An Analysis of Selfish Mining Attacker Incentives in Bitcoin and Ethereum

Yash Patel, Matt Weinberg

Abstract—**Both the Bitcoin and Ethereum decentralized systems rely on the same distributed public Blockchain mining model of transmitting and recording history. Previous thought was that this system would be held in check through a balanced proof of work incentive system. However, previous studies have revealed an attack dubbed "selfish mining" whereby miners can exploit this incentive system to increase their expected rewards. Such models have further been applied to studying the transaction fee system that is expected to largely replace the block rewards system over the following years. Despite extensive study in the past, such models have failed to include the associated effects of these selfish mining attacks on exchange rates, which is of primary focus herein. These models are further extended to the context of the Ethereum network, which has not been studied with respect to selfish mining previously. In addition, this study sought to align and compare the current empirical status of the Bitcoin and Ethereum networks to the model results, to determine whether it is currently in the miners' economic interest to engage in selfish mining or not. In the end, the necessary devaluation was studied as a function of the attacker's hashrate, selfish mining (SM) hashrate proportion, SM engagement delay, and uncle block reward (Ethereum), and it was found that the current state of Bitcoin and Ethereum are highly conducive to selfish mining, making it of interest to find countermeasures thereof in future studies.**

I. Introduction

Bitcoin and Ethereum are two technologies built off the Blockchain protocol that have recently been brought to pertinence with their "rich and extensive ecosystems" [9]. While the two systems have fundamentally different goals, namely Bitcoin being a "peer-to-peer version of electronic cash [that] allow[s] online payments to be sent directly from one party to another without going through a financial institution" [11] and Ethereum a "blockchain with a Turing-complete language and an effectively unlimited inter-transaction storage capability" [14], they both revolve around decentralized systems. That is, each of Bitcoin and Ethereum wish to create systems that rely on peer-to-peer communication rather than the modern-day prevalent client-server system architecture. As a result, both systems make use of the Blockchain protocol: a system designed and proposed by Satoshi Nakamoto. Through this introduction, a groundwork for the following systems is laid out, from which the results of the research conducted become clear:

- Blockchain Protocol
- Bitcoin Specifics
- Ethereum Specifics
- Selfish Mining

A. Blockchain Protocol

As described above, the fundamental goal of Blockchain is to create a system *without* a central authority to verify transactions in the network[1]. The Blockchain essentially serves as a *shared* public ledger that all the nodes on the network can consult to verify the transactions that have happened through history, thereby allowing the nodes to agree on the balances of all accounts. Thus, the difference between Blockchain and conventional networks is how and where this ledger is maintained. In conventional financial systems, there is a central bank that holds all such records privately (i.e. standard users of the financial system are unaware of what one another have spent) whereas in the Blockchain it is maintained and distributed to all the nodes *in* the network. Further, all transactions added to the Blockchain are added in units known as "blocks," though the size of each block (i.e. number of transactions in each) often varies.

The nodes responsible for maintaining this public ledger are known as the *miners* of the network, whose basic operational procedure is as follows. All nodes (potentially including miners) publicly broadcast any transactions they wish to make to the rest of the network. In the meantime, the miners aggregate these transactions into a block and try to solve "cryptographic puzzles" to essentially "earn the right" to add the block the public ledger. This system, described more extensively in the following two paragraphs, is referred to as "proof-of-work"[2]. Specifically, the cryptographic puzzle is finding a 32-bit nonce such that when a SHA-2 hash is calculated of the nonce with the transactions, the result is a value less than a certain threshold set by the Bitcoin system. That is, the Bitcoin system has an internal "difficulty level" numerical value that determines the maximum value the hashed result can take to be considered valid [12]. Clearly, by simply making this "difficulty constant" lower, the Bitcoin system effectively makes it harder for miners to validate blocks. This is done at regular intervals (i.e. every 2016 blocks) [7] to ensure that, as computers become more and more powerful, blocks are *not* confirmed at increasingly fast rates.

Since the SHA-2 hash function is thought to be a true hashing function, it is assumed to be a one-way function. In other words, given an input x, it is trivial to calculate $H(x)$. However, it is computationally infeasible, given some y, to

[1] As noted before, the Ethereum network does not deal directly with *transactions* per se, but this terminology is used through the following discussion for sake of clarity

[2] This is as opposed to "proof-of-stake," which is the major alternative currently conceptualized, though it has yet to achieve widespread adoption in any Blockchain system.

find the x that produces $H(x) = y$, i.e. the only way to do so is enumerating through all possible values of x. Thus, finding this 32-bit nonce is essentially a brute force search, whereby the miners try a random 32-bit number, calculate its hash with the transactions, and, if not less than the difficulty level (i.e. incorrect nonce), increment the nonce and repeat. Upon finding a correct nonce, the miner broadcasts that it has a valid block to the remainder of the network, who then check whether the broadcasted block was truly valid or a bogus. However, the Blockchain network is set up such that each computer is only connected to a small subset of the entire network[3], such that only after multiple re-broadcasts does the entire network hear about the new block. Specifically, upon receiving the broadcast and performing this verification calculation, mining rigs add the block to their ledgers and subsequently broadcast it to their neighbors. In this way, the block effectively becomes added to the network Blockchain. An important point to note regarding mining is that "nodes always consider the longest chain to be the correct one and will keep working on extending it" [11]. In this way, malicious users cannot easily rewrite history by broadcasting a block that ignores blocks recently added to the Blockchain ledger.

Regarding the aforementioned "difficulty level" value, the Bitcoin system has its difficulty level adjusted such that these block confirmations occur approximately every 10 minutes[4]. Further, as the hash values are independent through time, the arrival of successfully mined blocks can be modelled of as memoryless, meaning that a computer having mined for two weeks gives it no advantage over one just starting to mine. Thus, from a mathematical perspective, the mining of blocks can be viewed as a Poisson process distributed with 10 minutes. This implies that each successive block arrives as an exponential distribution with 10 minutes. For an attacker who wishes to rewrite history t blocks into the past (i.e. change the t previously confirmed blocks), he must solve and broadcast t "replacement blocks" to overwrite these confirmed blocks, namely because the other nodes will only consider the longest chain as the valid one. Thus, changing history becomes exponentially more difficult with time (i.e. overwriting the last block is significantly easier than overwriting the 5th to last block), confirming the system security.

Of course, the natural question arises of why it is even worthwhile for the miners to purchase computers and pay for the electricity necessary to engage in such mining. In particular, the Bitcoin system resolves this dilemma by paying the miner with a reward upon successfully mining a block, which is currently 25 BTC. Every 210,000 blocks that are confirmed, this reward is halved (i.e. will soon become 12.5 BTC) [13]. In other words, miners are given monetary incentives to contribute to these computational tasks. In addition, the transactions themselves optionally contain "transaction fees," added by the Bitcoin users, which go directly to the miners. For example, a Bitcoin user A who wishes to pay B 0.5 BTC could pay an additional .02 BTC as a transaction fee that would go to the miner who successfully adds the block to the Blockchain that contains said transaction and *not* to B (i.e. A pays .52 BTC but B only receives .50 BTC). Evidently, the miners would be partial to adding transactions to the Blockchain that contain higher transaction fees, since they get more personal reward. This system described, where the transaction fees only make up a relatively small portion of the mining rewards, is referred to as the "block rewards" system, in contrast to the "transactional fee" system (described later in the background).

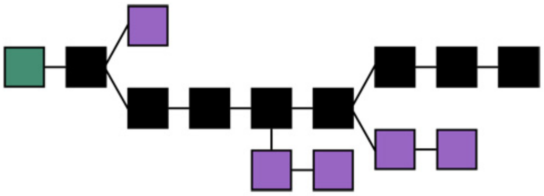

Fig. 1. An example of a Blockchain, where the purple blocks represent orphan blocks (i.e. not a part of the of longest chain) [10].

A complication that arises in this mining system is "if two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first" [11]. That is, since which transactions need to be included in successive blocks is not fixed, different mining rigs may potentially transmit differing *valid* blocks to the network simultaneously, in which case the protocol seems unclear/undefined per the above description. Clearly, it cannot be the case that both are added to the Blockchain, since there would likely be significant overlap between the two and may in fact be conflicting data transmitted in each. Thus, "in this case, they [the nodes] work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one" [11]. That is, each node adds the block it heard about first to its Blockchain but retains the other separately. The next block that is mined will build on top of one of these two, namely whichever of the two was heard first by the next successful miner (i.e. the one that is publicly accepted in the end is determined by the *subsequent* miner). The blocks that end up *not* being on the public Blockchain (i.e. the one on top of which the subsequent miner did *not* mine) is referred to as an *orphan block*.

With this basic foundation of Blockchain, we briefly step aside to discuss specifics of the Bitcoin and Ethereum implementations (more so the latter due to the focus on the former in the above presentation).

B. Bitcoin Specifics

As stated above, the Bitcoin system was largely presented in its specifics in the above presentation of Blockchain. In particular of note is the current exchange rate of $1,565.56 per BTC [2] and the transition to the transcation fee model. As mentioned previously, the Bitcoin system halves the block

[3] The exact proportion/size of this direct neighborhood set varies from Blockchain to Blockchain.

[4] Most of the discussion up to this point (aside from the length of the nonce) have been general discussions of the Blockchain protocol. However, this 10 minute separation is a property *specific* to Bitcoin. Namely, Ethereum's corresponding property is approximately 15 seconds.

reward for miners every 210,000 blocks, implying the last Bitcoin will be mined in 2140 [5]. However, well before that time (i.e. around 2025), it is anticipated that the primary incentive for miners will be the transaction fees, which clearly have relatively larger weights as rewards are halved. In doing so, many of the dynamics of miners change, since this system incentivizes including larger blocks (by number of transactions) more heavily because of the potential to reap more rewards. Although this is the case nowadays, the additional reward is negligible in comparison to the block reward.

C. Ethereum Specifics

Ethereum, while relying on the same Blockchain mining/confirmation system, has a fundamentally different end goal than Bitcoin. As previously described, the Ethereum network wishes to allow computers to run scripts in their Turing-complete scripting language in a distributed sense. That is, Ethereum essentially creates a distributed network of computers to run arbitrary scripts, which are then confirmed in the Blockchain as with transactions in the Bitcoin system. Similar to Bitcoin, upon mining blocks of history, miners receive a reward, although the currency in use in the Ethereum system is known as "ether," which is the second most valued cryptocurrency at an exchange rate of $87.86 [8]. Due to the fundamental differences, however, Ethereum does have a few unique aspects from Bitcoin, namely gas and the uncle block reward.

Gas technically serves the same role as transacation fees in the Bitcoin system; however, the way in which it is implemented differs. Namely, since Ethereum revolves around running scripts, there is a large concern of having users running (intentionally or not) malicious scripts, i.e. infinite loops. Thus, to discourage running such bogus scripts, the user must specify a "gas" when submitting their script to execute, which corresponds to a limit on how many instructions said script can execute. Different instructions use different amounts of gas, i.e. calculating a SHA-3 hash uses 20 gas [1]. Thus, for example, a user could say he is giving a script 2000 gas, meaning it will maximally execute instructions until 2000 gas has been used or the script terminates. Further, the gas has an associated cost in Ether, which is subject to the network, meaning that the equivalent of a transaction fee for Ethereum is the Amount of Gas Used times Gas Price.

The final major difference between Ethereum and Bitcoin is the treatment of orphan blocks, which are referred to as "uncle blocks" in Ethereum. In particular, the Bitcoin protocol does *not* give any reward to the miners if they find an orphan block. Ethereum, on the other hand, rewards a 75% of the block reward to miners of uncle blocks with the intention of "reduc[ing] centralization pressures, by reducing the advantage that well-connected miners have over poorly connected miners" [3]. Finally, we transition from this overall discussion of Blockchain, Bitcoin, and Ethereum to the specific focus of this paper: the selfish mining attack.

D. Selfish Mining

Selfish mining was an attack originally proposed by Eyal and Sirer in [9]. The essential idea behind this attack is getting the other miners in the network to waste their hashing (computational) power on invalid blocks while the attacker continues to mine on valid blocks, in turn effectively increasing his hashing power in the network[5]. To do so, the selfish miner mines as usual. However, after finding a valid block, rather than transmitting it to the rest of the network, he keeps it private and immediately returns to mining. If this attacking miner is successful in finding another block, he now has a private chain of length two. At this point, the attacker is in a *very* strong position compared to the rest of the network (i.e. once he has a private chain of size two), since if anyone discovers a block, the attacker can immediately release both his privately held blocks, which are guaranteed to be accepted by the remainder of the network, since they form a longer chain than just the single released block. Beyond a private length of two, the attacker only releases a single block and subsequently returns to mining to extend his private chain, since he is guaranteed that at some point later in the line of released blocks he will secure the longest chain, i.e. in releasing a private chain of length two.

For example, if the attacker has a private chain of length four and someone releases a block, he releases a block, such that there is now a block race. Thus, one of the two will be orphaned and the other accepted. However, as described above, the block is not confirmed to be in the longest Blockchain until the subsequent mine is completed. Thus, if someone else finds a valid block, the attacker will once again release his block, meaning the previous valid block that was not the attacker's will be invalidated (i.e. become an orphan) and there will be another block race. Finally, if another valid block is released, the attacker releases his private chain of length two, thus securing all the previous broadcasted blocks and reaping the entire cumulative reward.

The only downside of selfish mining is in the case where the attacker only has a private chain of length one, i.e. just after the first block has been mined. In this case, if someone else broadcasts a valid block, he must immediately release his block and hope that the network accepts his. If a sufficiently high proportion of the miners receive the attacker's transmitted block first, then it is likely the attacker's block will end up being added to the longest Blockchain, meaning he will receive the block reward. However, he does run the risk of having someone mining on top of the honest miner's block[6], in which case he will effectively have lost the potential block reward of one block.

As we also wish to perform some experimental analysis on selfish mining in both Ethereum and Bitcoin, empirical signs indicative of selfish mining are quite relevant. In particular, per the strategy defined above, when an attacker broadcasts his private chain, he will in turn create orphan blocks. Thus, having significantly higher count of orphan blocks is indicative of selfish mining attacks [6]. Similarly, since this strategy

[5] Although indirectly referred to before, explicitly the "hashing power" refers to the proportion of the total computational power a person owns of the entire network. That is, if miner A has 50% hashpower, his computers are doing half of all the SHA-2 hashing computations being performed in the network.

[6] An honest miner is simply anyone not engaging in selfish-mining in our case.

$f(\beta) : [0,1] \rightarrow [0,1]$	Depreciation factor as a function of selfish mining hashrate, namely β is the fraction of total hashrate going to selfish mining. The function value is how the exchange rate decreases to, i.e. .90 corresponds to a reduction in 10% of exchange rate.
α	Percent of the total network hashrate associated with the attacker
β	Percent of the total network hashrate going to selfish mining, necessarily $\leq \alpha$. That is, the attacker allocates $\alpha - \beta$ to simply honest mine
γ	Connectivity of the attacker to the remainder of the network, i.e. the probability the attacker's block will will the block race
$\mathcal{R}_{SM}(\alpha, \beta)$	Expectation for the revenue/reward mined by a selfish miner who has α of the total hashrate and β to selfish mining
$\mathcal{R}_H(\alpha, \beta)$	Expectation for the revenue/reward going to the honest miners when there is a selfish miner who has α of the total hashrate and β to selfish mining
$SM(\alpha, \beta)$	Expected proportion of total revenue/rewards going towards the attacker when he is with α of the total hashrate and β to selfish mining
$H(\alpha)$	Expected proportion of total revenue/rewards going towards an honest user with α of the total hashrate

TABLE I. VARIABLES AND MEANINGS

involves releases multiple blocks simultaneously, a closer distribution in block transmissions would be indicative of selfish mining attacks. Thus, these two empirical quantities are artifacts/evidence of an instance of selfish mining.

II. Selfish-Mining Theory

Our main novel contribution through this paper is studying under what circumstances it is in the interest of an attacker to engage in selfish mining. In particular, we wish to consider the reduction in value (exchange rate) of the cryptocurrency with an increase in selfish mining. The intuitive underpinning is that, if users of the Bitcoin system were to observe/detect instances of selfish mining, they would likely grow less confident in the system, thus resulting in a reduction of value of the system. Thus, we propose models to investigate these claims from a theoretical perspective and subsequently analyze them with experimental results obtained from the Bitcoin and Ethereum blockchains. These models were constructed in line with those proposed in [9] [4].

A. Bitcoin Block Reward Model

We formalize the model used to determine the circumstances under which a miner would wish to engage in selfish mining. Namely, we model the target function $f(\beta)$, which is the associated depreciation of value associated with β of the network hashpower going to selfish mining. As mentioned above, said depreciation is the result of a loss of trust in the network with a more obvious detectability of selfish mining in the network, meaning that $f(\beta)$ should decrease monotonically with a larger argument.

Thus, we wish to find under what circumstances we would have a greater reward through selfish mining as opposed to honest mining accounting for the reduction in valuation associated with selfish mining. Thus, we wish to find what $\alpha, \beta, f(\beta)$ result in:

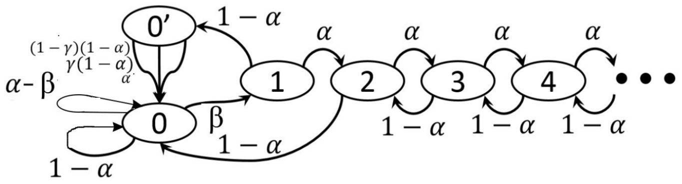

Fig. 2. Markov Model of private blockchain size, which continues ad infinitum towards the right, for the block reward model, adopted from the diagram in [9]

$$f(\beta)SM(\alpha,\beta) \geq H(\alpha)$$

$$f(\beta) \geq \frac{H(\alpha)}{SM(\alpha,\beta)}$$

$$f(\beta) \geq \frac{H(\alpha)}{\mathcal{R}_{SM}(\alpha,\beta)/(\mathcal{R}_{SM}(\alpha,\beta)+\mathcal{R}_H(\alpha,\beta))}$$

$$= H(\alpha)\left(1+\frac{\mathcal{R}_H(\alpha,\beta)}{\mathcal{R}_{SM}(\alpha,\beta)}\right) = \alpha\left(1+\frac{\mathcal{R}_H(\alpha,\beta)}{\mathcal{R}_{SM}(\alpha,\beta)}\right)$$

The final step follows because $H(\alpha) = \alpha$ by assumption. That is, we assume through this model that the proportion of hashrate owned by a user is equivalent to the probability he mines a block, meaning a miner with α hashrate proportion is expected to win α proportion of the blocks. Thus, to directly solve this explicitly, we wish to model both $\mathcal{R}_{SM}$ and $\mathcal{R}_H$, similar to how it was done in [9]. To explicitly calculate each, we can sum over the rewards associated with all possible states of the attacker's private chain. Specifically, representing the length of the private chain as S, we can break the reward function as:

$$\mathcal{R}_{SM}(\alpha,\beta) = \sum_{i=0}^{\infty} \mathbb{P}[S=i]\mathcal{R}_{SM}^{(i)}(\alpha,\beta)$$

Where $\mathcal{R}_{SM}^{(i)}(\alpha,\beta)$ is the reward associated with having a private chain of length i and $\mathbb{P}[S=i]$ the probability thereof. Thus, to calculate this, we model each separately. For the latter, we model the length of the private chain as a Markov chain and calculate the steady state probabilities. In doing this modelling, we introduce a supplementary $0'$ state, which corresponds to when two blocks are released simultaneously to system, namely the circumstance in which one block becomes orphaned and the other added to the longest chain, differing by time of arrival. In this analysis, γ of the network accepts the attacker's block, namely it transmits faster to a fraction γ of the network. Thus, the attacker's block will be considered a part of the network if either the attacker mines the subsequent block or if one of the miners in the γ fraction mine it. We formulate the following Markov Chain, with an explanation of the transitions and associated probabilities. Note that an honest miner will *always* release his block as soon as one is found:

- **State 0:**
  - ○ $\mathbb{P}_1[0 \rightarrow 0] = 1 - \alpha$: When an honest miner successfully mines the current block, namely since

  $1-\alpha$ is the hashrate of the rest of network excluding the selfish miner

  - $\mathbb{P}_2[0 \to 0] = \alpha - \beta$: The case where the hashrate of the attacker not going towards selfish mining succeeds in mining a block. In this case, the attacker simply broadcasts the block
  - $\mathbb{P}[0 \to 1] = \beta$: Corresponds to the case of the attacker finding a block with his hashrate dedicated to selfish mining, meaning this is the probability the attacker's private chain forming.

- **State 0':** In this state, there are two competing blocks with γ of the network having the attacker's block in the main chain and $1-\gamma$ having the honest miner's.

  - $\mathbb{P}_1[0' \to 0] = (1-\gamma)(1-\alpha)$: Assuming the attacker does *not* mine the subsequent block (occurring with probability $1-\alpha$), the successful miner has a $1-\gamma$ chance of mining on top of the honest miner's block
  - $\mathbb{P}_2[0' \to 0] = \gamma(1-\alpha)$: Similarly, if the attacker does *not* mine the subsequent block, there is γ chance the successful miner will be mining on the attacker's block
  - $\mathbb{P}_3[0' \to 0] = \alpha$: In this case, the attacker finds the subsequent block, occurring with probability α, releasing it immediately.

- **State 1:**

  - $\mathbb{P}[1 \to 0'] = 1-\alpha$: If an honest miner finds or broadcasts a block, the attacker broadcasts his block to create a block race
  - $\mathbb{P}[1 \to 2] = \alpha$: Once the attacker has one block in his chain, he puts all of his hashrate power towards selfish mining, meaning this transition corresponds to the possibility of increasing his private chain.

- **State 2:**

  - $\mathbb{P}[2 \to 0] = 1-\alpha$: If an honest miner finds or broadcasts a block, the attacker broadcast his entire private chain, since this guarantees winning the race and being added to the Blackchain
  - $\mathbb{P}[2 \to 3] = \alpha$: Explanation follows equivalently from state 1 transition

- **State $k \geq 3$:**

  - $\mathbb{P}[k \to k-1] = 1-\alpha$: If an honest miner finds or broadcasts a block, the attacker broadcast *only one*, since he are still in a position where he holds a private lead over the public ledger (as previously described)
  - $\mathbb{P}[k \to k+1] = \alpha$: Equivalent to state 1 transition

Taking these state transitions, we now formulate a system of equations to determine steady state probabilities as follows:

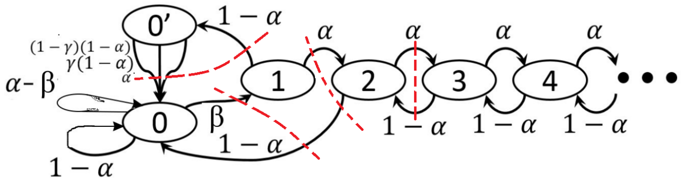

Fig. 3. To do analysis, we introduce the following cuts, from which we get the systems of equations used to analytically solve for the steady state probabilities.

$$
\begin{aligned}
(1-\alpha)p_1 - p_{0'} &= 0 \\
\beta p_0 - (1-\alpha)p_1 - (1-\alpha)p_2 &= 0 \\
\alpha p_1 - (1-\alpha)p_2 &= 0 \\
\forall k \geq 2 : \alpha p_k - (1-\alpha)p_{k+1} &= 0 \\
\sum_{k=0}^{\infty} p_k + p_{0'} &= 1
\end{aligned}
$$

Solving for these, we obtain the following probability expressions:

$$
\begin{aligned}
p_0 &= \frac{(1-2\alpha)}{\beta(1-\alpha)(1-2\alpha) + (1-2\alpha) + \beta(1-\alpha)} \\
p_{0'} &= \frac{\beta(1-\alpha)(1-2\alpha)}{\beta(1-\alpha)(1-2\alpha) + (1-2\alpha) + \beta(1-\alpha)} \\
p_1 &= \frac{\beta(1-2\alpha)}{\beta(1-\alpha)(1-2\alpha) + (1-2\alpha) + \beta(1-\alpha)} \\
p_k = \left(\frac{\alpha}{1-\alpha}\right)^{k-1} &\frac{\beta(1-2\alpha)}{\beta(1-\alpha)(1-2\alpha) + (1-2\alpha) + \beta(1-\alpha)}
\end{aligned}
$$

Thus, we have expressions that model the probabilities of being in each state. To complete the overall calculation of interest from before, we must now model $\mathcal{R}_{SM}^{(i)}(\alpha, \beta)$ for each state i. The reasoning behind these are as follows:

- $\mathcal{R}_{SM}$
  - **State 0**
    - $0 \to 0$ If the attacker successfully mines through an honest mean, he will immediately broadcast the block, giving 1 block reward
  - **State 0'**
    - $0' \to 0$ If the attacker successfully mines another block, he will immediately broadcast it, meaning he will get both said block *and* the one being contested, giving 2 block rewards
    - $0' \to 0$ If a miner with the attacker's block on their main chain mines successfully, he will broadcast it (being an honest miner), giving the attacker 1 block reward
  - **State 1**
    - $1 \to 0'$ If the attacker hears about an honest miner's block, he will broadcast his own immediately, but will *not* be rewarded with anything,

since the network will be in the middle of a block race (i.e. the network will be unresolved as to which block to add to the Blockchain)

  - **State 2**
    - $2 \to 0$ In hearing about an honest miner's block, the attacker broadcasts both his private blocks, thus guaranteeing 2 block rewards
  - **State** $k \geq 3$
    - $k \to k-1$ In hearing about an honest miner's block, the attacker broadcasts only one private block to maintain a private lead, thus only obtaining 1 block reward

- $\mathcal{R}_H$
  - **State 0**
    - $0 \to 0$ If an honest miner successfully mines through an honest mean, he will immediately broadcast the block, giving 1 block reward
  - **State 0'**
    - $0' \to 0$ If the honest miner successfully mines on the attacker's block, he will immediately broadcast, giving 1 block reward
    - $0' \to 0$ If the honest miner successfully mines on the honest block, he will also immediately broadcast, giving 2 block rewards

Thus, we now return to the original formulation of interest:

$$\begin{aligned}\mathcal{R}_{SM} &= p_0\mathbb{P}_2[0 \to 0] + p_{0'}(\mathbb{P}_2[0' \to 0] + 2\mathbb{P}_3[0' \to 0]) + \\ &2p_2\mathbb{P}[2 \to 0] + \sum_{k=3}^{\infty} p_k\mathbb{P}[k \to k-1] = p_0(\alpha - \beta) + \\ &p_{0'}(\gamma(1-\alpha) + 2\alpha) + 2p_2(1-\alpha) + \sum_{k=3}^{\infty} p_k(1-\alpha)\end{aligned}$$

$$\begin{aligned}\mathcal{R}_H &= p_0\mathbb{P}_1[0 \to 0] + p_{0'}(2\mathbb{P}_1[0' \to 0] + \mathbb{P}_2[0' \to 0]) \\ &= p_0(1-\alpha) + p_{0'}(2(1-\gamma)(1-\alpha) + \gamma(1-\alpha)) \\ &= p_0(1-\alpha) + p_{0'}(2-\gamma)(1-\alpha)\end{aligned}$$

Which we can then plug into the original formula for the empirical results. Thus this block reward is completed and now do so for the transaction fee model.

B. Bitcoin Transaction Fee Model

We now consider a similar analysis in the case of the transactional fee model. This analysis is done in line with the results found in [4], though we once again are interested in finding the devaluation boundary function $f(\beta)$. In line with this, the identical procedure as before is performed, namely in segmenting the analysis into first determining stationary probabilities and subsequently considering rewards for each state. The main difference arises in the the 0 state. Namely, unlike in the block rewards case, engaging in selfish mining now becomes dependent on how long it has been since the previous block was mined, since effectively the value of a block increases with time, due to the greater number of transactions that are included in the block. Specifically, the model is constructed in line with [4], where we assume that if it has been $\leq \lambda$ time since the previous block was mined, it is worth the attacker to selfish mine whereas if $t > \lambda$, he should just release the block, since if another honest miner successfully mines, the potential loss is too high.

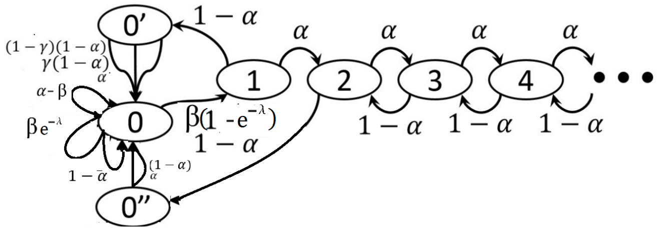

Fig. 4. Markov Model of private blockchain size, which continues ad infinitum towards the right, for the transaction fee model, adopted from the diagram in [4]

The two changes of note from the previous Markov model (i.e. the one for block rewards model), aside from the transition probabilities, are the introduction of the 0" state and the "continuous nature" of the 0 state. The former simply represents the state when the attacker releases his private chain from state 2. It is assumed that, from this state, the attacker will mine his next block honestly, which was assumed for sake of simplifying the analysis and was empirically found to not affect results significantly [4]. The latter corresponds to the fact that state 0 now has a different reward based on the time that has elapsed since the last block was transmitted, meaning we can essentially view this as a state $(0, t)$ for all $t \in \mathbb{R}^+$. That is, $p_0(t) = p_0 e^{-t}$, since the transactions and blocks successively arrive with exponential distribution; however, this only becomes relevant when later calculating the reward functions:

$$\begin{aligned}p_{0'} + p_{0''} - \beta(1-e^{-\lambda})p_0 &= 0 \\ (1-\alpha)p_2 - p_{0''} &= 0 \\ (1-\alpha)p_1 - p_{0'} &= 0 \\ \forall i > 2 : \alpha p_i &= (1-\alpha)p_{i+1}\end{aligned}$$

Solving this Markov model, we obtain:

$$\begin{aligned}p_1 &= \frac{\beta(1-e^{-\lambda})(1-2\alpha)}{\beta(1-e^{-\lambda})(2-3\alpha) + (1-2\alpha)} \\ p_0 &= \frac{1-2\alpha}{\beta(1-e^{-\lambda})(2-3\alpha) + (1-2\alpha)} \\ p_{0'} &= \frac{\beta(1-e^{-\lambda})(1-\alpha)(1-2\alpha)}{\beta(1-e^{-\lambda})(2-3\alpha) + (1-2\alpha)} \\ p_{0''} &= \frac{\alpha\beta(1-e^{-\lambda})(1-2\alpha)}{\beta(1-e^{-\lambda})(2-3\alpha) + (1-2\alpha)} \\ \forall i \geq 2 : p_i &= \left(\frac{\alpha}{1-\alpha}\right)^{i-1} \frac{\beta(1-e^{-\lambda})(1-2\alpha)}{\beta(1-e^{-\lambda})(2-3\alpha) + (1-2\alpha)}\end{aligned}$$

The rewards too were calculated for both the honest and selfish cases and are explicitly provided in the Appendix, whose results are:

$$\begin{aligned}\mathcal{R}_{SM}(\alpha,\beta) = p_0(\lambda\beta e^{-\lambda} + \alpha\beta(1-e^{-\lambda}(1-\lambda))+\\ (1-\alpha)\beta(1-e^{-\lambda}(1-\lambda))(2\alpha+\gamma(1-\alpha))) + p_0\beta e^{-2\lambda}+\\ p_{0'}\left(\gamma(1-\alpha)+2\alpha]\right) + \alpha p_{0''} + (1-\alpha)p_2 + (1-\alpha)\sum_{k=3}^{\infty} p_k\\ \mathcal{R}_H(\alpha,\beta) = (1-\alpha)p_0+\\ p_{0'}\left(2(1-\gamma)(1-\alpha)+\gamma(1-\alpha)]\right) + (1-\alpha)p_{0''}\end{aligned}$$

From which empirical results for the boundary function $f(\beta)$ can be obtained by plugging into the original inequality.

C. Ethereum Model

Each of the above models, namely the block reward and transaction fee, were modelled largely independent of the underlying cryptocurrency. That is, these are generic models for a Blockchain system in which the transaction fees respectively are not and are significant incentives. As a result, the latter model is more conducive to modelling the Ethereum Blockchain, since miners are heavily incentivized by the gas. The only modification that is introduced to the previous model is to account for the reward Ethereum miners get for mining uncle blocks. Namely, in the transition from the $0' \to 0$ state, rather than receiving no reward if an honest miner mines on top of another honest miner's block from the block race, he obtains some fractional reward ρ. It is clearly of interest to find what fraction ρ deincentivizes engaging in selfish mining for a particular fixed devaluation function. That is, finding how uncle blocks are valued in comparison to blocks on the longest chain will affect when miners would engage in selfish mining. That is, the reward state for state 0' would be updated to:

$$\begin{aligned}\mathcal{R}_{SM}^{(0')}(\alpha,\beta) &= p_{0'}\left(\rho\mathbb{P}_1[0'\to 0] + \mathbb{P}_2[0'\to 0] + 2\mathbb{P}_3[0'\to 0]\right)\\ &= p_{0'}\left(\rho(1-\gamma)(1-\alpha) + \gamma(1-\alpha) + 2\alpha]\right)\end{aligned}$$

Substituting this into the $\mathcal{R}_{SM}$ produces the model for the Ethereum Blockchain.

III. Selfish-Mining Empirical Analysis

Having developed these models, it follows that studying expected trends and relations between variables be the next natural step. That is, looking at each of the models above to determine what the natural $f(\beta)$ necessary devaluation function appears to be in each circumstance, namely for the Bitcoin Block Reward, Bitcoin Transaction Fee, and Ethereum models.

A. Bitcoin Block Reward

Here are the trends per the Bitcoin block reward model.

1) State Probability Distributions: Below are the probability distributions of the states as functions β across various fixed α values. Clearly, since the value of γ is independent of the state probabilities, this was not included in the below result presentations. These are all the steady state probabilities predicted through the solved Markov Model. Note that, through the below figures, the following conventions are followed for the point markers with the exception of those in the state probability figures. The color represents a fixed value of γ, and the shape represents the type of miner being shown in the figure, where the R_{SM} represents (as used in the analyses above) the expected rewards for a selfish miner and R_H that of an honest miner:

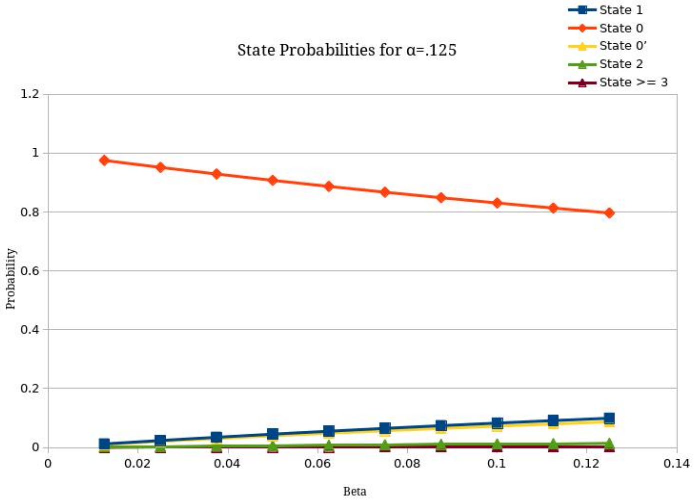

Fig. 5. State probabilities across β with $\alpha = .125$

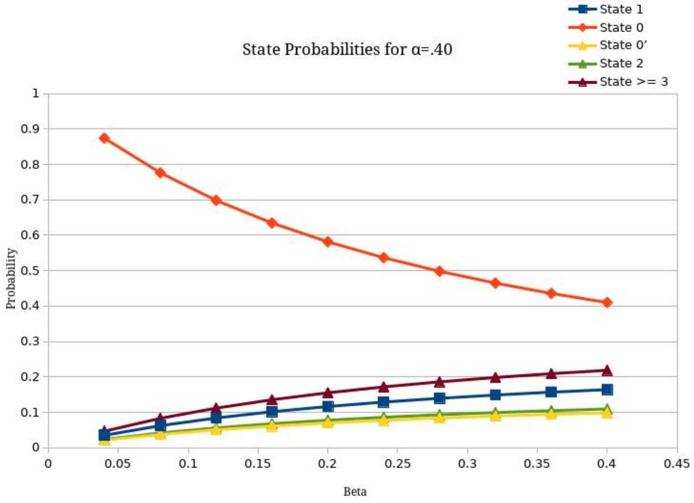

Fig. 6. State probabilities across β with $\alpha = .40$

2) Expected Rewards: Below are the value of the reward functions (both for selfish miners and honest miners) as functions β across various fixed α values:

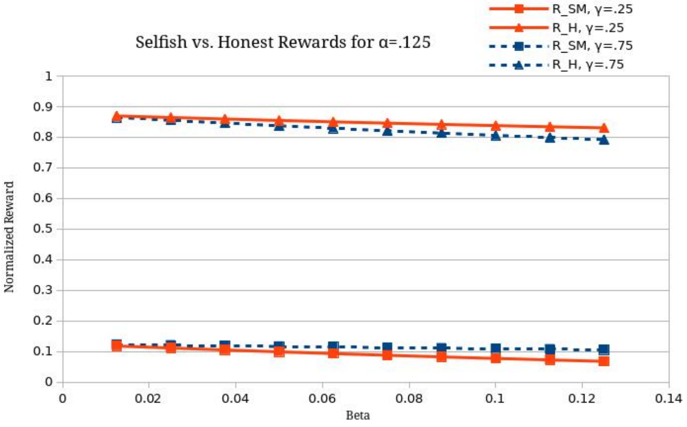

Fig. 7. Expected selfish and honest miner revenue/reward across β with $\alpha = .125$

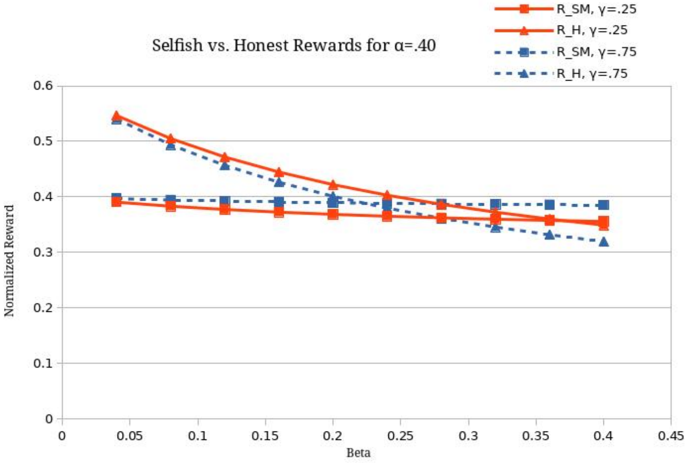

Fig. 8. Expected selfish and honest miner revenue/reward across β with $\alpha = .40$

3) Devaluation Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining as 2D figures, i.e. slices with fixed values of α, γ.

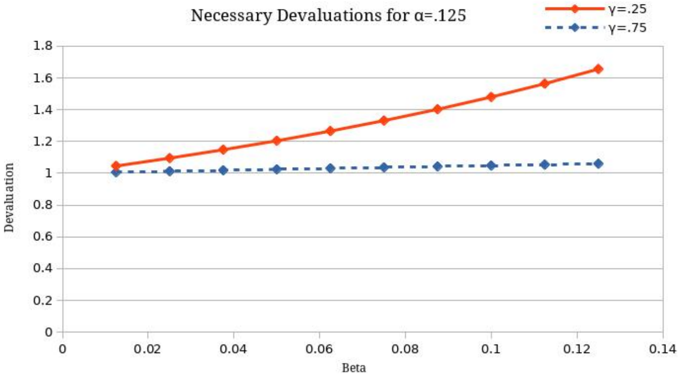

Fig. 9. Necessary devaluation across β with $\alpha = .125$

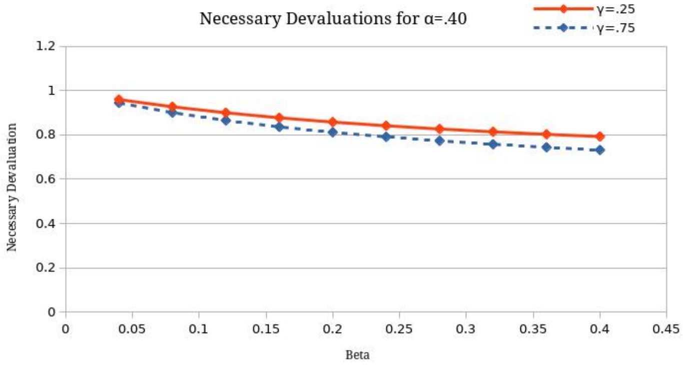

Fig. 10. Necessary devaluation across β with $\alpha = .40$

4) Necessary Devaluation 3D Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining in a 3D figure, namely as a function of both α and β:

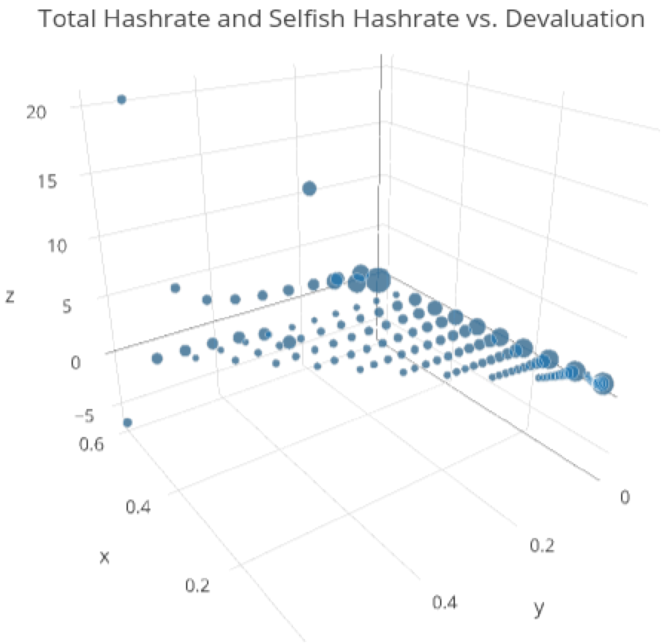

Fig. 11. Necessary devaluation as a function of both α, β. The size is used to pronounce the z value. The x, y, z axes respectively correspond to α, β and the necessary devaluation (i.e. $f(\beta)$)

B. Bitcoin Transaction Fee

Here are the trends per the Bitcoin transaction fees model. Note that, with the introduction of the new λ parameter, similar labelling conventions were applied in the below figures, except that colors were extended to cover representing a particular λ value rather than γ from above.

1) State Probability Distributions: Below are the probability distributions of the states as functions β across various fixed α, λ values:

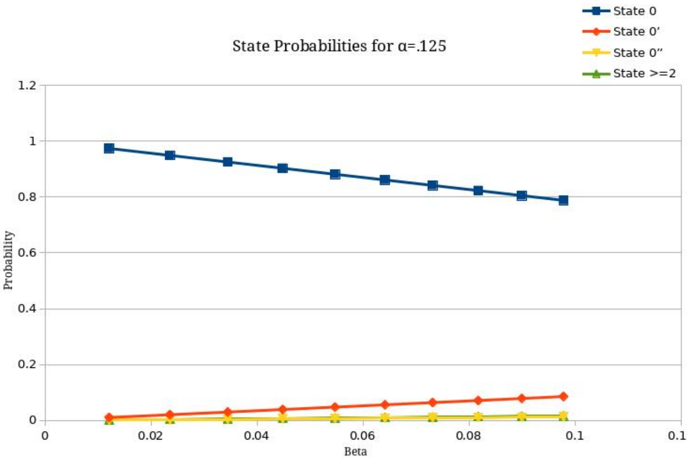

Fig. 12. Transaction model state probabilities across β with $\alpha = .125$

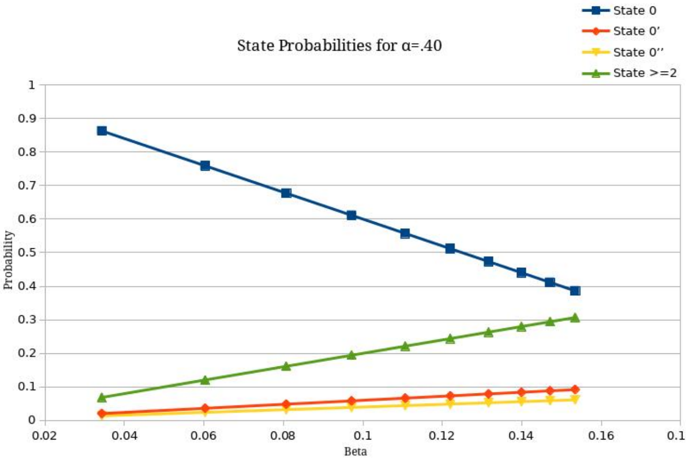

Fig. 13. Transaction model state probabilities across β with $\alpha = .40$

2) Expected Rewards: Below are the expected rewards of the selfish and honest miners as functions β across various fixed α, λ values:

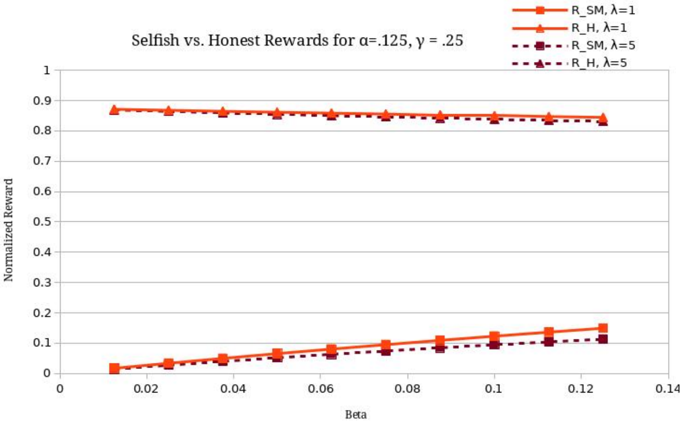

Fig. 14. Expected transaction model expected selfish and honest miner revenue/reward across β with $\alpha = .125$

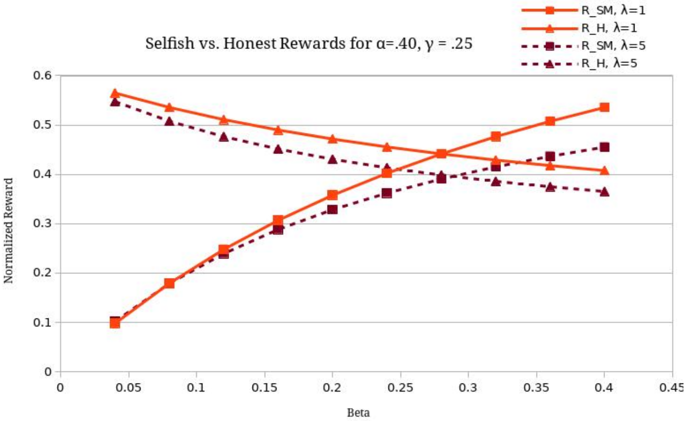

Fig. 15. Expected transaction model expected selfish and honest miner revenue/reward across β with $\alpha = .40$

3) Devaluation Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining as 2D figures, i.e. slices with fixed values of α, γ, and λ:

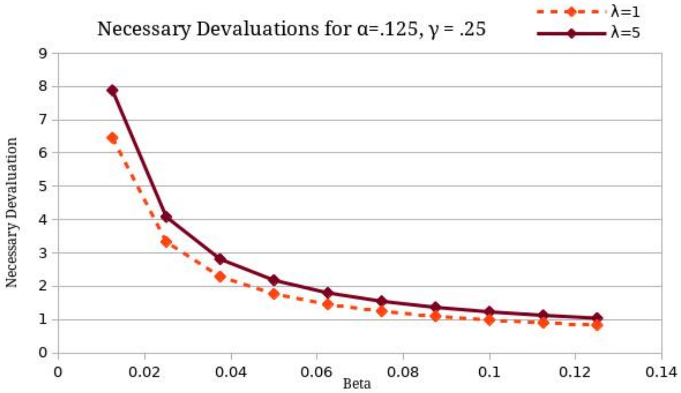

Fig. 16. Necessary devaluation for the transaction model across β with $\alpha = .125$

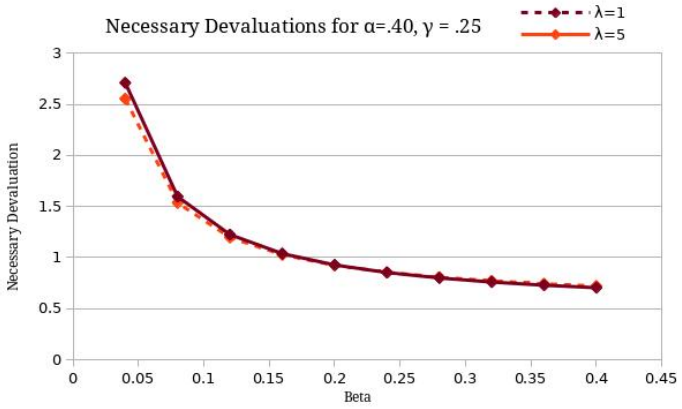

Fig. 17. Necessary devaluation for the transaction model across β with $\alpha = .40$

4) Necessary Devaluation 3D Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining in a 3D figure, namely as a function of both α and β, holding λ fixed:

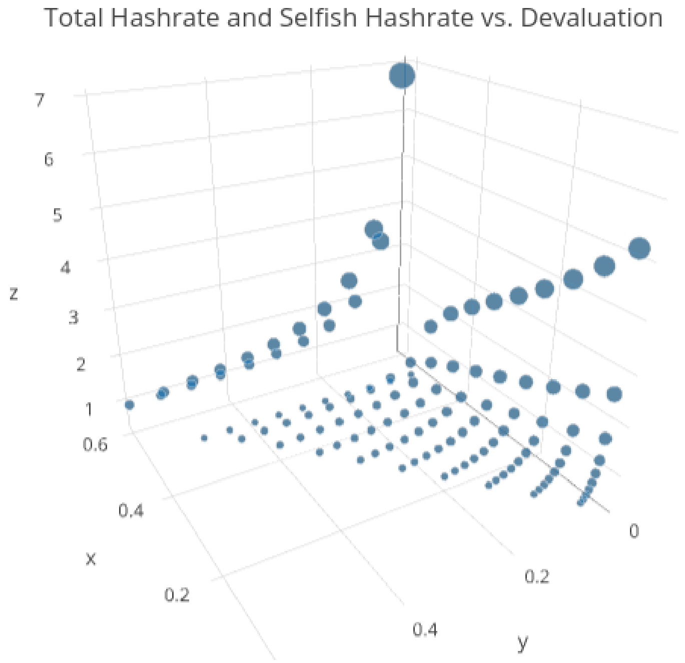

Fig. 18. Necessary devaluation as a function of both α, β for the transaction model, with x, y, z axes respectively correspond to α, β and the necessary devaluation (i.e. $f(\beta)$)

C. Ethereum Model

Here are the trends per the Ethereum Blockchain model. However, since the model transitions are identical to those in the transaction fee model presented in the case of Bitcoin, the probability state graphs and numerical results are identical to those from above. Thus, the inclusion of the ρ factor solely had an impact on the reward value and devaluation boundary, depicted in the sections that follow, which is once again distinguished in the figures below through color.

1) Expected Rewards: Below are the expected rewards of the selfish and honest miners as functions β across various fixed α, λ, γ values. Since only the behavior as a function of ρ was of interest in this version of the model, namely being the only modification from the transaction model to the Ethereum model, the graphs below have fixed α, λ, γ and multiple ρ:

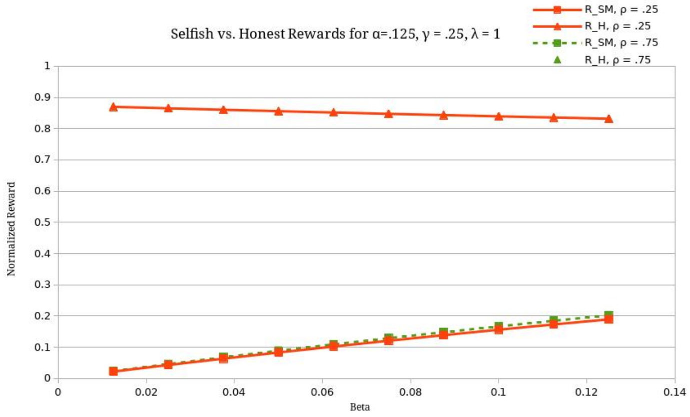

Fig. 19. Expected expected selfish and honest miner revenue/reward across β with $\alpha = .40$ in the Ethereum network

2) Devaluation Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining as 2D figures, i.e. slices with fixed values of α, γ, and λ. The graphs pictures below follow the same presentation as those in the previous section. For fixed α, λ and multiple values of ρ:

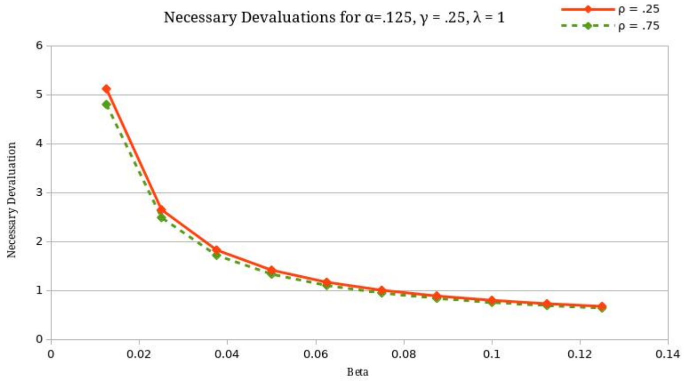

Fig. 20. Necessary devaluation for the transaction model across β with $\alpha = .40$ in the Ethereum network

3) Necessary Devaluation 3D Boundary: Below are the value of the necessary devaluations to demotivate use of selfish mining in a 3D figure, namely as a function of both α and β, holding ρ, λ fixed:

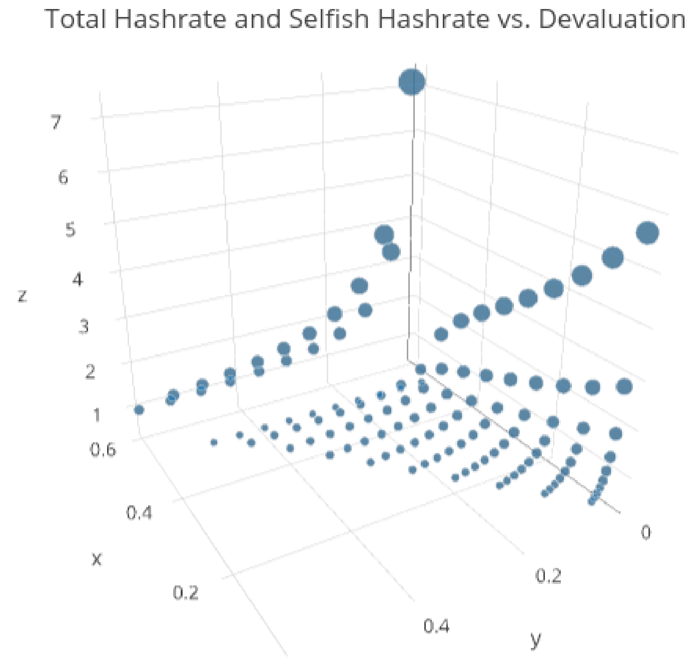

Fig. 21. Necessary devaluation as a function of both α, β for the Ethereum model, with x, y, z axes respectively correspond to α, β and the necessary devaluation (i.e. $f(\beta)$)

IV. Experimental Devaluation Findings

As discussed previously, while the search for selfish mining was fruitless up to 2014 per [6], searching for outlier instances in the number of orphan blocks and median block are sufficient for the purposes herein. That is, while there *may* be no instance of selfish mining taking place in the network, if the other Bitcoin users *believe* there is selfish mining taking place, there will be an associated decrease in valuation. Namely, there is no difference for users if there are actual instances of selfish mining resulting in the associated symptoms or if the symptoms happen to appear and coincide by chance. Further, it was assumed that any associated shifts in valuation caused by these properties would materialize a few days after they are observed, i.e. the number of orphan blocks and separation between block transmission are *lagging* indicators of the exchange rate. Thus, by studying these instances from the compiled Blockchain charts of [2] and similarly for Ethereum from [8], the following devaluations were observed:

A. Bitcoin Selfish Detection

Through figures 22, 23, and 24, the "indicator" put on the axes refers to a normalized variable defined to be the sum of the normalized number of orphan blocks and inverse of block time separation, since the lesser time separation is indicative of instances of selfish mining. That is, we have:

$$Indicator = ||\#_{blocks}|| + ||\frac{1}{T_{sep}}||$$

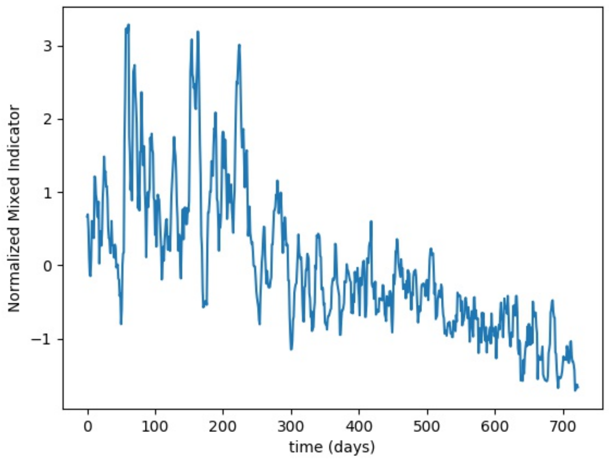

Fig. 22. Indicator variable as a function of time

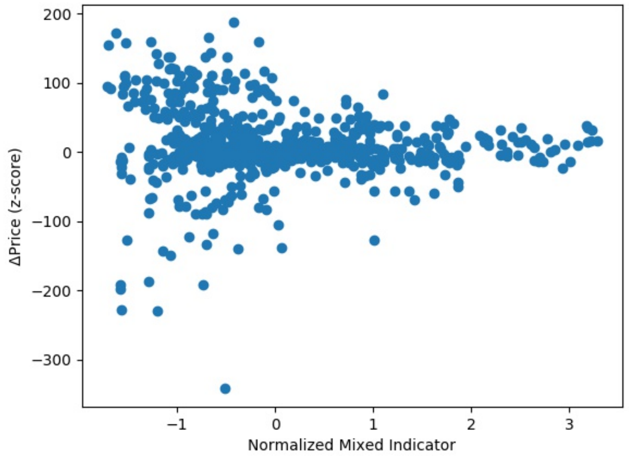

Fig. 23. Change in price as a function of the indicator variable. The change of price is $\Delta p = p_t - p_{t-1}$ for each unit of time t

B. Ethereum Selfish Detection

The identical graphs and analyses were conducted for Ethereum, whose graphs are 25, 26, and 27:

V. Discussion

From the above, the main points of discussion deal with the sensitivity of the devaluation function, which was the end result of main interest, to the various parameters, though the specific parameters that were different across the various models. That is, the devaluation function was studied in different contexts per the block reward model, transaction model, and Ethereum model.

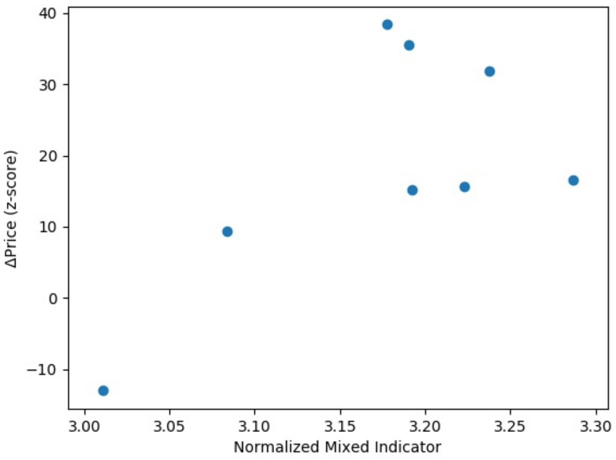

Fig. 24. These are the "outlier" values from the above the graph above. Namely, all the points that were outliers in the indicator variable and their associated prices.

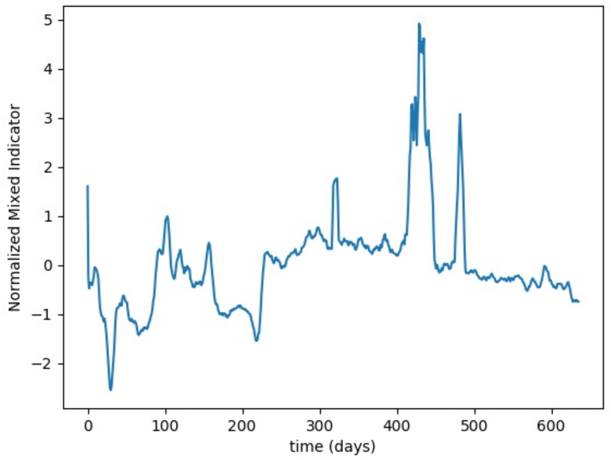

Fig. 25. Indicator variable as a function of time in the Ethereum network

A. Block Reward Discussion

The main unique point of variation of this model, aside from the clear relation between α, β, was studying γ. Regarding the probability states, from figures 5, 6, there is a clear trend toward the "higher states," i.e. those $n \in \mathbb{Z} \geq 1$, with increasing α, β. For a fixed α, increasing β is expected to result in a greater probability in "higher states" as the selfish miner now, for every block he mines, has a greater probability of stashing it as a private chain rather than broadcasting it. Further, for an increased overall hashrate (α), it is evident that the probability of having a private chain is greater than that for the same β and lower hashrate, as expected by there being an increase in probability of mining a block with a greater hashrate.

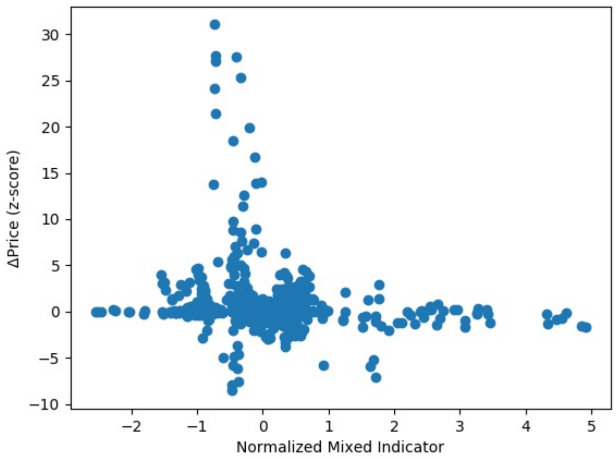

Fig. 26. Change in price as a function of the indicator variable. The change of price is $\Delta p = p_t - p_{t-1}$ for each unit of time t in the Ethereum network

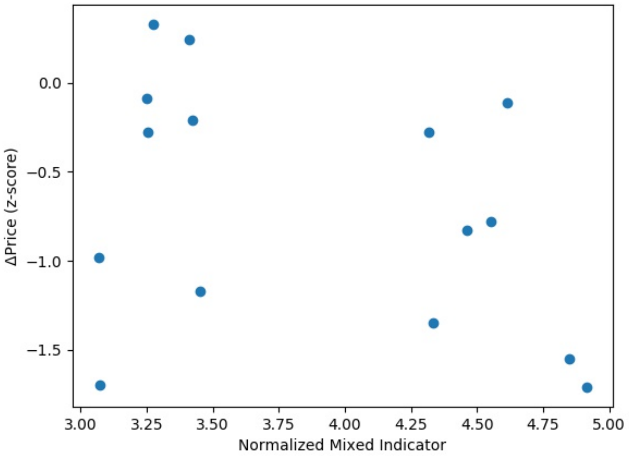

Fig. 27. These are the "outlier" values from the above the graph above. Namely, all the points that were outliers in the indicator variable and their associated prices in the Ethereum network.

Regarding the rewards function, from figures 7, 8, there was a clear shift in sensitivity to β per the level of α. Namely, for a lower α, shifting the β value had little effect on the expected reward of either the honest or selfish miner, likely because the attacker's hashrate being so low results in low probabilities of mining any block generally, equating to largely staying in state 0. Thus, changing the amount dedicated to selfish mining would not change the expected reward, since the β dependence of the miner's behavior is conditioned on him having blocks to work with. That is, the miner having no blocks make the decision to transmit to the network vacuous. In the $\alpha = .40$ case, on the other hand, there is a clear intuitive trend as a function of β. That is, with increasing β, the the honest miner's expected rewards sharply decreases while that of the selfish miner is relatively constant. Clearly, the intersection thereof is indicative of at what point it would be of interest to engage in selfish mining. This intersection shifts to the left (i.e. towards lower β) with increasing values of γ, meaning that, for greater expected return on a single private block, the miner should be more willing to engage at least a small fraction of power towards selfish mining.

The trend in γ, however, is consistently minimal across the different values of α, namely that a greater γ results in a slight increase in the expected reward of the selfish miner and corresponding dip in that for the honest miner. This is again a natural reaction, seeing that an increasing γ results in the miner effectively gaining more in the case of having a private block chain of length one, namely a single hidden block.

The expected trends were not as clearly defined for the devaluation function, with the increasing of β having offsetting effects, namely the increased expected number of mined blocks but diminishing value of each. However, per the results in figures 9, 10, it becomes clear that the dependence is *very* sensitive to the total α a miner has. Note that, although the *de*valuation largely only has a direct interpretation for values ≤ 1[7], those > 1 can qualitatively be seen as representing the extent to which it is *not* in favor of the miner to engage in selfish mining. Specifically, $f(\beta) = 2$ vs. $= 5$ cannot be directly compared quantitatively but qualitatively they signify increasing decentivization to selfish mine.

Thus, returning to the results, for low α, i.e. $\alpha = .125$, the miner is discouraged from putting any of his hashrate power towards selfish mining. That is, the devaluation is expected to grow as a function of β, meaning it best for weak miners to simply publish each block they find. This is presumably because any blocks they *do* find will very likely be followed by a block found by *another* miner. That is, for sufficiently low overall hashrates, miners are not expected to consecutively mine multiple blocks, meaning it is unlikely they would get any private chain longer than one. This makes the situation of low α heavily biased on the miner's ability to win block races, explaining the strong sensitivity of the devaluation function on γ. Namely, for sufficiently high γ, it almost becomes in favor of the miner to allocate some power towards selfish mining, although $\gamma = .75$ is not quite enough.

As for high α, the miner is expected to get long private blocks often being able to block multiple blocks consecutively, meaning that having more power dedicated to selfish mining is beneficial. That is, for these high hashrates, the miner is encouraged to expend all his resources towards selfish mining. Unlike the previous case, since the miner is likely not to spend time in the state of a single private block, there is only slight sensitivity to γ, though its effect is identical to that of the low α case.

B. Transaction Discussion

Unlike the previous analysis, that considered here is with respect to the newly introduced λ parameter, representing the cutoff for the modified selfish mining scheme, wherein the miner simply releases the mined block if it is discovered after $\Delta t = \lambda$ from the prior block. The state probabilities, from figures 12, 13, are relatively similar to those discovered in the previous analysis. The main difference is the largely linear relation between β and state probabilities for high α unlike the convex and concave shapes for the corresponding graph in the previous section.

The rewards functions, from figures 14, 15, for low α were also very similar to those mapped in the previous section, presumably because, once again, this low α corresponds to the miner generally mining few blocks, meaning any flexibility he has in behavior is largely flattened out by its insignificance in the overall result. For high α, however, there was some deviation in the rewards. Both scenarios depicted how the expected rewards for honest miners tended negatively as selfish miners allocated more of their resources towards selfish mining; however, unlike in the previous case, the transaction model revealed an increase in expected returns for selfish miners with increasing power allocation (largely constant in the previous model). This difference is likely attributed to the fact that, in the transaction model, the reward associated with the 0 state is dependent on β. Namely, β determines the behavior of directly releasing blocks vs. stashing them, meaning it has a significantly greater impact on the expected reward than it did in the previous case, in which it solely affected the extent of transitioning from state 0 to 1.

Further, the impact of λ is greater for larger α. λ appears to reduce the effectiveness of selfish mining. That is, if the cutoff of where an attacker should behave "honestly" is increased, then he gains less by engaging in selfish mining. In other words, the selfish miner, from solely an expected rewards viewpoint, would feel incentivized to reduce the cutoff time span between engaging in selfish mining and simply broadcasting discovered blocks.

The devaluation functions, from figures 16, 17, seem to be the main feature starkly in contrast with that from the previous section. Namely, unlike the previous case, there is little difference in the shape and trend of the devaluation functions in the cases of low and high α values. Namely, the hashrate, while change magnitudes, does *not* affect how $f(\beta)$ grows with increasing β. Specifically, in both cases, the necessary devaluation grows becomes lesser for for growing β, meaning for a fixed α the miner is encouraged to push all his resources towards selfish mining. The difference for the low α case is likely the result of the refinement in strategy accounting for time elapsed, making selfish mining even more versatile of a technique than was the case in its vanilla inception.

C. Ethereum Discussion

Unlike the previous model, this Ethereum model was not fundamentally different from the transaction fee model, in that the Markov Model was identical. Thus, the main point of discussion was the reward and devaluation functions, which capture the same overall conclusion. Namely, for the Ethereum network, the general trends as functions of β are identical to those visible in the transaction fee network. Namely,

[7] Namely indicating what relative drop in exchange rate marks the threshold for when a miner should engage in selfish mining or not

the shapes of the curves from figures 19, 20 are identical; the numerical values associated with the curves, however, differ. Specifically, comparing the corresponding curve from the transaction model, the devaluation result suggests that it becomes in favor of the miner to engage in selfish mining for $\beta \approx .06$ whereas the Bitcoin model is $\beta \approx .10$. Thus, it should be more the case the miners are incentivized to selfish mine in the Ethereum network than the Bitcoin network, due to the encouragement of uncle blocks. Surprisingly, however, the extent to which ρ affects the shape is not significant for relatively large β. This is likely because, for large β, miners are expected to broadcast their blocks (per the transaction selfish mining model), preventing them from reaching state 1, meaning there is no possibility of reaching a block race state, unlike for low β. Since ρ only affects the expected rewards for this block race outcome, there is a greater variance in devaluation for low β.

D. Experimental Discussion

From the data collected on the Bitcoin and Ethereum networks, it seems apparent that there were instances where both the number of orphan blocks and block broadcast time gaps spiked in a direction indicative of a selfish mining attack[8]. By looking at the outliers in figures 27, 24, namely the graphs in which solely those points for which the indicator was abnormally high were graphed as functions of the relative change in price, it becomes abundantly clear that there is no evidence of negative correlation. That is, there is no indication that signs suggestive of selfish mining are associated with a decrease in exchange rate. In terms of the model, this simply means that, on average, $\hat{f}(\beta) = 1$ in reality.

This, however, is quite a consequential outcome, in that it suggests that miners should be more willing to engage in selfish mining in any scenario illustrated previously for which $f(\beta) \leq 1$. Namely, for a sufficiently powerful miner, the natural course of action should be to *always* engage in selfish mining. This, however, is *not* conducive to stable decentralized systems such as Bitcoin or Ethereum, since they rely fully on the miners to maintain order without a single authority. With all of them being heavily encouraged to engage in selfish mining, even more so in the case of the Ethereum network, there is little "communal voting" captured by the mining process. In other words, due to the lack of punishment for selfish mining, the Bitcoin and Ethereum expose a gaping source of instability that can be easily exploited, especially with the strong mining power possible through mining pools.

E. Conclusion

Thus, through this paper, the instances in which it would be in favor of miners to engage in selfish mining vs. honest mining was expanded upon through the inclusion of a devaluation function. Further, through empirical evidence, it became abundantly clear that miners should engage in selfish mining should they wish to maximize personal gain if the Ethereum and Bitcoin systems' natural responses to selfish mining remain as they currently are. In turn, future studies may wish to tackle how to adjust the Ethereum protocol to discourage selfish mining while still promoting the general user to mine, which was the primary motivator for introducing uncle rewards. Similarly, future studies may expand upon why selfish mining has not been explicitly detected in the past, which can potentially be attribute to the opportunity risk associated with not releasing a block immediately after obtaining it in the network. These studies may further explore expansions from this model to mining pools, namely seeing if the introduction of a group mining dynamic would greatly affect the results of the study.

[8] As stated before, whether or not these truly instances of selfish mining occurring in the network is irrelevant, since the analyses conducted herein is independent of that fact

References

[1] "Account Types, Gas, and Transactions." Ethereum Homestead. N.p., n.d. Web. 07 May 2017.

[2] "Bitcoin Charts & Graphs." *Blockchain*. N.p., n.d. Web. 07 May 2017.

[3] Buterin, Vitalik. "Uncle Rate and Transaction Fee Analysis." *Ethereum Blog*. N.p., 31 Oct. 2016. Web. 07 May 2017.

[4] Carlsten, Miles, et al. "On the instability of bitcoin without the block reward." *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 2016.

[5] "Controlled Supply." *Bitcoin Wiki*. N.p., n.d. Web. 07 May 2017.

[6] Cui, Fangyang. *Detecting Selfish Mining in Bitcoin*. Thesis. 2015. N.p.: n.p., n.d. Print.

[7] "Difficulty." *Bitcoin Wiki*. N.p., n.d. Web. 07 May 2017.

[8] "Ethereum Blockchain Explorer." *Etherchain*. N.p., n.d. Web. 07 May

[9] Eyal, Ittay, and Emin Gn Sirer. "Majority is not enough: Bitcoin mining is vulnerable." *International Conference on Financial Cryptography and Data Security.* Springer Berlin Heidelberg, 2014.

[10] Hruska, Joel. "One Bitcoin Group Now Controls 51% of Total Mining Power, Threatening Entire Currency's Safety." *ExtremeTech*. N.p., 16 June 2014. Web. 07 May 2017.

[11] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.

[12] "Proof of Work." *Bitcoin Wiki*. N.p., n.d. Web. 07 May 2017.

[13] "Reward." *Bitcoin Wiki*. N.p., n.d. Web. 07 May 2017.

[14] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." *Ethereum Project Yellow Paper* 151 (2014). 2017.

VI. Appendix

A. Transacation Fee Markov Calculation

The rewards calculation is slightly more involved than that for the block reward model. Many are equivalent as they were in the case of block rewards calculation, with the exception of those corresponding to state 0 and 0" for the selfish miner. The 0" reward is simply 1 block reward respectively for the honest miner and selfish miner respectively for who honestly mines the block in that state. For the case of 0, on the other hand, there are different rewards per the span since the previous block was transmitted, which we denote as t. This analysis follows as from [4]:

- $t \leq \lambda$: In this case, since the state essentially represents a continuum of states, all the rewards paths must be analyzed in this case. Note that, in this case, the Markov Model previously shown is slightly modified such that the transition probabilities of $e^{-\lambda}$ are now $e^{-(\lambda-t)}$, since (for a given t) the probability we find a block before $\lambda - t$ is $e^{-(\lambda-t)}$:
  - $0 \to 0$: If the attacker finds the block after $\lambda - t$ time, he simply releases the block, getting a reward of 1, which occurs with probability $\beta e^{-(\lambda-t)}$
  - $0 \to 1 \to 0' \to 0$: If the attacker finds a block followed by an honest miner finding a block, the network is left in the same block race situation, in which case the analysis is identical to the Block Rewards model. This state is reached with probability $(1-\alpha)\beta(1-e^{-(\lambda-t)})$, at which point there are two paths that result in the selfish miner being rewarded.
  - $0 \to 1 \to 2 \to 0'' \to 0$: Attacker is guaranteed in this case to get 2 block rewards, since he will always have an advantage over the publicly mined blocks, occurs with probability of $\alpha\beta(1-e^{-(\lambda-t)})$
- $t \geq \lambda$: Attacker simply releases the block, getting a reward of 1, occurring with probability β

Thus, in other words, we have:

$$\mathcal{R}_{SM}^{(0)}(\alpha,\beta,t) = \begin{cases} p_0 e^{-t}(\beta e^{-(\lambda-t)} + \alpha\beta(1-e^{-(\lambda-t)}) + \\ (1-\alpha)\beta(1-e^{-(\lambda-t)})(2\alpha+\gamma(1-\alpha)) & t \leq \lambda) \\ p_0 e^{-t}\left(\beta e^{-\lambda}\right) & t > \lambda \end{cases}$$

Prior to combining the two, the total rewards must be simplified across all possible transmission times. Thus, we now integrate these across the entire time domain to find the expected reward from state 0 and add the others subsequently, since these follow equivalently from before:

$$\begin{aligned} \mathcal{R}_{SM}^{(0)}(\alpha,\beta) &= \int_{t=0}^{\infty} \mathcal{R}_{SM}^{(0)}(\alpha,\beta,t) \\ &= \int_{t=0}^{\lambda} \mathcal{R}_{SM}^{(0)}(\alpha,\beta,t) + \int_{t=\lambda}^{\infty} \mathcal{R}_{SM}^{(0)}(\alpha,\beta,t) \\ &= \int_{t=0}^{\lambda} p_0 e^{-t}(\beta e^{-(\lambda-t)} + \alpha\beta(1-e^{-(\lambda-t)}) + \\ (1-\alpha)\beta(1-e^{-(\lambda-t)})(2\alpha+\gamma(1-\alpha))) &+ \int_{t=\lambda}^{\infty} p_0 e^{-t}\left(\beta e^{-\lambda}\right) \\ &= p_0 \int_{t=0}^{\lambda} (\beta e^{-\lambda} + \alpha\beta(e^{-t}-e^{-\lambda}) + \\ (1-\alpha)\beta(e^{-t}-e^{-\lambda})(2\alpha+\gamma(1-\alpha))) &+ p_0\beta e^{-\lambda}\int_{t=\lambda}^{\infty} e^{-t} \\ &= p_0(\lambda\beta e^{-\lambda} + \alpha\beta(1-e^{-\lambda}-\lambda e^{-\lambda}) + \\ (1-\alpha)\beta(1-e^{-\lambda}-e^{-\lambda})(2\alpha+\gamma(1-\alpha))) &+ p_0\beta e^{-\lambda}e^{-\lambda} \\ &= p_0(\lambda\beta e^{-\lambda} + \alpha\beta(1-e^{-\lambda}(1-\lambda)) + \\ (1-\alpha)\beta(1-e^{-\lambda}(1-\lambda))(2\alpha+\gamma(1-\alpha)) &+ \beta e^{-2\lambda}) \end{aligned}$$

The remainder of the selfish miner rewards are largely equivalent to the previous case, meaning we have:

$$\begin{aligned} \mathcal{R}_{SM}^{(0')}(\alpha,\beta) &= p_{0'}\left(\mathbb{P}_2[0' \to 0] + 2\mathbb{P}_3[0' \to 0]\right) \\ &= p_{0'}\left(\gamma(1-\alpha) + 2\alpha\right]) \end{aligned}$$

$$\mathcal{R}_{SM}^{(0'')}(\alpha,\beta) = p_{0''}\left(\mathbb{P}[0'' \to 0]]\right) = \alpha p_{0''}$$

$$\mathcal{R}_{SM}^{(2)}(\alpha,\beta) = p_2\left(\mathbb{P}[2 \to 0'']\right) = (1-\alpha)p_2$$

$$\forall k > 2 : \sum_{k=3}^{\infty} \mathcal{R}_{SM}^{(k)}(\alpha,\beta) = \sum_{k=3}^{\infty} p_k\left(\mathbb{P}[k \to k-1]\right) = (1-\alpha)\sum_{k=3}^{\infty} p_k$$

The honest miner, however, is identical to the block reward model with the exception of adding in the $0''$ state, which is treated nearly identically to the 0' state:

$$\mathcal{R}_{H}^{(0)}(\alpha,\beta) = p_0\left(\mathbb{P}[0 \to 0]\right) = (1-\alpha)p_0$$

$$\begin{aligned} \mathcal{R}_{H}^{(0')}(\alpha,\beta) &= p_{0'}\left(2\mathbb{P}_1[0' \to 0] + \mathbb{P}_2[0' \to 0]\right) \\ &= p_{0'}\left(2(1-\gamma)(1-\alpha) + \gamma(1-\alpha)\right]) \end{aligned}$$

$$\mathcal{R}_{H}^{(0'')}(\alpha,\beta) = p_{0''}\left(\mathbb{P}[0'' \to 0]\right) = (1-\alpha)p_{0''}$$