An Exploration of $33.\bar{3}\%$ Attack Vectors in Pure Proof of Stake

Aparna Krishnan, Zubin Koticha, Alexis Gauba,
Maaz Uddin, Dev Ojha, Philip Hayes
University of California, Berkeley
Blockchain Lab, Sutardja Center for Engineering and Technology
Blockchain at Berkeley, Research and Development - Cryptoeconomics Research

Special Thanks to Karl Floersch from the Ethereum Foundation,
Profs. Dawn Song, Gireeja Ranade, Satish Rao Department of EECS, UC Berkeley,
Sankalp Aggarwal from Tendermint

January 2018

Abstract

Proof of Stake (PoS) is a blockchain consensus protocol in which validators staking some amount of funds can both propose and vote on new blocks in the network. These validators command voting power proportional to the percentage of funds they have bonded. A key vulnerability of the PoS systems that we model is that, if a single validator reaches $\frac{1}{3}$ stake, the attack threshold, because someone with $\frac{1}{3}$ stake has the ability to prevent network consensus. In this paper, we conduct an examination of the possibility of a validator reaching the attack threshold in the network via two methods.

We first examine the Superior Returns Attack Threshold (SRAT) method, in which a member of the network increases her ownership of tokens through investing in high-risk, high-return securities until she has sufficient wealth that when she stakes her wealth, she will command an attack stake. We next inspect the Random Attack Threshold (RAT) method, whereby a validator unintentionally accumulates an attack stake through stochastic randomness.

In the first case, we show that there is a non-zero probability of a validator obtaining $\frac{1}{3}$ stake when intentionally trying to do so, which may pose potential challenges to the security of pure PoS, but in the second case, we show that there is a near zero probability of a validator obtaining $\frac{1}{3}$ stake through stochastic randomness with constant rewards.

1 Introduction

Proof of Work and Proof of Stake:

Cryptocurrencies like Bitcoin have been gaining note recently for distinct benefits they offer over fiat currency including increased transparency among other parameters [BTC]. These currencies are backed by distributed ledgers known as blockchains. The computers that secure these blockchains are known as miners. These miners have a monetary incentive to come to consensus with other miners regarding the ledger's history. This process referenced as Nakamoto Consensus in the space, which is backed by a process known as Proof of Work (PoW) [Bit].

Proof of Work is wasteful of energy and computational power, slowing down the network considerably. Proof of Stake (PoS) has been proposed as an alternative blockchain consensus protocol. Proponents argue that PoS resolves challenges with Proof of Work such as energy inefficiency, high latency, and centralization risk, to name a few [Eth17]. PoS has been in development for major networks such as Ethereum, NEO, and Tendermint among others due to the strong arguments in its favor, [Eth17].

In a PoS system, consensus is reached through the agreement of validators who have bonded their coins to the network. These validators are akin to miners in PoW. However, in PoS, validators create a deposit (a bond) of coins which allows them to propose and vote on new blocks. In some iterations of PoS (BFT-style), one of these validators is randomly chosen as a proposer to propose a new block to add to the blockchain in every round[Eth17]. The rest of the validators vote on the inclusion of this new block, with voting power in proportion to the bonds they have staked. Consensus is achieved when the 66% of validators agree on accepting the new block .

The PoS system we examine is one in which a proposer is chosen from among the validators with a likelihood proportional to the funds they have bonded in the network [Eth17]. PoS depends on both rewards and penalties to incentivize agents in the system to act honestly. The analyses in this paper are inspired by BFT-style PoS.

Motivation (the Problem of $\frac{1}{3}$):

In traditional PoW blockchains, a malicious actor with 51% of mining power can rewrite the history of the blockchain, and thus 51% is considered the security threshold of a PoW blockchain [Nak08].

Brewer's CAP Theorem posits the trade-offs between consistency and availability in a distributed system [GL02]. To find a balance between consistency and availability, the PoS system we model can tolerate up to $\frac{1}{3}$ byzantine faults [GL02]. Given that voting power in PoS is proportional to the number of staked tokens one has bonded, at $\frac{1}{3}$ stake an attacker can refuse to finalize any further blocks, or censor the network to their specific views by refusing to finalize specific blocks of their choosing [Eth17]. Malicious validators can also indefinitely stall consensus, by not approving any new blocks at all. As such, it's important to decrease the likelihood of any validator commanding $\frac{1}{3}$ of the total staked tokens. While an attack can be mitigated by a hard fork, in which the network split can undo malicious behavior, such a solution is economically and socially costly, and undermines trust in the blockchain [But16]. Therefore, in PoS, preventing any single validator gaining enough stake to undermine consensus is a paramount issue for the network. As such, it's important to decrease the likelihood of any staker to commanding $\frac{1}{3}$ of the total staked tokens.

In this paper, we analyze the difficulty of a validator obtaining $\frac{1}{3}$ stake. We look at two possibilities of reaching $\frac{1}{3}$ stake. The first case is the Superior Returns Attack Threshold, in which a member of the network becomes sufficiently wealthy through high returns that when she decides to stake her wealth, she has enough for an attack stake. The second is the Random Attack Threshold; a case wherein a validator gets to $\frac{1}{3}$ through stochastic randomness.

Reward Scheme for our PoS:

We model a PoS system in which a proposer is chosen from the set of validators to compose a block upon which other validators vote. If a validator has bonded $X\%$ of all staked tokens, then that validator will be chosen as the next block proposer with $X\%$ likelihood. When a user of the

blockchain proposes a transaction, the user includes a transaction fee, all of which is credited to the block proposer. Then validators vote on each proposed block. Each validator, including the proposer, receives a small validation reward for voting on each block, and the validation reward is proportional to the validator's stake.

There are three mechanisms by which the protocol can incentivize honest behavior in PoS:

- Transaction Fees: These transaction fees disincentivize block proposers from including empty blocks, prevent users from spamming the network with trivial transactions, and provide a mechanism for block proposers to order transactions.
- Validation Rewards: Validation rewards are given to all validators who sign off on a block. This both incentivizes ETH holders to become validators, and incentivizes existing validators to be available to sign each block. Note that these validation rewards must be paid proportional to validators' stake.
- Punishments: There are a subset of punishments, some of which (e.g. slashing) are only levied on actors that are acting in a demonstrably malicious manner, and others which may be levied on honest actors when the source of malice is uncertain (e.g. data unavailability punishments).

Superior Returns Attack Threshold (SRAT):

Validators expect to get some return for their validation, r_v, from staking their tokens on the network via their share of validation rewards and transaction fees. If some ETH holder A decides to not to stake her tokens, and instead invests these in another asset hr which has higher risk than validation, but also has a higher expected rate of return $r_{hr} > r_v$, she can expect on average to grow her wealth at a faster rate than bonded validators. If she, in fact, does receive a higher return, she can potentially accumulate sufficient ETH that when she decides to stake her tokens, she has a stake over $\frac{1}{3}$.

Random Attack Threshold (RAT):

In this system, the stake of any validator with $x\%$ of bonded funds $< 33\%$, will be expected to remain at $x\%$ staked at any later time t given the following:

- All validators restake all their rewards
- No validators unbond their stake
- No other users stake, so no one is added to validator set

Note that proposers collect both transaction fees and validation rewards, while other validators collect only validation rewards. Since a proposer is selected at random from the group of validators, it is possible that one validator B is selected as the proposer much more often than others. Assuming B immediately restakes her income, her wealth will grow with a higher rate of return than other validators, and it is possible over time that this increases her stake to $\frac{1}{3}$.

We model the case of a validator probabilistically obtaining an attack stake using a Pólya's Urn model.

Before we analyze the SRAT and RAT methods, let us examine the rewards scheme in more depth, specifically the awards for proposers and other validators in this scheme.

1.1 Rewards Scheme - Awards for Proposers vs. other Validators:

We argue that block proposers must get rewards for each block at least as large as those of a validator possessing the same stake. This helps to prevent large stakers from forming multiple bonded Sybil identities.

We demonstrate that stakers have a strong incentive to break their stake into Sybil identities in the case that validation rewards are not paid to block proposers through the following example:

Consider a user who bonds her tokens, which amount to x% of all staked tokens, through one single account, she then has an x% likelihood of being chosen as block proposer.

1. In Case 1, if she is not chosen as a proposer, she receives x% of the validation rewards. If she is chosen to be a proposer, she receives transaction fees from that block but misses out on validation rewards.

2. In Case 2, if she is not chosen to be the block proposer, she receives x% of validation rewards, similar to Case 1. However, unlike in Case 1, if she is chosen to be a proposer, she receives transaction fees from that block *as well as* x% of the validation rewards

She would respond to Case 1, a design lacking validation rewards to block proposers, by forming Sybil identities: since she would not be getting x% of validation rewards when chosen as a block proposer, she would have been better off splitting her x% into k accounts each with $\frac{x}{k}$% of total staked tokens. This maintains her likelihood of being chosen as block proposer at x%. However, when she is chosen as a proposer, her other accounts will still deliver $x - \frac{x}{k}$% of validations rewards to her for that block. As k becomes larger, in the case the probability of her being chosen as block proposer remains constant while she increases her share of validation rewards.

We want to prevent Sybil identities because increasing the number of distinct accounts staked will increase network latency [But17]. An increase in network latency would result in discrepancies in information disseminated to the validator pool, in that not all validators would hear about a specific block, leading to both inefficiency and inconsistency within the network [Jae16].

2 Superior Returns Attack Threshold (SRAT)

2.1 Sketch of a SRAT:

A risk-tolerant attacker with some wealth of Ether $w < 33\%$ of all staked tokens invests in some Ethereum-based financial asset that offers a greater return than staking, until the attacker exceeds $\frac{1}{3}$ of staked tokens and can effectively control consensus.

2.2 How to obtain SRAT:

A user can increase her wealth by investing and obtaining superior returns; these returns may be accumulated either off-chain or on-chain. As of now, superior returns are commonly earned off-chain (in fiat currency). There are two ways in which the network can stay protected from off-chain wealth. The first situation is if the staking currency becomes increasingly expensive in terms of fiat currency. A second method we propose is to make the supply of tokens inelastic by making sure that a minimum percentage of staked tokens remains locked at any given time, with a penalty fee being charged for unbonding before the end of the staking period.

While currently superior returns are usually available only off-chain, this is likely to change. Blockchain platforms designed for smart-contracting, such as Ethereum, will eventually allow for a wide variety of securities to be trustlessly traded on-chain. Some of these investment classes will provide higher expected returns (and higher risk) than the low-risk validation rewards and transaction fees provided by staking. Thus, if an investor is able to purchase assets with expected returns (and risk) greater than those of being a staker, then the investor may be able to grow her money faster than bonded stakers, if her realized returns are indeed higher than those for stakers. If so, eventually she can obtain sufficient wealth to be able to stake it and get control over $\frac{1}{3}$ of the network's total stake. (As the security is denominated in Ether, it can readily be sold and the Ether can be restaked, as opposed to a dollar-denominated security that would have to be exchanged for dollars and then for Ether.)

According to the Capital Asset Pricing Model, in an efficient market, assets with greater risk associated with them must provide a higher rate of expected rewards [Sha64]. Since many new blockchains, such as Ethereum, provide the flexibility to allow for a wide variety of different investment types on chain, in the long run, there should exist some security in the Ethereum ecosystem which provides an expected return greater than rewards gained by validation, at a greater level of risk. That is, if bonding as an honest validator provides expected reward r_v and expected punishment p_v (a function of unavailability punishments) over some distinct time period, there exists some high-risk security which provides some superior expected return $r_{hr} = r_v - p_v + \varepsilon$ (where $\varepsilon > 0$) over the same time period.

Let's say an investor begins with a certain amount of wealth that, were she to stake at the beginning, would give her a stake of x_0 (as a percentage of the total bonded stake). Let's say that, instead of staking this initial wealth, she invests in the high-risk asset. Then x_N, the percentage amount of stake she could possess after N time periods is given by:

$$x_N = x_0 * (1 + r_{hr})^N \tag{1}$$

or

$$x_N = x_0 * (1 + r_v - p_v + \varepsilon)^N \tag{2}$$

This is assuming that all stakers rebond their validation earnings and that no other Ether gets bonded or unbonded. The expected amount of time it would take an attacker with x% of stake to obtain 1/3 stake is therefore N periods as demonstrated by the following equation:

$$\frac{1}{3} * 100 * (1 + r - p)^N = x * (1 + r - p + \varepsilon)^N \tag{3}$$

Solving for N, the expected amount of time it would take a user to get to $\frac{1}{3}$ stake:

$$N = \frac{\ln(100) - \ln 3x}{\ln(1 + r - p + \varepsilon) - \ln(1 + r - p)} \tag{4}$$

In this case, as long as the return on these high risk investments is less than or equal to the return provided by staking, being a staker is preferable since it provides an equivalent or greater risk-free rate of return. However, if the return from the risky asset is greater, a risk-tolerant attacker would potentially be incentivized to purchase that asset and thus accumulate tokens, which they could then stake.

2.3 Difficulty of such a SRAT:

It is unclear that such returns as below are reasonable on Ethereum, but are provided as a rough exercise to see what would happen if Ethereum rewards matched those that are expected for USD assets.

Since the annualized return of the SP500 has averaged around 10% for the last 90 years [Sni], the future amount of Ether invested at a comparable rate for N years is:

$$FV_{ETH} = PV_{ETH} * (1.1)^N \tag{5}$$

(Again, it is not clear that you'd be able to attain a 10% annualized Ether return, even if you would be able to get a 10% annualized dollar return.)

Assuming the following:

- Staked Ether grows at a rate $r_v < 10\%$ due to validation rewards and transaction fees
- A party commands $x\%$ of the initial Ether
- No new individuals enter or exit the validator set
- All validators restake their new earnings

Then we see that the time until that party commands $\frac{1}{3}$ of the total staked amount is:

$$\frac{1}{3} * 100 * (1 + r)^N = x(1.1)^N \tag{6}$$

Solving for N, we see:

$$N = \frac{\ln(100) - \ln(3x)}{\ln 1.1 - \ln(1 + r)} \tag{7}$$

The risk-free rate (1-yr treasury note) is 1.4% currently. Assume that validation rewards are exactly 1.4%. An attacker starting with 10% of the total staked tokens will thus command $\frac{1}{3}$ of the network in 6.27 years.

$$N = \frac{\ln(100) - \ln(60)}{\ln 1.1 - \ln 1.014} \tag{8}$$

Assuming she invests in leveraged assets, such as 2x leveraged SP500 (by margin trading), she can approach 20% returns. An attacker would own $\frac{1}{3}$ of staked tokens in 3.03 years in this case.

$$N = \frac{\ln(100) - \ln(60)}{\ln 1.2 - \ln 1.014} \tag{9}$$

From the above we can deduce the following:

- To protect the network, we need to Maximize N:

$$\frac{1}{3} * (100 - x)(1 + r - p)^N = x * (1 + r - p + \varepsilon)^N \tag{10}$$

- Accordingly, the following must hold true: $r - p > 0$

Takeaway:

Two arguments often used in favor of PoS are:

- As a wealthy actor tries to "corner the market" on ETH through purchasing more, she will find herself paying more for each additional ETH she buys.
- As any malicious validator approaches $\frac{1}{3}$ of the total share of staked tokens, the members of the network will respond by bonding their ETH and joining the validator set, so that they can dilute the attacker's share and prevent a $\frac{1}{3}$ attack.

However, this type of attack may call into question the validity of those two arguments. As the Ethereum ecosystem develops, eventually individuals will be able to invest in on-chain assets and earn ETH-denominated returns to their existing ETH. Therefore, a wealthy actor never needs to 'buy' her way into the network; she simply needs to invest at some high-reward ETH-denominated interest-rate. While is it not guaranteed that she will, indeed, recieve the high returns needed to obtain the wealth required for an attack stake (given the higher risk of her assets), it is quite likely that she, or some other attacker, will be ultimately able to achieve such wealth.

Further, a big problem with this kind of attack is that as any actor accumulates additional ETH, she is reducing the rest of the network's ability to bond more tokens in order to dilute their stake. That is, since there is a fixed supply of tokens which can be earned from external non-validation activities, as an actor gains their 20% returns, that actor is simultaneously reducing the token share of others. Therefore, the rest of the network has less ETH to add to the validator set in order to prevent a $\frac{1}{3}$ attack.

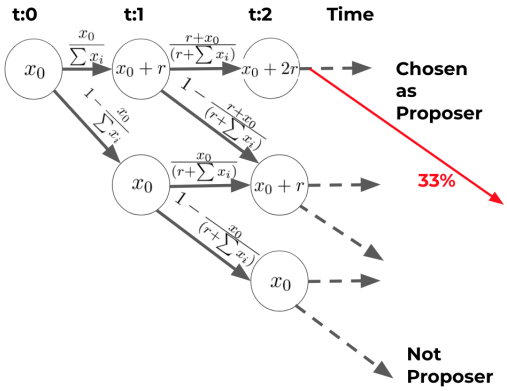
3 Random Attack Threshold (RAT) Probability of Reaching $\frac{1}{3}$ through Randomness:

One way to reach $\frac{1}{3}$ stake is through stochastic randomness in that there is a small likelihood that validators could be repeatedly chosen to select a block, collect transaction fees, immediately restake rewards, and continually increase their fractions of the network's total deposit until they reach $\frac{1}{3}$. Thus we analyze the probability of any validator reaching this $\frac{1}{3}$ threshold.

Here, we consider a simplified version of a typical Proof-of-Stake protocol in which we have m validators. Since we are mostly concerned with local dynamics in this particular discussion, we will assume that the number of validators remains fixed over time and there is no external capital in or out flow, i.e., no validators bond additional stake or cash out their stake. Our abstract process proceeds as follows: at every round, a single validator is chosen with probability equal to their proportion of staked tokens. If a validator is chosen, then they receive a reward of some constant number of staking tokens. For the purposes of simplification we consider the reward to be 1 token. The general case for a reward of r tokens can easily be derived from this simplification. We examine how long it takes until any validator's proportion of the tokens passes the byzantine threshold.

3.1 Constant Reward Polya Urn

The Proof-of-Stake process lends itself quite naturally to an urn model in which we view the validators as different-colored balls in an urn. The number of i^{th} colored balls in the urn is then equal to the number of staking tokens owned by the i^{th} validator.



State Transition Diagram

The above state transition diagram is a visualization of the different stakes of the 0^{th} validator which we simulate with a Pólya Urn Model. We model a state transition diagram where each state represents the number of tokens staked. At time $t = 0$ the validator has x_0 number of tokens staked. The probability the 0^{th} validator gets chosen to be a proposer is $\frac{x_0}{\sum x_i}$ at $t = 0$ where x_i is the number of tokens the i^{th} validator has deposited. The probability that the 0^{th} validator is not chosen to be a proposer is $1 - \frac{x_0}{\sum x_i}$ at $t = 0$.

Definitions

A Pólya urn is an urn containing balls of up to K different colors, where $K >= 2$. The urn evolves in discrete time steps - at each step, one ball is sampled uniformly at random; The color of the withdrawn ball is observed, and the ball is returned to the urn. In addition, a ball of the same color as the ball that was drawn is added to the urn.

In section 4.1, we consider a Pólya urn with m validators. Let the composition of the validator set at time t be represented by the vector $X_t = (x_t^{(1)}, ..., , x_t^{(m)})$ where $x_t^{(i)}$ is the stake (number of tokens) of the i^{th} validator at time t. We consider a distribution after we simulate the process for n rounds. y_j is the number of times the j^{th} validator wins in the n rounds. Let Γ_0 be the total stake at the start in the urn.

$$\Gamma_0 = \sum_{i=1}^{m} x_0^{(i)}$$

Notations:

- $\langle x \rangle_n = x(x+1)..(x+(n-1))$
- $Y[j]$ is the j^{th} element of the Y vector.
- $\binom{n}{y_1,..,y_m} = \frac{n!}{y_1! * .. * y_m!}$

3.2 Convergence of the Distribution

Theorem 3.2.1. *Let $p_n^{(i)} = \frac{x_n^{(i)}}{\sum_{j=1}^{m'} x_n^{(j)}}$, be the proportion of stake of the i^{th} validator after n rounds. Let $\mathcal{F}_n = \sigma(v_i)$, where v_i is the i^{th} validator. Then $p_n^{(i)}$ is a Martingale w.r.t $\mathcal{F}_{n-1}$*

Proof.

$$E[p_n^{(i)}|\mathcal{F}_{n-1}] = E[\frac{x_n^{(i)}}{\sum_{j=1}^{m'} x_n^{(j)}}|\mathcal{F}_{n-1}] = E[\frac{x_n^{(i)}}{\sum_{j=1}^{m'} x_0^{(j)} + n}|\mathcal{F}_{n-1}]$$

$$= \frac{1}{\sum_{j=1}^{m'} x_0^{(j)} + n} E[x_n^{(i)}|\mathcal{F}_{n-1}] = \frac{1}{\sum_{j=1}^{m'} x_0^{(j)} + n}(x_{n-1}^{(i)} + \frac{x_{n-1}^{(i)}}{\sum_{j=1}^{m'} x_0^{(j)} + n - 1}) = p_{n-1}^{(i)}$$

□

The takeaway from the above theorem is that since $p_n^{(i)}$ is a martingale and is bounded by 1 on the upper end, it must surely converge to some value.

3.3 Distribution and Expected Value for a Constant Reward

Theorem 3.3.1. *Suppose a Pólya urn starts with an initial composition of X_0 and initial total stake Γ_0. Let X_n be the composition of the Pólya urn after n draws, then*

$$P(x_n^{(j)} = x_0^{(j)} + y_j, j = 1, .., m) = \frac{\prod_{j=1}^{m} \langle x_0^{(j)} \rangle_{y_j}}{\langle \Gamma_0 \rangle_n} \binom{n}{y_1, .., y_m} \tag{11}$$

Proof. If the i^{th} colored ball is drawn y_i times, then $x_0^{(i)}(x_0^{(i)}+1)..(x_0^{(i)}+y_i-1)$ is in the numerator in some ordering. Thus independent of ordering the numerator will remain $\prod_{j=1}^{m} \langle x_0^{(j)} \rangle_{y_j}$ and $\langle \Gamma_0 \rangle_n$ will be in the denominator. There are $\binom{n}{y_1,..,y_m}$ ways of ordering the distribution in the numerator. □

Using Theorem 4.3.1, we can calculate results like the expected value vector and the probability of any validator reaching a certain threshold after t steps.

3.3.1 Calculating the probability that someone in the network reaches the threshold

Let v_i be the i^{th} validator and V be the set of all validators. $S(v_i)$ be the set of events such that the i^{th} validator crosses the threshold. Let $P(v_i)$ be the probability that the i^{th} validator crosses the threshold. Let z_i be the minimum number of times the i^{th} validator would have to be chosen to reach the threshold p (p is a proportion like $\frac{1}{3}$) in n rounds.

$$\frac{x_0^{(i)} + z_i}{\Gamma_0 + n} \geq p$$

$$z_i = \lceil p * (\Gamma_0 + n) - x_0^{(i)} \rceil \tag{12}$$

We can model the probability of each validator v_i reaching a certain threshold by considering a 2 validator system - the first being validator v_i and the second being all the other validators.

Corollary 3.3.1.1. *The probability of a single validator v_k crossing the given threshold after n rounds is:*

$$\frac{\sum_{i=z_k}^{n} \binom{n}{i} \langle x_0^{(k)} \rangle_i \langle \Gamma_0 - x_0^{(k)} \rangle_{n-i}}{\langle \Gamma_0 \rangle_n} \tag{13}$$

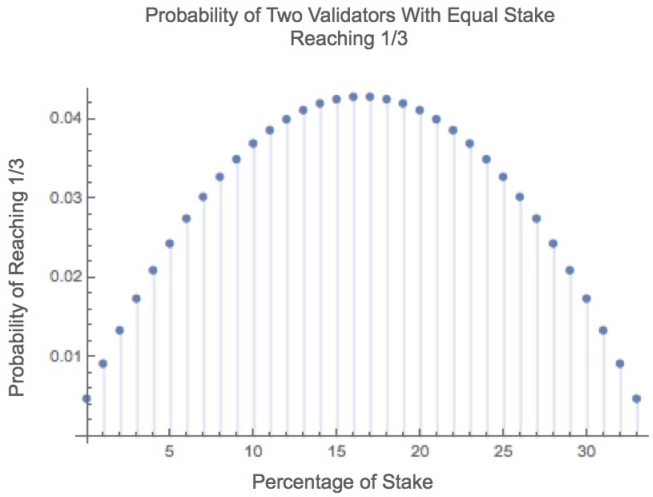
Corollary 3.3.1.2. *The probability of an index set S of the m' validators such that $m' <= m$, all reaching the threshold p after n rounds is represented by $P(\frac{x_n^{(S_k)}}{\Gamma_n} >= p, k = 1..m')$*

$$= \frac{\sum_{i_1=z_{S_1}}^{n-\sum_{k=1}^{m'} z_{s_k}} \sum_{i_2=z_{S_2}}^{n-i_1} \cdots \binom{n}{i_1, i_2, .., i_{m'}} \langle x_0^{(S_1)} \rangle_{i_1} \langle x_0^{(S_2)} \rangle_{i_2} .. \langle x_0^{(S_{m'})} \rangle_{i_{m'}}}{\langle \Gamma_0 \rangle_n} \tag{14}$$

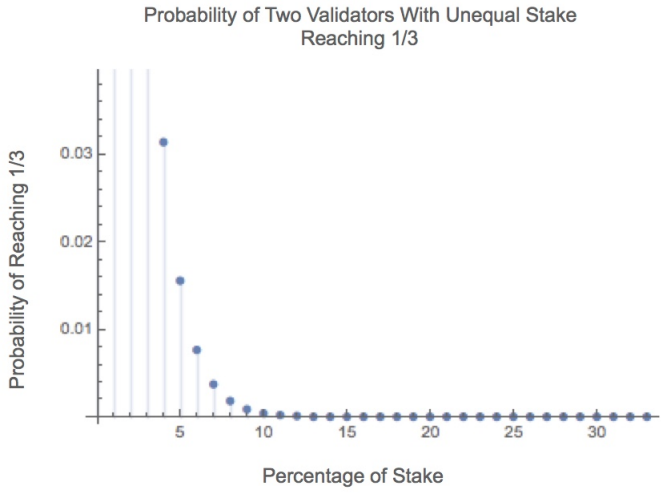
Corollary 3.3.1.3. *The probability that any validator crosses the given threshold p is represented by* $\frac{|\bigcup_{i=1}^{m} S(v_i)|}{|S(U)|}$

$$= \sum_{i=1}^{m} P(\frac{x_n^{(i)}}{\Gamma_n} >= p) - \sum_{i=1}^{m}\sum_{j=1}^{m-i} P(\frac{x_n^{(i)}}{\Gamma_n} >= p \ \ \& \ \ \frac{x_n^{(j)}}{\Gamma_n} >= p) + \sum_{i=1}^{m}\sum_{j=1}^{m-i}\sum_{k=1}^{m-i-j} P(\ldots) \qquad (15)$$

Each successive term should be significantly smaller than the previous term. For $Threshold = \frac{1}{2}$, there is only one term, since when there are more than two validators, it is impossible for two validators to each have $\frac{1}{2}$ of the total stake. Similarly, for $Threshold = \frac{1}{3}$ you only need the second order approximation.

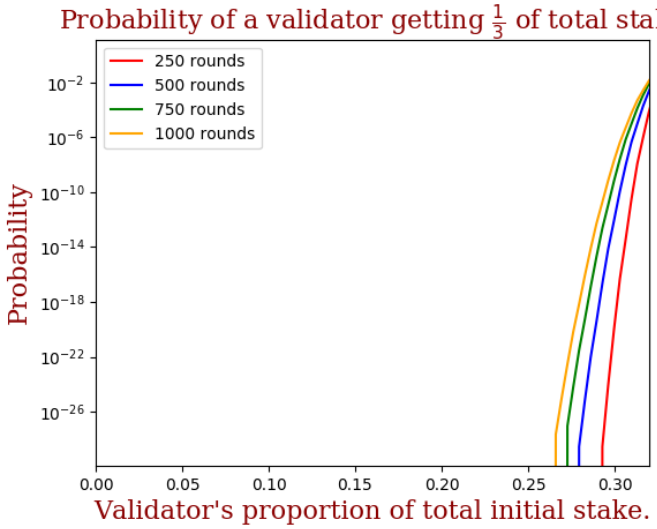


Beta Binomial Distribution (probability density graph) conveying the probability of validator one reaching $\frac{1}{3}$ stake, when the two validators begin with equal stake.

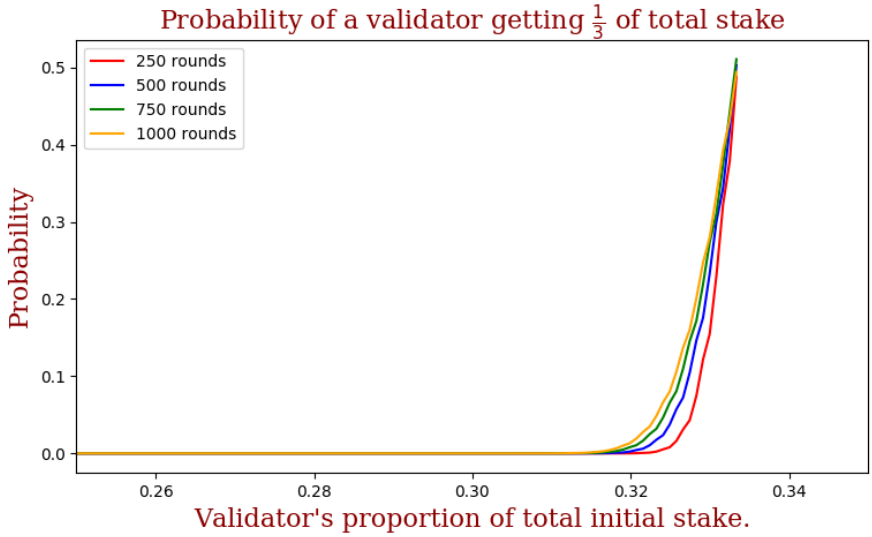


Beta Binomial Distribution (probability density graph) conveying the probability of validator one reaching $\frac{1}{3}$ stake, when the two validators begin with skewed stake.

In the following two scenarios, we model a system with 64 validators where the total amount of stake initially in the system is 2000 tokens and the reward to validators is one token.



Probability that one validator starting at different initial proportions of stake reaches $\frac{1}{3}$ stake, in a system where each of the 64 validators begin with equal stake. In the 1000 rounds the probability of reaching $\frac{1}{3}$ is 2.19912034213425E-101 %. Modelled here.



Probability that one validator starting at different initial proportions of stake reaches $\frac{1}{3}$ stake,in a system where each of the 64 validators begin with skewed stake.In the 1000 rounds the probability of reaching $\frac{1}{3}$ is 1.50938677083072E-130 %. Modelled here.

From the above graphs we see that the probability of any validator reaching the byzantine threshold is near zero when each validator has the same amount of stake. However, with a skewed distribution as seen in the second graph, validators with greater amounts of stake have a significantly higher likelihood of reaching $\frac{1}{3}$ stake.

Regardless of what the initial distribution is, section 4.1 proves that if the reward is a constant amount significantly less than the initial stake of each validator, the likelihood of any validator reaching the $\frac{1}{3}$ threshold is very low.

3.3.2 Calculating Expected Value

Let the expected value of stake of a validator after n rounds be $E[x_i^{(n)}]$.

$$E[x_0^{(i)}] = x_0^{(i)}$$

Corollary 3.3.1.4.

$$E[x_n^{(i)}] = x_0^{(i)} \frac{\Gamma_0 + n}{\Gamma_0}$$

Proof. Let $P(x_n^{(i)} = s)$ represent the probability of validator i having stake s in round n.

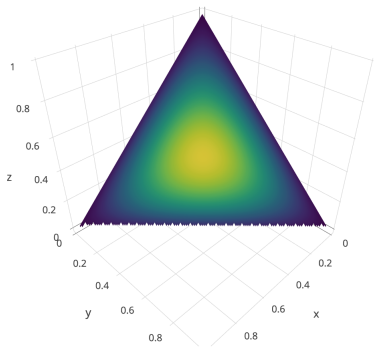
$$E[x_n^{(i)}] = \sum_{s \in values} \left(s + \frac{s}{\Gamma_0 + n - 1} \right) P(x_i = s)_{n-1} = E[x_{n-1}^{(i)}] \left(\frac{\Gamma_0 + n}{\Gamma_0 + n - 1} \right)$$

$\Rightarrow E[x_{n-1}^{(i)}] = E[x_{n-2}^{(i)}] \left(\frac{\Gamma_0 + n - 1}{\Gamma_0 + n - 2} \right)$ and so on. Repeatedly substituting this in the above equation we can represent $E[x_n^{(0)}]$ as $E[x_n^{(i)}] = x_0^{(i)} \frac{\Gamma_0 + n}{\Gamma_0}$. □

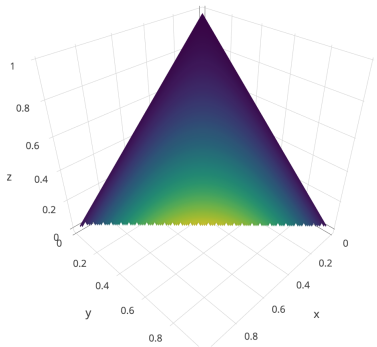
Theorem 3.3.2. *Consider a Pólya Urn which starts with an initial stake distribution vector X_0. As the number of draws $n \to \infty$*

$$\left(\frac{x_n^{(1)}}{n}, .., \frac{x_n^{(n)}}{n}\right) \to^D Dir(x_0^{(1)}, .., x_0^{(n)}). \tag{16}$$

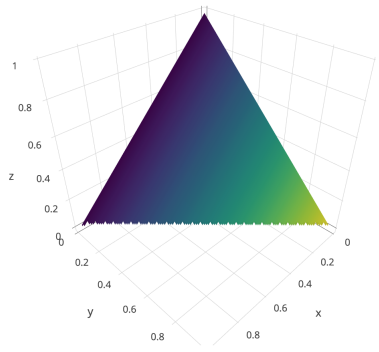
For the full proof, read [Bla73].



The above image shows a dirichlet graph of 3 validators all with initial equal stake.



The above image shows a dirichlet graph of 3 validators, 2 of which have equal stake and the other has slightly less stake than the others.



The above image shows a dirichlet graph of 3 validators, 2 of which have equal stake and the other has slightly more stake than the others.

Takeaway:

The Pólya Urn model reveals that there is a near zero probabilistic likelihood of an actor getting to $\frac{1}{3}$ stake based off of stochastic randomness when the reward is some constant value. When stake is split evenly amongst validators, it is least likely that any single one reaches $\frac{1}{3}$. This indicates that a setup where each validator starts of with equal stake would be ideal. In future work, we will explore what happens when modifying the reward matrix, inflationary rewards, random rewards and if a breaking up of large stake is less secure from a probabilistic perspective.

4 Conclusion:

In looking at the possibility of validators obtaining $\frac{1}{3}$ stake in the network, both intentionally and through stochastic randomness, we have exposed vulnerabilities in PoS regarding 33% attacks.
Key Findings:

1. It is possible to reach $\frac{1}{3}$ stake in the system when actively trying to do so as described by SRA
2. There is a near zero probabilistic likelihood of an actor getting to $\frac{1}{3}$ stake based off of stochastic randomness dependent upon the number of validators

5 Further Research:

Going forward, we will extend the Pólya Urn model, looking into Pólya's Urn with inflation. We plan to analyze the following 33% attacks: Deposit Panic, Bribing Attacks, and P+ Epsilon Attacks. Other areas of research include looking into block rewards, punishments, tradeoffs between security and latency, delegated proof of stake, and a two token model. We aim to optimize constraints and parameters as a linear program to figure out how to optimize a PoS system.

6 Appendix:

6.1 Intuition behind $\frac{1}{3}$ Byzantine Fault Tolerance

To further expand on why $\frac{1}{3}$ byzantine faults provides a balance between consistency and availability, consider the case of $\frac{3}{4}$ fault tolerance. In this scenario, a distributed system could handle up to $\frac{1}{4}$ availability faults, meaning that if any more than $\frac{1}{4}$ of validators were offline, the system would not be able to function, but allows for $\frac{1}{2}$ consistency faults. Thus in the case of the $\frac{1}{3}$ threshold, the system can handle up to $\frac{1}{3}$ consistency faults, and also $\frac{1}{3}$ availability faults, allowing for this balance between consistency and availability that major PoS systems including Ethereum and Tendermint employ.

References

[Sha64] William F. Sharpe. "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk". In: *The Journal of Finance* 19.3 (Sept. 1964), pp. 425–442.

[Bla73] David Blackwell. *Ferguson distributions via Polya urn schemes*. 1973. URL: https://projecteuclid.org/euclid.aos/1176342372.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem". In: *ACM Transactions on Programming Languages and Systems* 4 (1982).

[Pem90] Robin Pemantle. "A Time-Dependent Version of Polya's Urn". In: *Journal of Theoretical Probability* 3 (1990), pp. 627–637.

[GL02] Seth Gilbert and Nancy Lynch. *Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services*. 2002. URL: https://www.glassbeam.com/sites/all/themes/glassbeam/images/blog/10.1.1.67.6951.pdf.

[Nak08] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: https://bitcoin.org/bitcoin.pdf.

[Hel13] Nora Helfand. "Poly'a Urn and the Beta Bernoulli Process". 2013.

[But16] Vitalik Buterin. *A Proof of Stake Design Philosophy*. 2016. URL: https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51.

[Jae16] Ethan Buchman Jae Kwon. *Cosmos: A Network of Distributed Ledgers*. 2016. URL: https://cosmos.network/whitepaper.

[Sch16] Kurt Schmidheiny. *Monte Carlo Experiments*. 2016. URL: https://www.schmidheiny.name/teaching/montecarlo2up.pdf.

[But17] Vitalik Buterin. *Parametrizing Casper: the decentralization/finality time/overhead tradeoff*. Jan. 2017. URL: https://medium.com/@VitalikButerin/parametrizing-casper-the-decentralization-finality-time-overhead-tradeoff-3f2011672735.

[BG17] Vitalik Buterin and Virgil Griffith. *Casper the Friendly Finality Gadget*. 2017. URL: https://arxiv.org/pdf/1710.09437.pdf.

[Eth17] EthereumFoundation. *Proof of Stake FAQ*. 2017. URL: https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ.

[Bit] Bitcoin. *Hard Fork, Hard-Forking Change*. URL: https://bitcoin.org/en/glossary/hard-fork.

[BTC] BTCCore. *Bitcoin FAQ*. URL: https://bitcoin.org/en/faq#what-are-the-advantages-of-bitcoin.

[Pol] Pollyanna. *Polya Urn*. URL: http://dornsife.usc.edu/assets/sites/406/docs/505b/polya.urn.pdf.

[Sni] Scott Snider. *What is the average annual return for the S&P 500?* URL: https://www.investopedia.com/ask/answers/042415/what-average-annual-return-sp-500.asp.