data was smoothed using a generalized additive model with the gray band representing the 95% confidence level interval. It is important to note that the smoothing is used to help identify possible trends in the presence of overplotting, however it does not guarantee that any trends actually exist.

Because we are plotting relatively volatile variables against each other, I first plotted the independent variable against time before plotting the independent variable against network difficulty. This helps visualize how the independent variable has changed as Ethereum has matured. Below are the graphs for each of the variables with data from 2015 to 2023:

Figure 11: Network Difficulty vs. Time

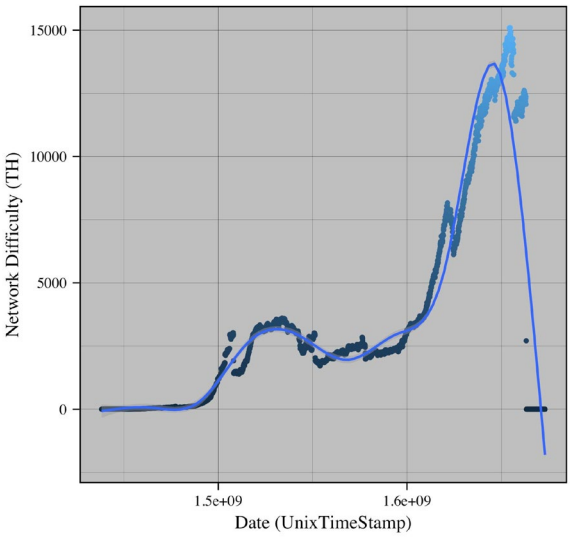


Figure 12: Network Utilization vs. Time

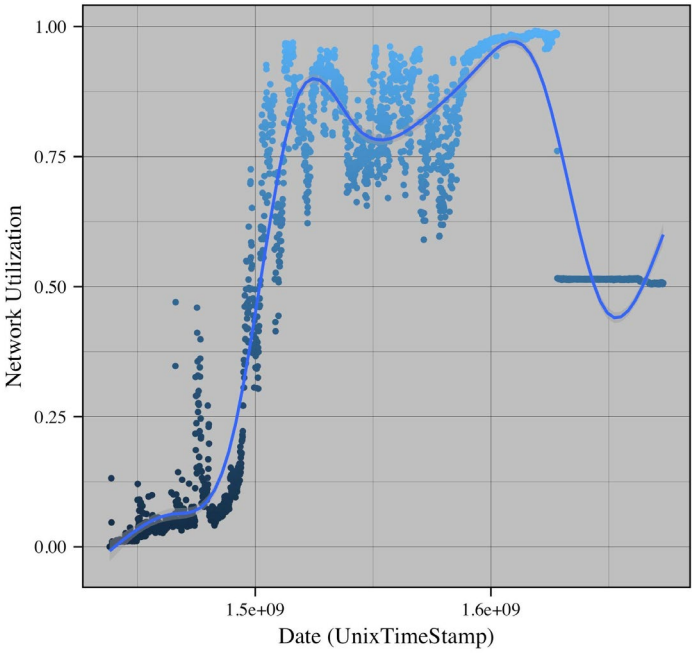


Figure 13: Network Difficulty vs. Utilization

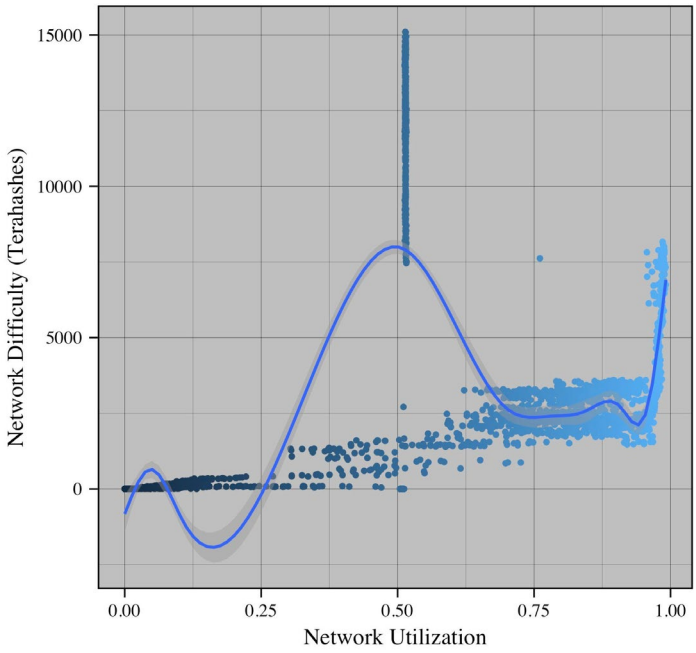


Figure 14: Daily Transactions vs. Time

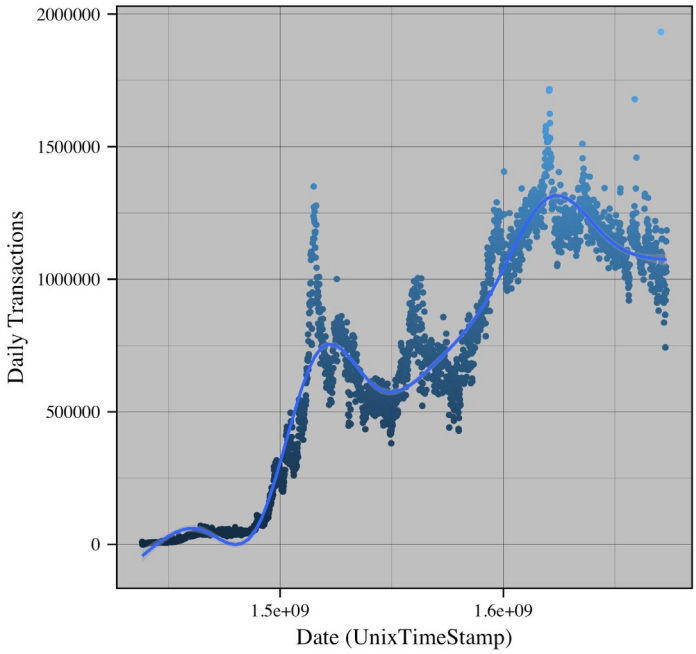


Figure 15: Network Difficulty vs. Daily Transactions

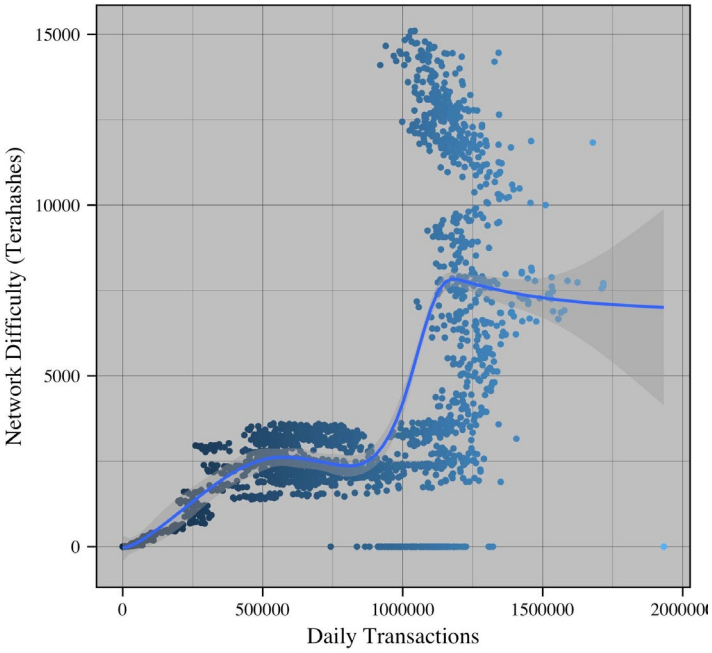


Figure 16: Cumulative Unique Addresses vs. Time

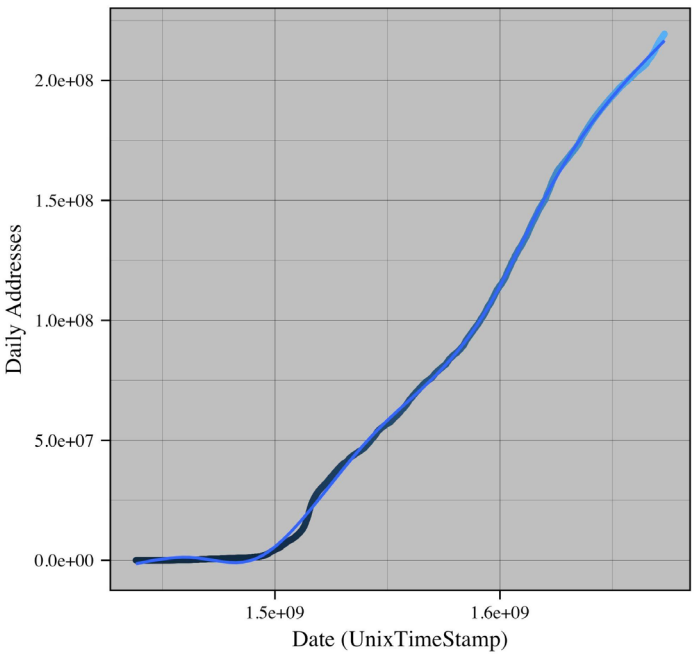


Figure 17: Network Difficulty vs. Cum. Addresses

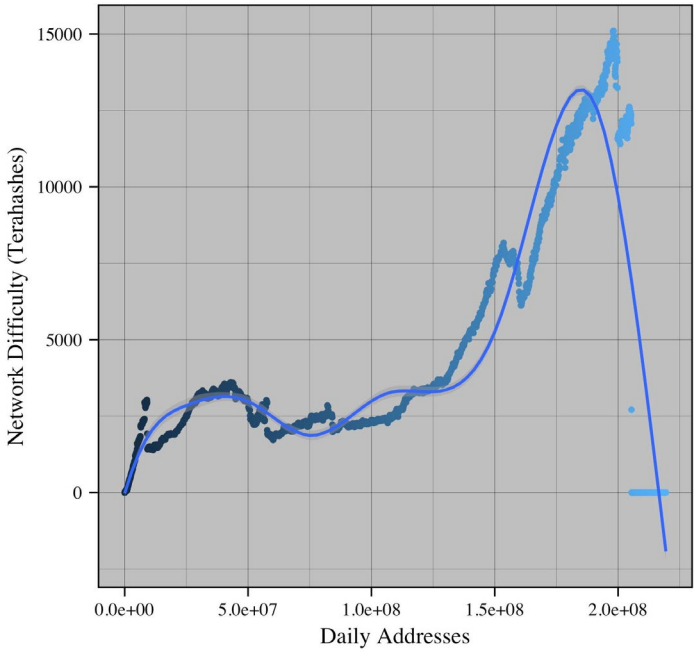


Figure 18: Market Capitalization vs. Time

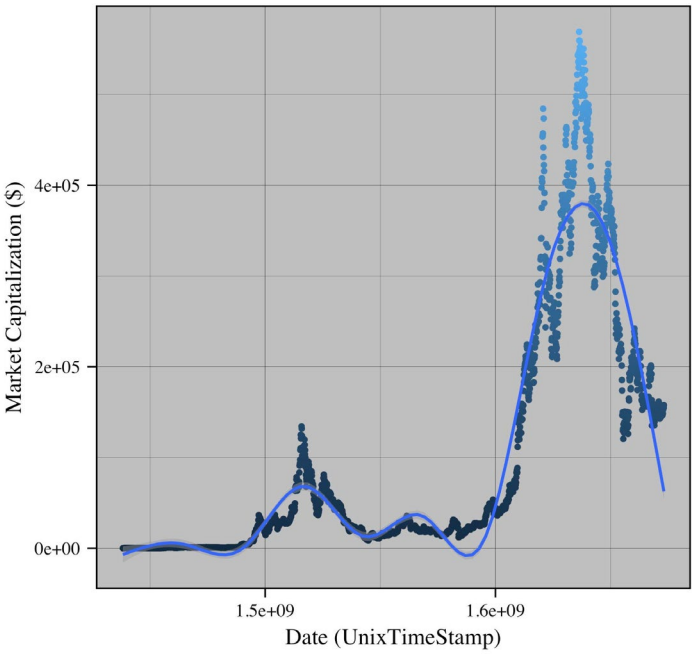
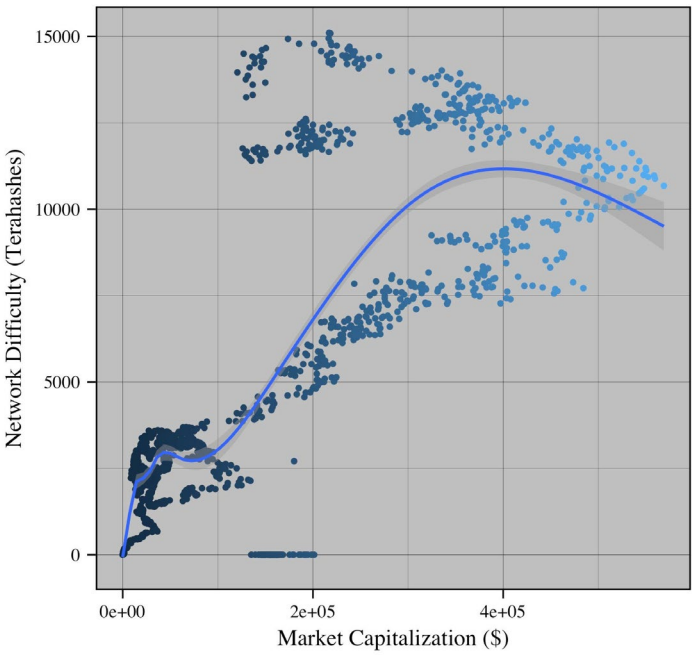


Figure 19: Network Difficulty vs. Market Capitalization



In these graphs a couple interesting things stand out. The first is that in network utilization, 50% has an extremely high frequency. This is because of a change Ethereum made to their fee market known as EIP-1559 (Ethereum Improvement Proposal). This change made the gas-limit and block-size variable instead of constant, and set a target of 15 million gas but a hard limit of 30 million. The purpose of the change was to increase security, by removing denial of service attack vectors, and also make fee markets more predictable through some other changes. However in our graphs, this makes it appear as if the network utilization is only 50% most of the time when in reality, that 50% is 100% of "normal" gas prices before they get very expensive. It will be very important to reflect this in our OLS model so the data is not skewed.

Additionally, in the network difficulty vs. daily addresses chart there is a clear cluster of outliers where the days with the most daily addresses happened on a day with extremely low network difficulty. Similarly, in the network difficulty vs. daily transactions there are a couple clusters where the network difficulty was zero. Both of these factors can be attributed to the fact that in August 2022, Ethereum switched from proof-of-work to proof-of-stake during a hot swap and incredible feat of engineering known as "the Merge". As a result the difficulty went to zero as no one was mining blocks anymore.

Besides these anomalies, daily addresses appear to actually be strongly correlated with network difficulty, not including the data after Ethereum switched to proof-of-stake and difficulty went to zero. Although part of this can be attributed to reverse causality as you must generate an active daily address to participate in the mining process, it is also possible that daily addresses could be a strong factor in the decision to mine in the first place.

Before running the OLS model, I cleaned the data to remove anything after EIP-1559. This also includes all of the data that was biased because of the Merge when proof-of-stake difficulty went to zero. The rationale for doing this is that if post-EIP-1559 data is included, network utilization data would be skewed as there will be a higher frequency of 50% usage than should actually be represented. If post-Merge data is included the effect on the dependent variable (network difficulty) will be heavily skewed as it will be zero despite any changes in the independent variables. As a result, to remove bias from the data, anything after EIP-1559 was deleted.

Figure 20: OLS Model

	Dependent variable:
	network_difficulty
network_utilization	1,922.732***
	(304.414)
daily_transactions	−0.002***
	(0.0004)
daily_addresses	0.00002***
	(0.00000)
market_capitalization	0.020***
	(0.001)
Constant	210.223***
	(34.819)
Observations	2,720
R^2	0.708
Adjusted R^2	0.707
Residual Std. Error	2,134.398 (df = 2715)
F Statistic	1,642.665*** (df = 4; 2715)
Note:	*p<0.1; **p<0.05; ***p<0.01

This model has an R squared of 0.708 meaning that 70.8% of the variation in the dependent variable is explained by the independent variables. All of the coefficients are statistically significant to the 0.01 level. The first variable, network utilization, has a coefficient of 1,922.732. This can be interpreted as a 1% increase in network utilization as having a 1,922.732 increase in network difficulty. The coefficient daily transactions has a coefficient of -0.002 which although statistically significant, is negligible on the overall network difficulty. Similarly, the daily addresses coefficient was 0.00002 and statistically significant, but given the context is negligible. In addition, the market capitalization variable has a coefficient of 0.020 which is negligible. Finally, network difficulty has a constant of 210.223. This means that without any network utilization, daily transactions, daily addresses, and a market capitalization of zero, the network has a difficulty of 210.223 tera hashes per block. Given that an Apple M1 Pro can generate 5 megahashes per second (Lovejoy, 2021) and an Ethereum block is produced about every 12 seconds, a difficulty of 210.223 would be equivalent to 3,504 M1 Pro Macbooks mining the network concurrently.

$$(210.223terahashes * 1000)/(12sec * 5megahashes) = 3503.71\ M1\ Pro\ Macbooks$$

Although mining equipment is much more efficient than an M1 Macbook pro, this comparison still demonstrates how unlikely a network difficulty of 210.223 tera hashes per block would be for a network with zero activity. In turn, this indicates that my model most likely suffers from omitted variable bias and a number of factors that I did not control for. Despite this, if we make strong assumptions about omitted variable bias and causality, the model reflects that the decision to mine Ethereum is largely driven by network utilization.

Evaluation: RQ #2

Although the model does not capture every variable that affects proof-of-work difficulty, it provides strong evidence that mining difficulty is strongly correlated with various network activity metrics. This conclusion follows the intuition that originally led to my hypothesis, the decision to front high startup costs and participate in mining, is driven by key performance indicators. The high statistical significance of the results also reflects that my experiment was well suited for an OLS model unlike the first research question.

In retrospect, it may have been appropriate to conduct a lagged regression. This is because it is likely that some time passes before a node operator observes a key metric and actually begins participating in the network. In addition, it may have been interesting to conduct a difference-in-differences experiment to look at the discontinuities caused by EIP-1559 and the Merge. It would be interesting to look at how fee markets changed before vs. after EIP-1559 or the energy consumption before vs. after the Merge, or even voting power centralization before vs. after the Merge. To conduct the experiment, we would need a counterfactual to observe the treatment effects of each change. For these experiments, since both changes were hard forks, we could observe the forks that did not accept the change. Although this would not be a perfect counterfactual, it may still provide an interesting experiment design. Results from this type of experiment could be used to better inform future protocol changes such as the heavily researched Multidimensional EIP-1559 (Buterin, 2022).

Conclusion

This research explored the relationships between blockchain consensus parameters and network externalities. The results of this experiment yielded a lack of statistical significance, failure to reject the null hypothesis, and an overall conclusion that the model had many omitted variables. Because of the weak conclusions of the first experiment, we conducted a second experiment that better fit within the constraints of OLS and the available data. The direction of the second experiment was largely guided by the fact that causal inference was an impractical task given the immaturity and insufficient population of blockchains available to sample from.

Nevertheless, the work of this research can be extended in multiple unique directions. To hunt for causality, researchers could isolate a specific treatment variable such as a particular feature of a consensus protocol. Researchers could then create two blockchains that are similar in every aspect except for the treatment variable, which would allow for the isolation of the treatment effect of the specific variable. Although this could be done with consensus parameters, this could be generalized back to the original question of differences between consensus protocols themselves. Isolation of the treatment effect could be done via a difference in differences experiment as the blockchain without the treatment effect could serve as an appropriate contrapositive. Because this may take many resources, it may be better to run the difference in differences experiment in a simulation setting by using Monte Carlo methods or imputing data from blockchains that are running in production.

Regardless, there are many different ways for us to expand on our previous work. We hope that the research and topics explored in this thesis will help improve the decentralization and robustness of blockchain networks.

Bibliography

Al-Bassam, Mustafa, Alberto Sonnino, and Vitalik Buterin. 2018. “Fraud and Data Availability Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities.” arXiv. https://arxiv.org/abs/1809.09044.

Attiya, Hagit, Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1994. “Bounds on the Time to Reach Agreement in the Presence of Timing Uncertainty.” https://dl.acm.org/doi/10.1145/174644.174649.

Benjyz. 2014. “satoshi's posts - a selection.” Bitcoin Talk. https://bitcointalk.org/index.php?topic=521880.0.

Ben-Sasson, Eli, Iddo Bentov, Yinon Horesh, and Michael Riabzev. 2018. “Scalable, transparent, and post-quantum secure computational integrity.” Cryptology ePrint Archive. https://eprint.iacr.org/2018/046.pdf.

Buterin, Vitalik. 2014. “A Next-Generation Smart Contract and Decentralized Application Platform.” Ethereum.org. https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf.

Buterin, Vitalik, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X. Zhang. 2020. “Combining GHOST and Casper.” arXiv. https://arxiv.org/abs/2003.03052.

Castro, Miguel, Barbara Liskov, and Operating Systems Design and Implementation. 1999. “Practical Byzantine Fault Tolerance.” https://pmg.csail.mit.edu/papers/osdi99.pdf.

Daian, Phil, Rafael Pass, and Elaine Shi. 2016. “Snow White: Robustly Reconfigurable Consensus and Applications to Provably Secure Proof of Stake.” Cryptology ePrint Archive. https://eprint.iacr.org/2016/919.pdf.

D'Amato, Francesco, Joachim Neu, Ertem N. Tas, and David Tse. 2022. “No More Attacks on Proof-of-Stake Ethereum?” arXiv. https://arxiv.org/abs/2209.03255.

Danezis, George, Eleftherios K. Kogias, Alberto Sonnino, and Alexander Spiegelman. 2021. “Narwhal and Tusk: A DAG-based Mempool and Efficient BFT Consensus.” arXiv. https://arxiv.org/abs/2105.11827.

Diffie, Whitfield, Martin E. Hellman, and IEEE Transactions on Information Theory. 1976. “New Directions in Cryptography.” Stanford Electrical Engineering. https://ee.stanford.edu/~hellman/publications/24.pdf.

Dwork, Cynthia, Nancy Lynch, and Larry Stockmeyer. 1988. “Consensus in the Presence of Partial Synchrony.” https://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf.

Fischer, Michael J., Nancy A. Lynch, Michael Merritt, and Distributed Computing. 1986. “Easy impossibility proofs for distributed consensus problems.” https://groups.csail.mit.edu/tds/papers/Lynch/FischerLynchMerritt-dc.pdf.

Garay, Juan A., and Aggelos Kiayias. 2015. “The Bitcoin Backbone Protocol: Analysis and Applications.” Cryptology ePrint Archive. https://eprint.iacr.org/2014/765.pdf.

Kiayis, Aggelos, Alexander Russell, Bernardo David, and Roman Oliynykov. 2016. “Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol.” Cryptology ePrint Archive. https://eprint.iacr.org/2016/889.pdf.

Kwon, Jae. 2014. “Consensus without Mining.” Tendermint. https://tendermint.com/static/docs/tendermint.pdf.

Lamport, Leslie, Robert Shostak, and Marshall Pease. 1982. “The Byzantine Generals Problem.” *SRI International*. https://lamport.azurewebsites.net/pubs/byz.pdf.

Lewis-Pye, Andrew, and Tim Roughgarden. 2021. “Byzantine Generals in the Permissionless Setting.” arXiv. https://arxiv.org/abs/2101.07095.

Lovejoy, Ben. 2021. “MacBook Pro cryptocurrency mining profitable, but only just.” 9to5Mac. https://9to5mac.com/2021/11/10/m1-pro-macbook-pro-cryptocurrency-mining/.

Ma, Fan-Qi, Quan-Lin Li, Yi-Han Liu, and Yan-Xia Chang. 2021. “Stochastic Performance Modeling for Practical Byzantine Fault Tolerance Consensus in Blockchain.” arXiv. https://arxiv.org/abs/2107.00183.

Nakamoto, Satoshi. 2008. “Bitcoin: A Peer-to-Peer Electronic Cash System.” Bitcoin.org. https://bitcoin.org/bitcoin.pdf.

Nasrulin, Bulat, Martijn De Vos, Georgy Ishmaev, and Johan Pouwelse. 2022. “Gromit: Benchmarking the Performance and Scalability of Blockchain Systems.” arXiv. https://arxiv.org/abs/2208.11254.

Smith, Corwin. 2022. “Opcodes for the EVM.” Ethereum.org. https://ethereum.org/en/developers/docs/evm/opcodes/.

Sompolinsky, Yonatan, and Aviv Zohar. 2013. “Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains.” Cryptology ePrint Archive. https://eprint.iacr.org/2013/881.

Trail of Bits. 2022. “Are Blockchains Decentralized? Unintended Centralities in Distributed Ledgers.” https://assets-global.website-files.com/5fd11235b3950c2c1a3b6df4/62af6c641a672b3329b9a480_Unintended_Centralities_in_Distributed_Ledgers.pdf.

Tripoli. 2022. “The Merge and crypto's electricity consumption - by Tripoli.” Data Always. https://dataalways.substack.com/p/the-merge-and-cryptos-electricity.

Zhang, Gengrui, Fei Pan, Michael Dang'ana, Yunhao Mao, Shashank Motepalli, Shiquan Zhang, and Hans-Arno Jacobsen. 2022. “Reaching Consensus in the Byzantine Empire: A Comprehensive Review of BFT Consensus Algorithms.” arXiv. https://arxiv.org/abs/2204.03181.