Selfish Mining and Networking Effects

Thesis zur Erlangung des Grades
Master of Science (M. Sc.)
im Studiengang Computer Science

Alexander Wagner
wagner.2@campus.tu-berlin.de

Distributed Security Infrastructures
Institut für Softwaretechnik und Theoretische Informatik
Fakultät Elektrotechnik und Informatik
Technische Universität Berlin

Gutachter:
Prof. Dr. Florian Tschorsch
TBA: Second supervisor

eingereicht am: TBA

Hiermit erkläre ich an Eides statt, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein.

Berlin, TBA

. . . .

(Alexander Wagner)

Zusammenfassung

Abstract

Contents

Chapter 1

Introduction

Bitcoin is the most prominent example of a decentralized cryptocurrency [2]. Before the development of Bitcoin a decentralized cryptocurrency had been envisioned for many years. It is a system, where a ledger is kept consistent among multiple parties in a peer-to-peer network without the need of trust. It enables the deployment of electronic cash without a central authority figure like a bank. For this reason it is an enhancement to the currently established electronic banking system.

It is important to keep the ledger consistent and correct. Since there is no central authority, the ledger becomes a consensus problem, which has to be solved in a cooperative, distributed manner. It can also be seen as a Byzantine Agreement problem [6], because multiple indepedent parties have to agree on the state of the ledger. Solving this problem is the main contribution of the development of Bitcoin.

Bitcoin assumes an honest majority in a public system [15]. Therefore, the consistence and correctness of the ledger becomes a voting problem. If every participant in the system is able to cast one vote, then the ledger is correct and consistent if there exists an honest majority. However, voting in a public distributed system remains a hard problem, especially considering sybil attacks [3]. Sybil attacks enable an attacker to forge identities and obtain a dishonest majority. To solve this Bitcoin has to protect against sybil attacks, without a central authority figure. Bitcoin achieves this by binding voting right to computational power. Since computational power is a ressource harder to increase than the number of forged identities, obtaining a dishonest majority becomes much harder.

Bitcoin binds votes to computational ressources through cryptographic puzzles. In order for a peer to participate in the system, he has to solve a cryptographic puzzle. This process, also known as mining, consumes the computational ressources of the peer. Simply forcing participants to waste all their computing power, only to participate in the system would lead to no participants. Therefore, this mining process has to be incentivised. If a miner mines a block, he receives a mining reward. This helps spreading the overall computational power of the network among multiple different parties, since every party is competing for mining rewards. Without a mining reward there be no economical reason to spend computational ressources on bitcoin.

Mining is a process, which builds inherently on the idea of incentives. It is logical, that miners will strive for the best strategy to maximize rewards. A mining protocol maximizing rewards is called incentive compatible. It can also be assumed that a miner will always execute the mining protocol, which maximizes rewards. The original bitcoin mining protocol is assumed to be incentive compatible [2]. However, Eyal and Sirer show the existence of deviant mining protocols with greater rewards [4]. Miners executing such protocols are called selfish miners. This imposes a threat, since it reduces the performance of the overall system. Additionally, selfish miners obtain a greater voting power than their computational ressources allow and as a result tilt the honest majority balance.

The central goal of this master thesis is to analyze the impact of selfish mining as an attack on blockchain systems. While it has been established that selfish mining imposes a threat on blockchain, it remains unassessed how big the impact is. Additionally, selfish mining is highly influenced by networking effects. Therefore, in order to assess the impact of selfish mining, analysis has to be performed in a model, which also captures the underlying network.

Chapter 2

Related Work

Selfish mining is a statistical attack. In order to analyze profit, it is therefore beneficial to analytically model selfish mining. Moreover to study the impact of deviating mining strategies and gain realisitic resulsts, it is very important to represent the blockchain network as close to reality as possible in a mining model. In the following recent selfish mining models as well as network models will be discussed.

2.1 Selfish Mining Models

Proof-of-work Mining is most commonly modelled through Markov decision processes. It is generally used to model decision making, where the outcomes are influenced by random processes and the decision of the decision maker [9]. In the case of selfish mining controls his decision making process and for the rest of the network, the block arrival and block propagation is modelled through stochastic processes. Once implemented, a Markov model can be used to analytically estimate system properties. Selfish mining is concerned with maximizing obtained mining rewards, which is also called revenue.

Eyal and Sirer first described a selfish mining model based on Markov decision processes [4]. They ran simulations to estimate revenue gain. Since blockchain systems use a peer-to-peer network to propagate mined blocks, it is important to analyze how the network was implemented in this model. Block propagation time is considered negligible compared to block generation time [4]. As a result, Eyal and Sirer consider communication to be instantaneous [4]. They found that selfish mining increases profitiability for a relative mining power greater than 25% compared to the network.

Sapirshtein et al. further extended the model to consider all possible actions a selfish miner can perceive. Block propagation time remains unassessed, since it is again considered to be much smaller than block generation time. Sapirshtein et al. again utilize Markov decision processes to model the system. They find a number of optimal policies and provide an analysis on the upper bound of revenue increase through selfish mining strategies. This Markov model was widely used and adopted in other research directions studying other aspects of selfish mining. For example, the behavior of multiple selfish miners was simulated through Leelavimolsilp et al. [11]. One of the main findings is that the lower bound on the profitability treshhold decreases with the number of selfish miners. Bai et al. [1] extended the model of Sapirshtein et al. [14] even further to analyze multiple selfish miners. This resulted in a more complex state space of the markov decision process. They show that in fact for the symmetric selfish miner, i.e. all selfish miners have the same hashrate, the profitability threshhold decreases, but for the asymmetric case the threshhold increases. The focus of this research was to deepen the understanding of different selfish mining strategies. However, networking factors remained unassessed.

Xiao et al. study the impact on the profitability threshhold and revenue gain of a networking advantage possessed by the selfish miner [16]. They model the network as a graph and find that networking advantage correlates to the betweenness centrality of the selfish miner. Additionally it highly affects the profitability threshhold and revenue gain of the attacker. This indicates that the structure of the network influences the selfish mining strategy. However, this model remains very abstract, since only the peer-to-peer layer and structure is modelled as a graph, disregarding any limitations imposed by physical infrastructure such as bandwidth. Nonetheless, it indicates that the underlying network influences the blockchain overlay, strengthening the assumption that there is a highly influencial dependency between networking effects and selfish mining.

It is not contested by any of the previous research that network capabilities and communication delay impact selfish mining [11], although most research model block propagation as instantaneous. In addition, most research which is concerned with selfish mining builts on top of the model presented by Sapirshtein et al.. This contributes to the negligence of networking effects, when analyzing selfish mining. Assuming that the underlying network does influence the system built on top, this master thesis aims to analyze the impact of networking effects on selfish mining. It is therefore important to represent the network in the model, that is used to analyze selfish mining.

2.2 Blockchain Network Models

Bitcoin and Proof-of-Work blockchains in general have been modelled and analyzed from a networking perspective. In order to study selfish mining with the context of networking effects it is necessary to analyze the network. Most blockchain network models are concerned with the estimation of consistency. Consistency is the property of a blockchain that all honest parties output the same block sequence. Garay et al. study the core of the Bitcoin protocol formally [6]. They analyze the protocol in a synchronous communication network and show persistence and liveness of committed transactions. They further proof that the adversarial computational power bound to reach Byzantine Agreement is $1/2$ of the network for a synchronized network. The adversarial bound decreases as the network drifts further away from synchronization [6]. The analysis of Garay et al. indicates that the networks synchronicity highly influences the behavior of proof-of-work blockchains.

Pass et al. propose a new network model to analyze blockchains in terms of consistency and liveness in an asynchronous network [13]. They do not make any assumptions of synchronicity and proof consistency in a network with adversarial delays that are a-priori bounded. They show that the proof of work hardness needs to be set as a function of the maximum network delay. New peers joining the network or peers getting corrupted are also modelled. They prove that Nakamoto's protocol satisfies consistency even in a network with message delays. However, those network delays are modelled to be always adversarial. This leaves out networking factors impacting the system which are caused by honest behavior.

Kiffer et al. built on top of the models of Garay et al. and Pass et al., but formulate a simple Markov chain based method to analyze consistency. Additionally, they provide lower bounds for consistency. They also analyze the GHOST protocol, where consensus is built over the heaviest observed subtree, in addition to the longest chain rule [10]. The model is based on rounds of communication. The modelled adversary controls a fraction of honest peers and can delay and reorder messages within a threshhold δ. The model therefore again captures only network attacks from an adversary, but disregards other networking effects.

Gervais et al. introduce a novel framework to analyse security and performance [7]. They model how network and consensus parameters influence stale block rate, block propagation times, throughput and security. Stale blocks are blocks which do not contribute to the consensus mechanism. Selfish mining is modelled as a Markov decision process. The network layer is characterized by block size and the information propagation mechanism. Gervais et al. simulate the system over a network consisting of point-to-point connections between peers [7]. Those channels are defined by latency and bandwidth. Latency is set using global IP latency statistics. One major result is that an increasing block size increases block propagation time linearly and stale block rate exponentially.

Gopalan et al. utilize rumor-spreading to implement a new stochastic network model [8]. They study stability and scalability of their model. Each peer communicates at a given rate his oldest blocks to his neighbors. Communication channels are also bandwidth limited. This setup introduces network delays to blocks, which depends on the instantaneous network congestion. Unlike previous stochastic network models, Gopalan et al. do not introduce delay based on sampling data, but rather on the communication behavior of peers. Since network congestion depends on the behavior of peers and selfish mining is a deviating behavior, the model introduced by Gopalan et al. will be used in the following to analyze selfish mining and networking effects.

Chapter 3

Model

In order to study the relationship of networking effects and selfish mining, it is essential to capture network properties in an analytic model. The model can then be used to estimate selfish mining profitability in dependence of assumed parameters

3.1 Bitcoin Mining Fundamentals

To understand selfish mining and its implications on network behavior, it is essential to understand Bitcoin mining in general. Bitcoin utilizes a proof-of-work blockchain as a distributed ledger technology. It includes transactions into so called blocks. Each block possesses a unique ID and references a previous block [15]. This construct builds a directed acyclic graph rooted in the genesis block.

A correct block includes a nonce, which solves the cryptographic proof-of-work puzzle. The challenge is to alter the nonce until the hash of the set of transactions, the hash of the previous block, and the nonce produce a partial hash collision. Essentially, the hash has to be smaller than some threshold value, which is also referred to as difficulty [15]. Thus, Bitcoin binds block creation to the computational resources a peer possesses, since the partial hash collision can only be solved through trial and error. The correctness of the block is easily verifiable through third parties. Thus, Bitcoin ensures a fair leader election through this process.

Bitcoin uses a peer-to-peer network to propagate the mined blocks in the system. The network is unstructured as every peer tries to maintain a minimum of eight connections and performs neighbor discovery over DNS and by asking neighbors [15]. Blocks are propagated over the peer-to-peer layer through flooding. Once a miner finds a proof-of-work solution, he can publish the block and receives rewards through transaction fees and mining rewards. This provides an incentive for the miner to generate as many correct blocks as possible [2].

Consensus is established over the longest chain rule [2]. This means that the block ending the longest chain determines the state of the blockchain. This also implies that a miner only receives rewards, if his mined blocks are included in the main chain. Thus, a miner wants to produce as many correct blocks, that are part of the main chain, as possible. A protocol maximizing reward gain is thus incentive compatible. A miner produces a share of blocks pro-

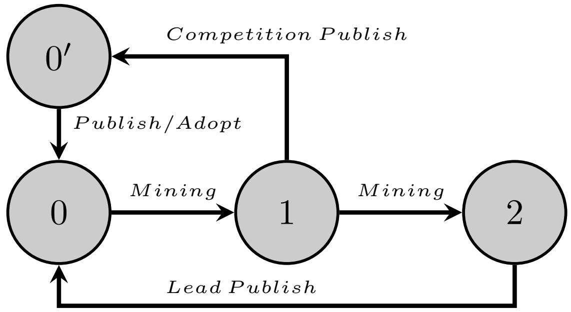


Figure 3.1: Abstract representation of state transtitions of eyal and sirer model for one selfish miner

portionally to his relative share of computational power of the whole network. Thus, a miner should produce a share of the main chain proportional to his relative share of computational power.

The original protocol, in the following also referred to as honest mining, assumes publishing blocks immediately after mining. Honest mining is assumed to be incentive compatible [2]. It follows, that no miner can earn disproportionate rewards by deviating from the protocol. Consequently, earning disproportionate rewards through deviation from the honest mining protocol would disprove Bitcoin's incentive compatability claim.

3.2 Selfish Mining

One protocol deviation is selfish mining, which was first introduced by Eyal and Sirer [4]. Selfish Mining is a vulnerability which aims at increasing revenue through block withholding. The selfish miner aims at producing a greater share of blocks in the main chain than his relative share of computational power of the network. Therefore, selfish mining violates Bitcoin's incentive compatibility claim, as it offers a more profitable mining protocol than honest mining. This is problematic, since it not only breaks fair leader election, but also results in potentially longer confirmation times for transactions of users. Eyal and Sirer model the network over a set of miners. Each miner finds a subsequent block after a time interval that is exponentially distributed [4]. They further define the revenue of a miner as his fraction of total blocks on the longest chain. The selfish miner withholds mined blocks [4]. This selfish miner now possesses a private chain, which differs from the publicly known chain. Based on the difference between those two chains, the selfish miner performs actions. For clarification the state space and actions are modelled in Figure 3.1. The numbers in the states indicate the lead of the private to the public chain, where s denotes the lead of the private chain compared to the public chain. We can identify a total of five different actions.

- $Mining$: This action means that the peer has mined a block. Mining adds the block to the private chain. It therefore causes s to increase.

- *Lead Publish*: When s increases to 2, the selfish miner will publish his private chain. It therefore causes s to change from 2 to 0.
- *Competition Publish*: When s is 1 and the selfish miner receives a block from another peer, he will publish his block of the same height from the private chain instead of the received one, to compete against the other miner. This causes a state transition to $0'$.
- *Publish*: If the selfish miner is in state $0'$, he is in a competition situation. The selfish miner will immediately publish his next mined block. This will cause the selfish miner to transition to state 0.
- *Adopt*: The selfish miner will adopt the main chain once he receives a new block in a competition situation.

Executing this protocol leads to a strict revenue increase, if the mining power is greater than 33% according to Eyal and Sirer [4].

Network Propagation Factor Eyal and Sirer [4] also introduced a selfish mining focused network metric, called γ. γ is measured in competition publish situations 3.2. In a competition publish situation two competing blocks are propagated through the network. An honest peer will adopt the first block increasing his blockchain height. γ describes the computational power fraction of the network the selfish miner block reaches before the other block. This is an advantage since now the selfish miner and the fraction he reached first start mining together on a consecutive block, securing the selfish miners block in the main chain while the competing block remains a fork.

3.3 Blockchain Gossip Model

The Blockchain Gossip Model of Gopalan et al. consists of a set of peers P connected through a peer-to-peer network. Peers add blocks to the blockchain through a process called mining. The peer-to-peer network is modelled as an undirected Graph $H = (V, E)$. An edge $(i, j) \in E$ represents communication possibilities between $v_i \in V$ and $v_j \in V$. The set of vertices is finite, such that $|V| = N \in \mathbb{N}$. Vertices are associated with peers, such that v_i represents peer $p_i \in P$. Additionally, a directed acyclic graph $G_{p_i}(t) = (B_{G_{p_i}}(t), E_{G_{p_i}}(t))$ is associated with each peer p_i, at each point in time $t \in \mathbb{R}+$. The vertex set $B_{G_{p_i}}(t) \subset \mathbb{N}$ represents the blocks known of peer p_i at time t. The associated edge set of $E_{G_{p_i}}(t)$ represents references between blocks. The following holds true for shorter notations:

$$B_G(t) = \cup_{i=1}^{N} B_{G_{p_i}}(t) \text{ and } E_G(t) = \cup_{i=1}^{N} E_{G_{p_i}}(t) \tag{3.1}$$

Furthermore, the following equations hold for the principle of blockchains:

$$\forall p \in P : G_{p_i}(0) = (\{0\}, \emptyset) \tag{3.2}$$

$$t_1 < t_2 \rightarrow B_{G_{p_i}}(t_1) \subseteq B_{G_{p_i}}(t_2) \tag{3.3}$$

Key Elements	**Short Description**
$G_{p_i}(t)$	Blockchain graph representation consisting of $(B_{G_{p_i}}(t), E_{G_{p_i}}(t))$ for every peer, time variant
$B_{G_{p_i}}(t)$	Blockset of each peer, time variant
$E_{G_{p_i}}(t))$	Edgeset of each peer, time variant
T_{p_i}	The communication process for each peer, sends blocks from p_i outwards
$L_{p_i}(t)$	The set of blocks farthest away from the genesis block

Figure 3.2: Overview of Key Model Elements

$$t_1 < t_2 \rightarrow E_{G_{p_i}}(t_1) \subseteq E_{G_{p_i}}(t_2) \tag{3.4}$$

Note that in this representation 0 denotes the genesis block described in Equation 3.2. $G_{p_i}(t)$ evolves over time. Blocks arrive over continuous time according to a stationary point process A with intensity λ. Each block $b \in \mathbb{N}$ arrives at a random peer p_i. This models peer p_i mining block b at time t and that at this time the block is also added to $B_{G_{p_i}}(t)$. References are added to $E_{G_{p_i}}(t)$ according to policy and depending on $G_{p_i}(t^-)$, where t^- is a moment in time infinitesimally before t. O_i denotes the set of outgoing neighbors of block i.

The communication is modelled as a marked point process T_{p_i}. Each mark corresponds to another peer $p_j \in P \backslash \{p_i\}$. In an epoch peer p_i contacts p_j and thus, adds the lowest numbered block of $B_{p_i}(t) \backslash B_{p_j}(t)$ to the set of Vertices B_{p_j}. If $B_{p_i}(t) \backslash B_{p_j}(t)$ is not empty, E_{p_j} is also updated accordingly.

The peer-to-peer network dynamics are modelled as a continuous time rumor-spreading process with exogenous arrivals [8]. Since communication is bound to the process T_{p_i}, the block dissemination is bandwidth limited. Reference selection and thus O_{p_i} is chosen according to longest chain policies [8]. Let $L_{p_i}(t)$ denote the set of blocks farthest away from the genesis block 0, known to peer p_i at time t.

$$L_{p_i}(t) := \{j \in B_{p_i}(t) : d(j, 0) \geq d(j', 0), \forall j' \in B_{p_i}(t)\} \tag{3.5}$$

Let $max\_dist(G_{p_i}(t))$ denote that distance. Note that the set $O_{p_i} \cap L_{p_i}(t)$ is non empty. This constructs a simple directed acyclic graph. The Tree Policy [8] can then be determined as $|O_{p_i}| = 1$ and establishes the following relationship:

$$|E_{G_{p_i}}(t)| = |B_{G_{p_i}}(t)| - 1 \tag{3.6}$$

Every block will have exactly one outgoing reference, according to a deterministic rule [8]. Gopalan et al. assume that the block with the lower index number will be chosen.

To sum up we have given a short overview of important key elements of the model introduced by Gopalan et al. [8]in Table 3.2.

Bitcoin's mining protocol has the goal of fair leader election. However, deviating mining protocols show vulnerabilities in the incentive system. The above section introduced a model to analyze network properties of blockchain systems in an analytical manner. Additionally, it discussed adversarial mining protocols and the model they have been analyzed on. The next section will develop a new model, based on the Blockchain Gossip Model introduced by Gopalan et al. [8]. This model can then be utilized to also analyze adversarial mining strategies, such as selfish mining.

Chapter 4

A Network-centric Model for Selfish Mining

The goal of this thesis is to analyze the relationship between selfish mining and networking effects. The model proposed by Gopalan et al. [8] appears to be suitable to analyze this relationship. It offers networking effects because it is based on rumor spreading, as well as analytical properties, such as that it can be utilized to simulate the system execution time. The first step to analyze the relationship between selfish mining and networking effects is to identify important system factors. Based on the model introduced by Gopalan et al. [8], a new model will be constructed which also includes adversarial behavior, such as selfish mining. This will be called the Selfish Rumor Model. Through the combination of a rumor-spreading model and an abstract blockchain mining protocol, the impact of networking effects on selfish mining can be studied.

4.1 System Factors

Under the assumption that adversarial mining strategies and network properties influence each other it is important to categorize those factors and characterize their interdependencies. Factors can be split up into local and global factors, as well as network and mining properties.

4.1.1 Network Factors

A blockchain system is a distributed system and utilizes a network to propagate information. As such the network behavior is important for the blockchain system. Network factors influence how information is spread. Those factors are visualized in Table 4.1. An important characteristic is how fast information is spread. On a global level this is highly influenced by how big the network is. In addition to network size, the network graph topology also plays a major role in how information is propagated through the network. In general a more densely connected network can propagate information faster to each peer. Another crucial factor is

	Network Factors	**Mining Factors**	**Adversarial Factors**
Global Factors	Network Graph Topology Network Size Bandwidth Distribution	Mining Power Distribution Difficulty	Multiple Selfish Miners
Local Factors	Geografic Location Topological Location Bandwidth	Mining Power Mining Strategy	

Figure 4.1: Key factors influencing the selfish rumor model

bandwidth distribution. The Bandwidth of a peer determines at what rate he can communicate. On a given network size the combination of bandwidth distribution and network graph topology influences drastically how fast information can spread. As an example let us consider an exponential degree distribution and an exponential bandwidth distribution. On the one hand information will spread reasonably fast, when a high bandwidth node also has a high degree. On the other hand if a node with a very low bandwidth has a high degree, he will become a bottleneck.

From a local point of view the location of a peer becomes important. Both geografic and topological location determine how fast a peer receives and sends information. Together with the amount of bandwidth the location also determines how much information is routed through that specific peer. Generally speaking a more central peer receives and sends information faster and also becomes a intermediate routing point more often. One can measure the centrality of a node by analyzing his betweeness centrality. The betweeness centrality is a measure on how many shortest paths a node lies between any other pair of two nodes.

4.1.2 Mining Factors

Additionally, in a blockchain system key factors include mining factors. Those factors influence the mining process of each individual miner. On a global scale mining power distribution and difficulty are characteristic. Difficulty determines at which rate a new block is introduced to the network. Mining power distribution determines, where the new block is how likely to arrive. They are both key factors influencing the average block propagation, since an ill combination of both can lead to network congestion. As an example consider a peer with a high relative mining power, who only has one outgoing connection to the network. This one connection is clearly a bottleneck prone to lead to congestion. With a low difficulty the peer will mine many blocks in a short amount of time and will flood the peer connecting him to the network. This will result in an overall higher average block propagation time.

From a local point of view a miners mining process is mainly influenced by his mining power and the executed mining strategy. If a miner recruits additional computing power, he will gain increased mining power and produce more blocks. If a miner executes an adversarial mining strategy such as selfish mining in contrast to honest mining, he will have a different reference selection and a different block publishing behavior.

4.1.3 Adversarial Factors

Adversarial factors also influence a blockchain system. Because of the mining protocol a blockchain system faces additional threads outside of adversaries attacking the underlying network structure. Peers can execute block withholding strategies such as selfish mining. By executing selfish mining strategies the forkrate of the blockchain is increased and the overall growth decreased. Multiple selfish miners escalate this problem.
We can denote that the network, mining and adversarial factors mentioned above play a crucial part for the behavior of the overall system. It is therefore important to select those factors carefully and construct scenarios which analyze a wide variety of possible system factor combinations. In the next section we will now introduce the Selfish Rumor Model.

4.2 Selfish Rumor Model

The selfish miner behaves different to an honest miner and therefore can be modelled by altering the reference selection and communication process. The reference selection process is policy driven, and can thus be modified by providing a new selfish policy. However, for the analysis of Gopalan et al. [8] the edge set is only represented in an abstract form, since at most the height of a block is important. As a result, specifics concerning the edge set are never discussed. For selfish mining to be implemented into the model, the reference selection process becomes very important and has to be further specified.

Reference Selection Process Specification Gopalan et al. [8] introduce an edge set $E_{G_{p_i}}(t)$ for each peer for each point in time. Those edges represent selected references. Additionally, we will introduce the notion of a Topblock $b_{top_{pi}}$. The following rules apply:

1. $b_{top_{pi}}(t) \in L_{p_i}(t)$

2. If peer p_j updates peer p_i with a block b' at time t, $b_{top_{pi}}(t) = b'$ if the height of b' is strictly greater than the height of the previous Topblock $b_{top_{pi}}(t^-)$.

3. If on an event of A a new block b arrives to peer p_i at time t, he will select $b_{top_{pi}}(t^-)$ as the parent reference of b. As a result $(b, b_{top_{pi}}(t^-)$ will be added to $E_{G_{p_i}}(t))$ and will be communicated through the network. Furthermore, $b_{top_{pi}}(t)$ will be set to b, such that $b_{top_{pi}}(t) = b$.

Through the above described rules reference selection is deterministically defined. It binds reference selection to the block arriving first to a peer. This is important, since a clear reference selection definition is crucial for estimating the success of selfish mining. The remainder of the section will discuss the new Selfish Rumor Model.

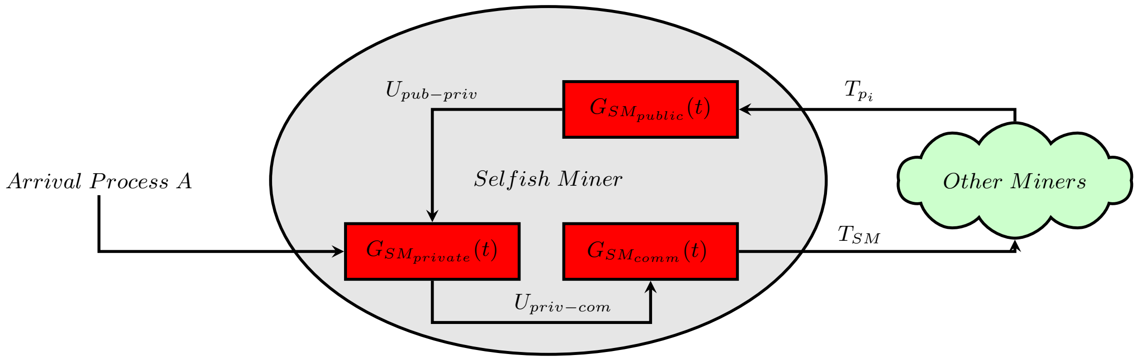


Figure 4.2: Abstract representation of model entities and communication processes

Selfish Rumor Model The Selfish Rumor Model is based on the Blockchain Gossip Model introduced by Gopalan et al. [8]. The construction of the Selfish Rumor Model can be seen as a modification of the Blockchain Gossip Model and is best explained in that manner. However, introducing adversarial mining strategies and altering key factors such as mining power distribution, changes the new model in such a drastic way that it has to be considered a new model. The first aspect to be modified is the communication process. Key idea of selfish mining is block withholding. The selfish miner possesses three blockchain representations.

- $G_{SM_{public}}(t)$: which is updated by other peers.
- $G_{SM_{comm}}(t)$: which is used to update other peers.
- $G_{SM_{private}}(t)$: with the following relations:
  - $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$.
  - $G_{SM_{private}}(t) \backslash G_{SM_{public}}(t)$ represents blocks mined but unpublished by the selfish miner.

The concept has been visualized in Figure 4.2. A total number of five processes is used to let all entities interact with each other.

- $Arrival\ Process\ A$: Blocks arrive to the selfish miner over the external arrival process A.
- T_{p_i}: Ensures blocks from other peers are communicated to $G_{SM_{public}}(t)$.
- $U_{pub-priv}$: Ensures that $G_{SM_{public}}(t) \subseteq G_{SM_{private}}(t)$ holds true, meaning $U_{pub-priv}$ updates $G_{SM_{private}}(t)$, when new blocks arrive to $G_{SM_{public}}(t)$ from other peers.
- $U_{priv-com}$: Updates $G_{SM_{comm}}(t)$ according to $G_{SM_{private}}(t)$ and the selfish mining rules S.
- T_{SM}: Ensures other peers are updated with blocks from $G_{SM_{comm}}(t)$.

Peer $SM \in P$ has an associated policy slightly different to the policy described in the gopalan model 3.5. Note that to follow the Tree Policy [8], a deterministic rule has to be established for the case that $|O_{SM} \cap L_{SM}(t)| > 1$. Assume that SM has the knowledge of the set of blocks mined through him, $M_{SM}(t) \subset B_{G_{SM}}(t)$. SM will set

$$(L_{SM}(t) \cap M_{SM}(t)) \neq \emptyset \rightarrow L'_{SM}(t) \subset (L_{SM}(t) \cap M_{SM}(t)) \tag{4.1}$$

It then follows that $|L'_{SM}(t)| = 1$. This modified tree policy sets references according to the original selfish mining protocol described by Eyal and Sirer.
S is a set of rules which describes how $G_{SM_{private}}(t)$ updates $G_{SM_{comm}}(t)$. The rules have to follow the state description of Eyal and Sirer 3.2. Therefore we need a state variable describing the difference between private and public chain. Let s be the state variable determining selfish mining actions [4]. Then s can be described as a difference between $G_{SM_{private}}(t)$ and $G_{SM_{public}}(t)$.

$$max\_dist\_mined(G_{SM_{private}}(t)) := d(j, 0), j \in M_{p_i}(t) \tag{4.2}$$

$$s(t) := max\_dist\_mined(G_{SM_{private}}(t)) - max\_dist(G_{SM_{public}}(t)) \tag{4.3}$$

Let t_{inc} refer to the set of times, where s increased and analogous t_{dec} refer to the set of times, where s decreased. Selfish mining is protocol, which needs a formulation of states in order to be characterized. Gopalan et al. introduced t^- as a point in time infinitesimally before t. In addition to describe selfish mining a function is needed to access the point in time where s changed last. Let $f_{-1}(t)$ be a function that outputs the point in time, where s changed the latest before t. Now all tools are available to characterize the selfish mining protocol on top of the stochastic network model introduced by Gopalan et al..
$U_{priv-com}$ can be characterized through four kind of update actions. Analogous to Subsection 3.2, those actions are $Lead\ Publish$, $Competition\ Publish$, $Publish$ and $Adopt$. $Mining$, the fifth action described in Subsection 3.2, is modelled through the arrival process. This can be used to model the selfish mining protocol desribed by Eyal and Sirer.

1. $Lead\ Publish$: Assume $t \in t_{inc}$ and $s(t) \geq 2$, then $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{comm}}(t) = G_{SM_{private}}(t)$. Once the selfish miner has established a lead of two blocks against the public chain, he will update the blockchain representation used for communication towards other peers. In other words, he publishes the private chain.

2. $Competition\ Publish$: Assume $t \in t_{dec}$, $s(t) = 0$, $s(f_{-1}(t)) = 1$, $s(f_{-1}(t)^-) = 0$. This means that the selfish miner mined a block, did not publish it and now received a block from another of the same height. This leads to the competition scenario. Accordingly, $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0. The selfish miner will publish the block, which caused the private chain to lead by one against the public chain, before he received a new block. This transitions to

$$0'(t) \rightarrow \big(t \in t_{dec} \wedge s(t) = 0 \wedge s(f_{-1}(t)) = 1 \wedge s(f_{-1}(t)^-) = 0\big) \tag{4.4}$$

   This situation $0'$ is also shown and visualized in Subsection 3.2 and causes the selfish miner to execute honest mining for only the next step.

3. $Publish$: Assume $0'(t^-) = \top$ and $t \in t_{inc}$, $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that it includes the subgraph induced by the nodes on the paths between $L'_{SM}(t)$ and 0. The selfish miner will publish his newly mined block, because he was previously in $0'$.

4. $Adopt$: Assume $0'(t^-) = \top$ and $s(t) = -1$, then $U_{priv-com}$ updates $G_{SM_{comm}}(t)$, such that $G_{SM_{comm}}(t) = G_{SM_{private}}(t)$. The selfish miner will adopt the public chain, because he was previously in $0'$.

The above section introduced a new model, the Selfish Rumor Model. Through the combination of rumor-spreading mechanisms for an abstract network layer representation and adversarial mining strategies, this model can be used to analyze the relationship between the selfish mining attack and networking effects.

Chapter 5

Evaluation

The following section utilizes the simulative implementation of the Blockchain Gossip Model to evaluate the relationship between selfish mining and networking effects. Additionally, the model will be validated against data provided by Gopalan et al. [8]and real world data of the Bitcoin system.

5.1 Simpy Blockchain Simulator

The core implementation is based on simpy [12]. Simpy is a discrete event simulator written in python. As a result the Simpy Blockchain Simulator is also written in python. The Selfish Rumor Model consists mainly of four parts.

- Networkgraph representation
- Blockchain representation
- Block Arrival Process representation
- Communication Process representation

The network graph is represented by an adjacency matrix. The blockchain representation is a set of blocks and a set of edges for each peer, which are developing over time. The block arrival process and the communication process are modelled as a Poisson process [**?**]. This is mirrored in a Simpy process with an exponentially distributed interarrival time between scheduled events. On each event of the block arrival process a block arrives at a random peer. This means that the event triggered by the block arrival process updates the blockchain datastructure accordingly. At each event of the communication process T_i a peer p_i tries to update a certain peer p_j according to the epoch associated with the event. This results in a comparison between the datastructures associated with p_i and p_j and an update of p_j, if it is possible.

Even though the basic implementation is simple, there are various parameters which influence the system behaviour greatly. The following list shall give an overview:

- Average of interarrival times - This is the rate block arrival process and communication process trigger events.
- Topology of the network graph - The Network resulting from the adjacency matrix has a great influence on the bahviour of the system.
- Block selection - In a scenario, where multiple blocks could be transferred from one peer to another one has to be selected. How this block is selected influences system baviour.
- Network size - The number of peers
- Mining power distribution - The mining power distribution influences the peer selection. Peer selection is the process of deciding which peer gets the new block once the block arrival process triggers an event.

The above discussed parameters can be modified in order to capture different systems.

5.2 Validation of the Simpy Blockchain Simulator

In the following section the Simpy Blockchain Simulator is validated against synthetic experiments published by the original authors of the blockchain gossip model. Additionally real world data from Bitcoin will be additionally used to validate the legitimacy of the Simpy Blockchain Simulator. This will lay the fundamant for further analysis concerning the network and selfish mining.

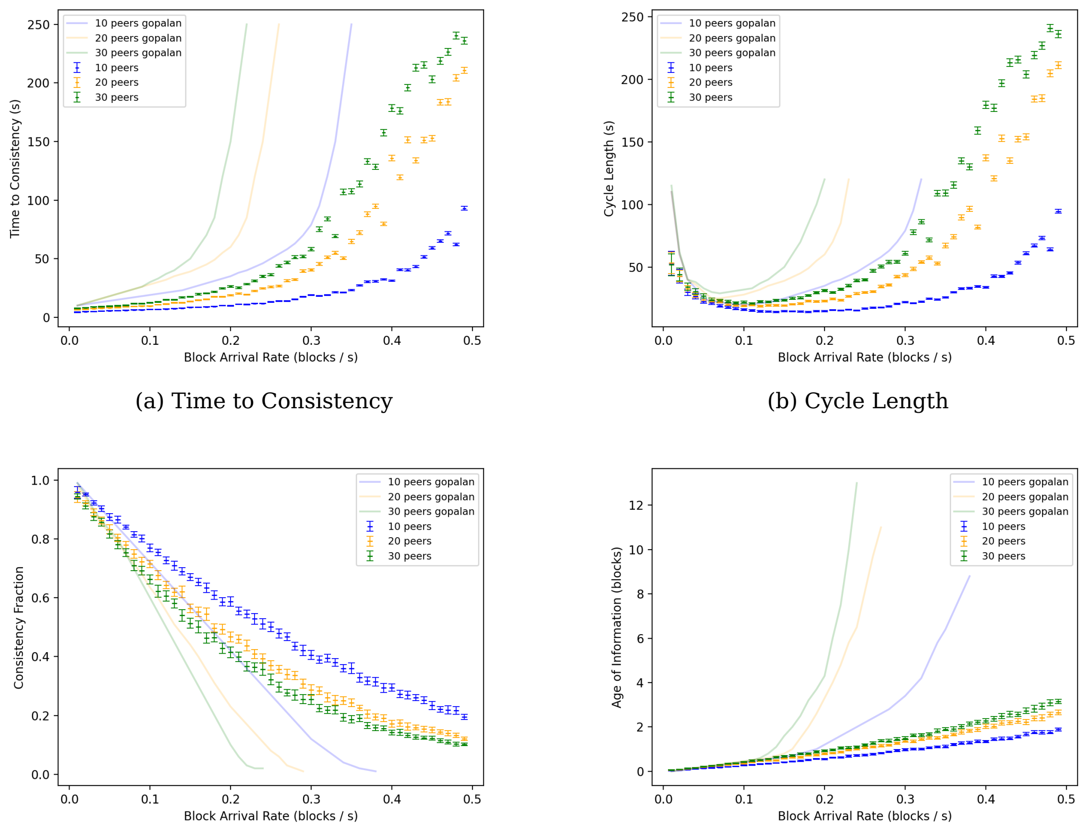
5.2.1 Validation of Simulator against Gopalan et al. [8]

In the synthetic data experiments of Gopalan et al. [8] they analyze the the network for 10, 20 and 30 peers. The network topology is a complete graph. Thus, the adjacency matrix is the unit matrix. The authors introduce four key metrics to analyze the system. Those are

- Time to Consistency — The average time an inconsistent system needs to reach a state of consistency
- Cycle Length — The sum of the average time to consistency and the average of the time the system stays consistent
- Consistency Fraction — The average fraction of peers that are consistent at each point in time
- Age of Information — The average number of blocks an average peer is away from the consistency state

All metrics mentioned above refer to the term consistency. The Consistency is defined as $B_G(t)$ 3.1, the unison of all blocks produced by the block arrival process A. In order to evaluate to capture the same system, that was analyzed by the authors the parameters are setup similar. These metrics can be used to verify whether the Simpy Blockchain Simulator is achieving similar numbers to the implementation of Gopalan et al. [8].

- Average of interarrival times - The interarrival time of the communication process is set to 1s. The interarrival time of the block arrival process is a variable.
- Topology of the network graph - The network topology is a complete graph.
- Block selection - In a scenario, where multiple blocks could be transferred from one peer to another the block with the lowest index number is chosen.
- Network size - The number of peers is set to 10, 20 and 30 accordingly.
- Mining power distribution - The mining power distribution is uniform.



(a) Time to Consistency

(b) Cycle Length

(c) Consistency Fraction

(d) Age of Information

Figure 5.1: Comparison between Simpy Blockchain Simulator and values produced by Gopalan et al. [8]

The metrics of time to consistency and cycle length are very closely related, because both rely on the time the system needs to reach consistency. Figure 5.1a and Figure 5.1b show this close

relationship. Additionally the comparison between the Simpy BLockchain Simulator shows a very similar tendency in both metrics. Especially in Figure 5.1a it is observable that the curve has the same shape, only flatter. Figure 5.1a shows that peernumber and block arrival rate are proportional to the average time to consistency. Since cycle c´ength is the sum of the average time to consistency and the average of the time the system stays consistent the same behavior can be observed in Figure 5.1b. Additionally Figure 5.1b shows that for very small numbers for the block arrival rate the cycle length increases again. When the system has a low block arrival rate the system tends to stay longer in a state of consistency, which is due to the fact that the idle time increases.
Consistency fraction and age of information are both metrics measuring the consistency of an average peer. The consistency fraction is the fraction of peers, which have a blockset equal to $B_G(t)$ 3.1. For both the simulation results by Gopalan et al. [8] and the Simpy Blockchain Simulator we can observe, that the consistency fraction decreases with an increasing blockrate and peer number. While the exact numbers do differ, similar shapes can again be observed. The age of information metric analyzes how much an average peer differs from $B_G(t)$ 3.1. It showcases an increase for an increasing blockrate and peer number.
The differences indicate that information spreads faster in the Simpy Blockchain Simulator. After a brief discussion with Gopalan et al. [8] , they confirmed that this might be due to the fact, that in the simpy version communication processes are handled truly concurrently.

5.2.2 Validation of Simulator against Bitcoin data

This section verifies the model against a real world blockchain system, the Bitcoin network. The Selfish Rumor Model implements a network model with a block level abstraction of blockchain systems. Researchers of the Karlsruher Institut für Technologie [5] monitor the Bitcoin network and obtain data of, for example, the current block propagation delay distribution. Since the Selfish Rumor Model implements a block level abstraction layer the current block propagation delay distribution becomes an important metric to analyze in order to compare the model against the real world system. Figure 5.2 visualizes the current block propagation delay distribution of Bitcoin. The 1% of the largest delays have been filtered out and the data points have been grouped in $50ms$ steps. A significant characteristic seen in the plot is the high peek at $400ms$ and the long tail going up to about $10000ms$ delay. Around 72% of peers have a block propagation delay below $500ms$. Figure 5.3 shows the block delay distribution for the selfish rumor model. The average interarrival time of the communication processes was set to $1s$. The average maximum of peers have bock propagation delay of $7s$ and 60% of the network receive the block in under $7s$. The absolute difference between the observed selfish rumor model and the delays of Bitcoin have a difference in magnitude. This is mostly linked to the average interarrival time of the communication processes. However, we can observe a similar distinct longtail distribution as in Bitcoin, which indicates that gossiping models are a good way to model blockchain systems on a block abstraction level.

Bitcoin Parameter Setup The simulator was evaluated against real world data of the Bitcoin system with a setup of 500 peers, a 16-regular-random graph as the network topology, block arrival process rate of 600s and a communication process rate of 1s. Because Bitcoin is

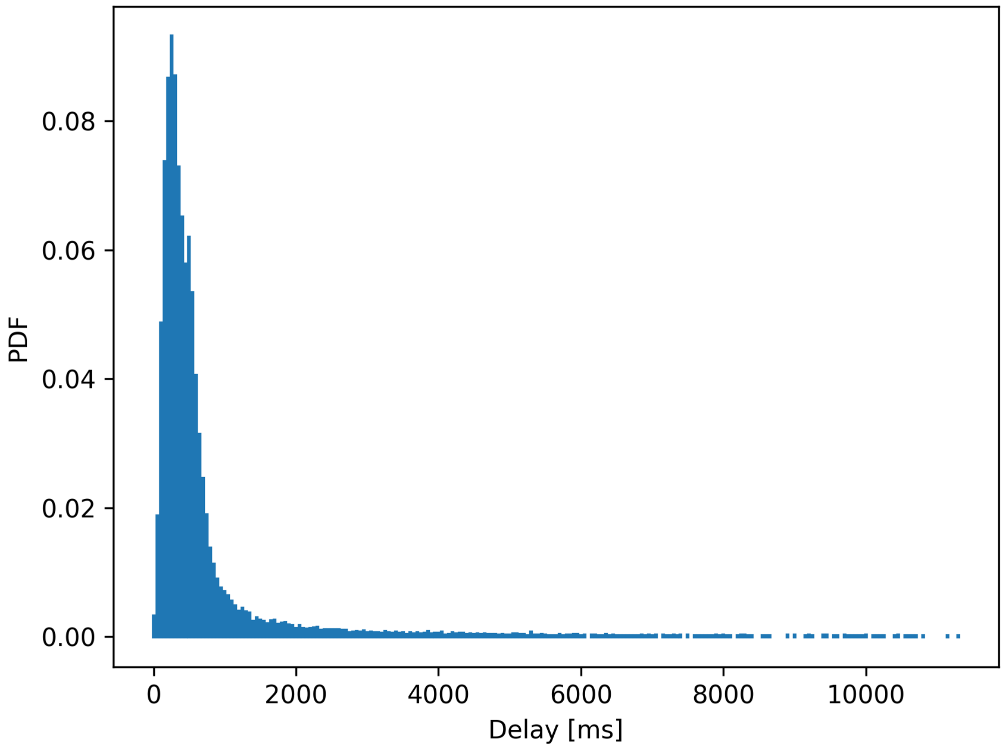


Figure 5.2: Bitcoin Current Block Propagation Delay Distribution [5]

the most prominent blockchain system, this parameter setup will be used for following evaluations, unless stated otherwise. This choice of parameters will also be reffered to as the Bitcoin parameter setup.

5.3 Selfish Mining and Networking Effects

Networking effects and selfish mining can be analyzed from a global and a local point of view, cf. Table 4.1. The system analysis introduced by Gopalan et al. [8]assesses the system mainly in terms of consistency and blockchain growth, as discussed in the previous section. Both, blockchain growth and consistency, are influenced by adversarial mining strategies such as selfish mining. The next subsection will analyze the system from a global perspective to determine the relationship between consistency, blockchain growth and selfish mining.

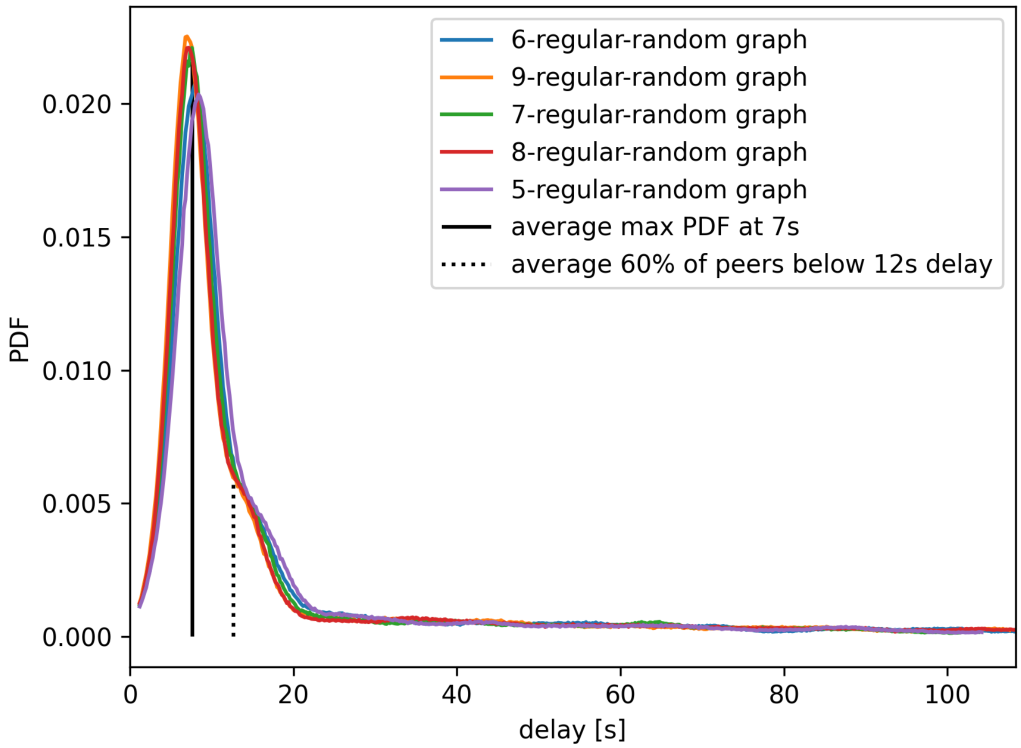


Figure 5.3: Selfish Rumor Model Block Propagation Delay Distribution, 300 peers, multiple degrees, selfish miner star topology, TODO: Bitcoin* Simulations

5.3.1 Selfish Mining and Global Network Characteristics

vllt das ans ende?

mainly focus on growth?

This section will analyze how global system behavior changes when peers executing selfish mining are introduced to the system. In Section 5.2.1 the Simpy Blockchain Simulator was verified against the results published by Gopalan et al. [8]. Gopalan et al. [8]analyzed the blockchain system from a global perspective using metrices based on growth and block propagation. Thus, those base metrices will be first used to analyze the global state with adversarial miners.

TODO: vgl. zwischen sm und nicht etc.

Selfish mining leads to intentional forks in the blockchain.

5.3.2 Selfish Mining in homogenous Network Setting

Eyal and Sirer [4] discovered a relationship between the relative computational power and the resulting revenue gain. They described that an increase in networking propagation factor and relative computational power results in an increased revenue gain. In a homogenous network every peer has the same degree and the same bandwidth. Thus, the peer executing selfish mining has no networking advantage. Figure 5.4 shows the revenue gain for 25%, 30%,

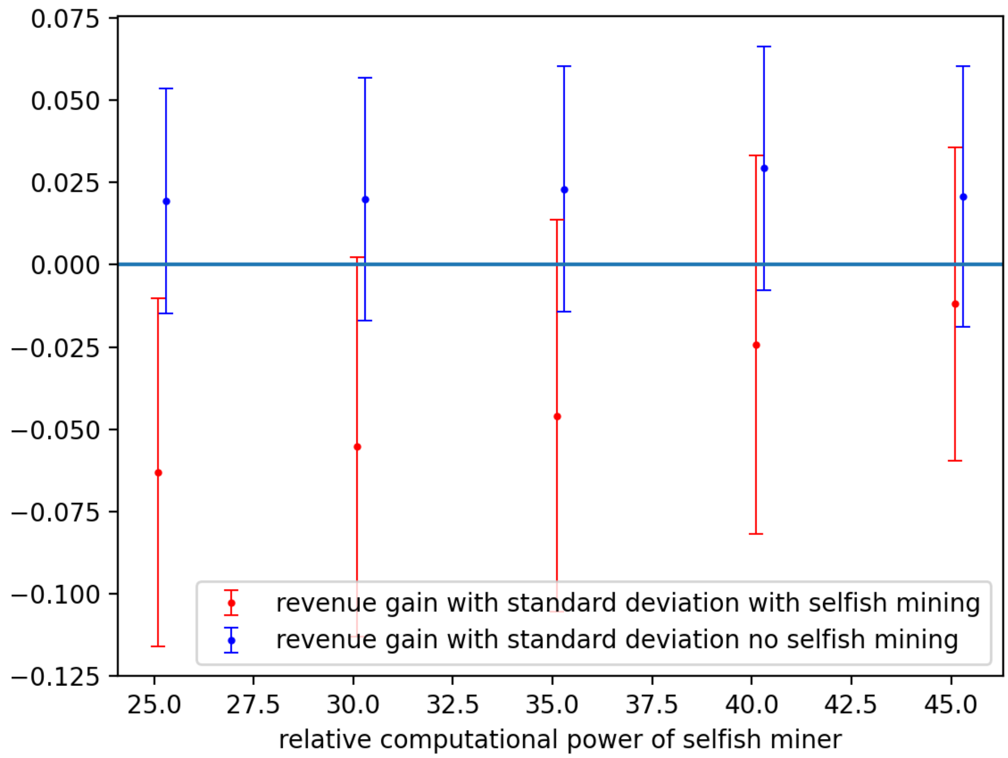


Figure 5.4: Revenue and blockproduction with standard deviation for peer 0 executing selfish mining for various relative hashrates

35%, 40% and 45% relative compuational power. The selfish miner possessed no networking advantage. The network consisted of 500 peers and the topology followed a 16-regular-random graph. Figure 5.4 shows that the revenue gain is strictly below 0. Increasing the relative computational power also increases revenue gain. Looking at the standard deviation and 98% confidence interval the revenue gain seems to be wide spread. Nonetheless, this contradicts the results of Eyal and Sirer [4] since they showed a strict revenue increase for $\alpha > 33\%$. Since α can be seen as the fraction of blocks a miner produces it is directly linked to the relative computational power this miner possesses. Thus, according to Eyal and Sirer [4] Figure 5.4 should be positive for the relative compuational power greater than 33%.

5.3.3 Selfish Mining with Networking Advantage

5.3.4 On Achievability of Networking Advantage

5.4 Networking effects and selfish mining

5.4.1 Global network characteristics and selfish mining

5.4.2 Networking effects impacting selfish mining

Central to this thesis remains the question how impactful network effects are on the performance of selfish mining. The comparison of local and global factors from a single peer point of view can be utilized to describe a networking advantage this peer possesses. Selfish mining is executed in order to gain revenue. Revenue gain can be measured by comparing the actual revenue to the relative computational power of the peer.

Computational Power and Selfish Mining

Network Advantage and Selfish Mining

Key aspects determining networking characteristics of a specific peer are his location relative to the network and his bandwidth, while key factors on a global scale are topology, network size and bandwidth distribution, as described in Table 4.1. If for example all peers in the network possess the same amount of bandwidth, increasing the bandwidth of the selfish miner will put him at a network advantage. If the network topology is described as a k-random regular graph, then allowing the selfish miner to connect to more then k peers will result in a networking advantage. This topological factor can be measured by graph metrices, such as betweeness centrality.

Betweeness Centrality If revenue gain is positively correlated to network advantage, than a goal of a selfish miner should be to increase his network advantage. As discussed before one can analyze the resulting networking advantage utilizing network metrices such as betweeness centrality. Figure 5.5 visualizes the betweeness centrality of one peer in various k-random-regular graphs. This specific peer, peer 0, is connected to all other nodes. The rest of the graph follows a random-regular structure. The x-axis visualizes the increasing k of the k-random-regular graph. As k increases the betweeness centrality of peer 0 decreases. For smaller node amounts the decrease is faster than for bigger node amounts. This is due to the absolut increase in k since a 20-regular-random graph of 100 peers is better connected than a 20-regular-random graph of 1100 peers. Better connected means in this case it is closer to a fully connected graph. The closer the graph gets to being fully connected the more the central role of peer 0 decreases, since the number of alternative paths between any other two peers rises. Nonetheless, Figure 5.5 shows the upper limit for peer 0. This means that in general a

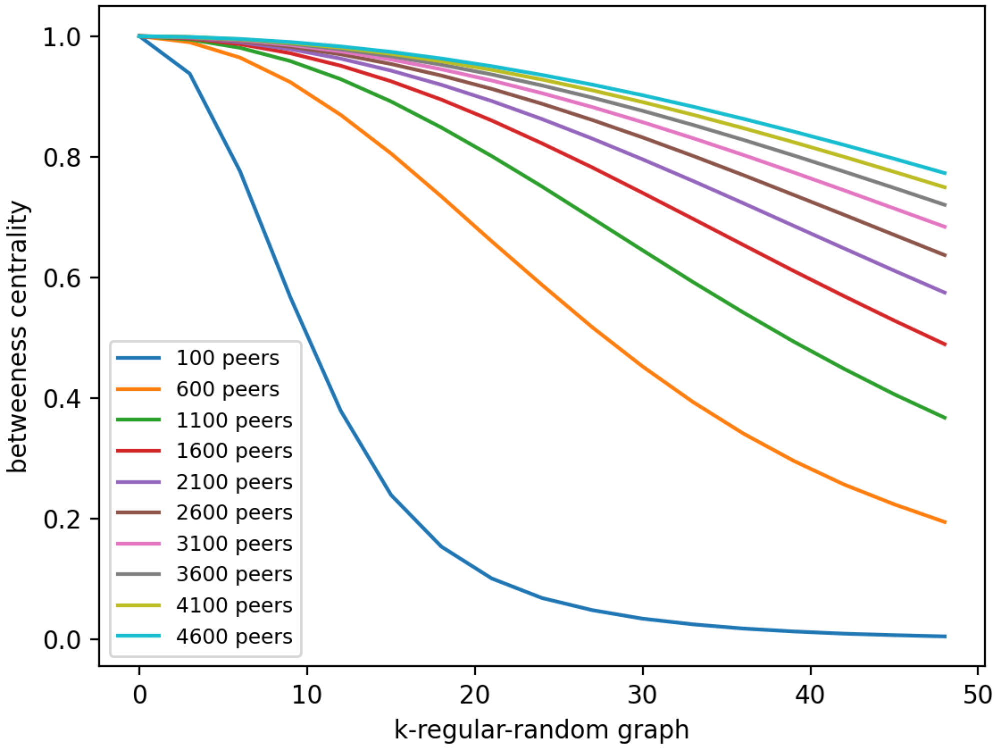


Figure 5.5: Betweeness centrality for various k-random-regular graphs for an all-connected peer 0

peer has to connect to a large amount of peers, compared to the network average in order to obtain any networking advantage.

Topological Networking Advantage

It can be assumed that a peer with a higher degree than the network average possesses a networking advantage. Thus, raising the degree of the selfish miner puts him at a topological networking advantage. We can then estimate the actual networking advantage based on the betweeness centrality and γ of the peer.

Figure 5.6 shows results of experiments, which were based on analyzing the impact of topological network advantage. The experiments were carried out with 100, 200 and 300 peers respectively. Figure 5.6 shows the results of the simulations for 200 peers, since no significant difference could be observed between 100, 200 and 300 peers simulations. The selfish miner possessed a relative computational power of 45% while the rest was exponentially distributed

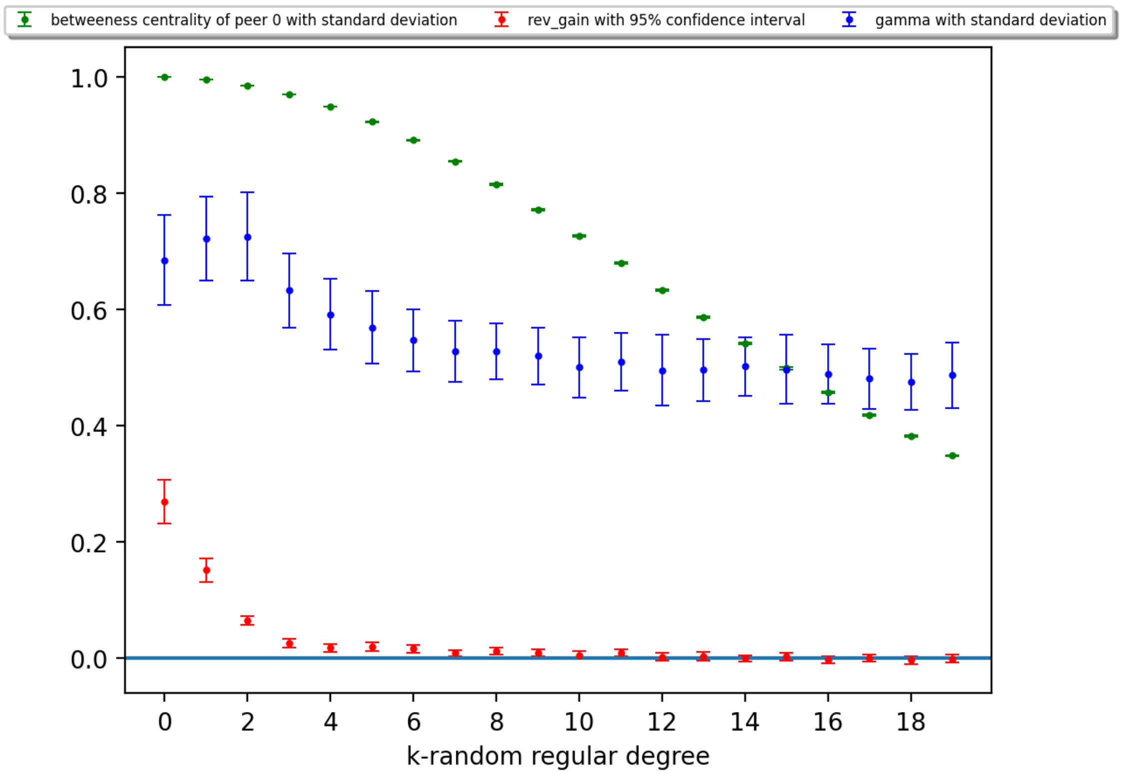


Figure 5.6: Betweeness Centrality, Revenue Gain and Network Propagation Factor, Topological Advantage Simulations with 200 Peers

on the remaining network. The interarrival time for communication process events was set to $1s$. The interarrival time for the block arrival process was set to $100s$. The selfish miner was connected to all peers. The rest of the network topology formed a k-random-regular graph, with an increasing k shown in the x-axis.

The green bars represent the betweeness centrality of the selfish miner. The betweeness centrality of the selfish miner decreases as k increases. The blue bar represents γ. We can observe that for each categroy of peer amounts γ is slightly highest for $k \in \{0, 1, 2\}$ and is rather constant for $k > 2$. γ constant at around 0.55 even though the betweeness centrality drops for increasing k's. Thus, it is likely that γ and betweeness centrality are not correlated.

We can further observe that the revenue gain, shown in red, is highest for $k = 0$. This is understandable since at $k = 0$ the selfish miner can effectively eclipse every peer. The selfish miner is in total control over the information flow. For $k = 1$ and $k = 2$ revenue gain begins decreasing towards 0 revenue gain. At $k = 1$ and $k = 2$ the selfish miner can still eclipse parts of the network but since overall connectedness increases, it becomes harder for the selfish miner to effectively eclipse every peer in the network. For $k > 2$ revenue gain still seems to be slightly above 0 which suggests that selfish mining is profitable.

Chapter 6

Conclusion

Bibliography

[1] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, and Q. Kong. A deep dive into blockchain selfish mining. Cryptology ePrint Archive, Report 2018/1084, 2018. `https://eprint.iacr.org/2018/1084`.

[2] S. Barber, X. Boyen, E. Shi, and E. Uzun. Bitter to better — how to make bitcoin a better currency. In A. D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 399–414, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-32946-3.

[3] J. R. Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[4] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. *CoRR*, abs/1311.0243, 2013. URL `http://arxiv.org/abs/1311.0243`.

[5] K. I. für Technologie. Bitcoin network monitor. `https://dsn.tm.kit.edu/bitcoin/`. Accessed: 2020-11-17.

[6] J. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[7] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.

[8] A. Gopalan, A. Sankararaman, A. Walid, and S. Vishwanath. Stability and scalability of blockchain systems, 2020.

[9] O. Ibe. *Markov processes for stochastic modeling*. Newnes, 2013.

[10] L. Kiffer, R. Rajaraman, and A. Shelat. A better method to analyze blockchain consistency. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 729–744, 2018.

[11] T. Leelavimolsilp, L. Tran-Thanh, and S. Stein. On the preliminary investigation of selfish mining strategy with multiple selfish miners. *CoRR*, abs/1802.02218, 2018. URL `http://arxiv.org/abs/1802.02218`.

[12] N. Matloff. Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, 2 (2009):1–33, 2008.

[13] R. Pass, L. Seeman, and A. Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.

[14] A. Sapirshtein, Y. Sompolinsky, and A. Zohar. Optimal selfish mining strategies in bitcoin. *CoRR*, abs/1507.06183, 2015. URL `http://arxiv.org/abs/1507.06183`.

[15] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys and Tutorials*, 18:1–1, 03 2016. doi: 10.1109/COMST.2016.2535718.

[16] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou. Modeling the impact of network connectivity on consensus security of proof-of-work blockchain, 2020.