Acki Nacki Tokenomics

Mitja Goroshevsky Nikita Sattarov

January 2025

1 Abstract

We present Acki Nacki network Tokenomics, optimized for maximum decentralization from the start of the network, security and fairness.

For Acki Nacki technical description please refer to: “ACKI NACKI: a Probabilistic Proof-of-Stake consensus protocol with fast finality and parallelization” [6].

2 Parameters

In Acki Nacki, there are five types of Network Participants: Block Producer, Block Keeper, Block Verifier (or Acki-Nacki), Block Manager, and Mobile Verifier. Throughout this paper, we sometimes refer to the first three participants — Block Producer, Block Keeper, and Block Verifier — as ’Block Keepers’ when we don’t need to address their roles specifically.

Since the construction and analysis of Tokenomics require a deep delving into mathematical theory, this article contains a large number of formulas, variables, and their dependencies. Therefore before each block of formulas, a brief description of the variables used in that block is provided. Additionally, a table with descriptions of all the variables used in this document is included in the Appendix "Glossary" A.

In many formulas, parameters will be used that are not functions of other variables but depend on the internal condition of the network, individual actions of the network participants, etc. In that case we will use the notation $|_t$ to indicate that these parameters are captured in the formula at time t.

Since many of the formulas are quite large and complex, throughout the document, additional redefinitions will often be introduced to simplify reading the formulas. For such purposes, a variable u with a subscript will be used, which is not described in the glossary and is not defined before each block of formulas, as it does not carry significant meaning.

3 Quick Facts

Token	Supply	Emission	Function
NACKL	10.4 B	Curve, final	Network security
SHELL	Unlimited	Pledge	Computation

4 Background

Satoshi Nakamoto, the creator of Bitcoin, emphasized the significance of the Cost of Production Theory of Value in the cryptocurrency’s economic model. In a 2010 forum post [14], Nakamoto articulated:

> "The price of any commodity tends to gravitate toward the production cost. If the price is below cost, then production slows down. If the price is above cost, profit can be made by generating and selling more. At the same time, the increased production would increase the difficulty, pushing the cost of generating towards the price."
>
> — Satoshi Nakamoto

This perspective aligns with the classical economic theories proposed by Adam Smith and David Ricardo. Adam Smith, in his seminal work An Inquiry into the Nature and Causes of the Wealth of Nations (1776), introduced the Labor Theory of Value, suggesting that the value of a commodity is fundamentally derived from the labor required for its production (Smith, 1776). Smith posited that:

> "The real price of everything, what everything really costs to the man who wants to acquire it, is the toil and trouble of acquiring it."
>
> — Adam Smith, 1776, p. 36 [20]

David Ricardo expanded on this in On the Principles of Political Economy and Taxation (1817), arguing that the value of goods is determined by the total quantity of labor necessary for their production, stating:

> "The value of a commodity... depends on the relative quantity of labor which is necessary for its production."
>
> — David Ricardo, 1817, p. 11 [17]

By integrating a self-adjusting difficulty mechanism into Bitcoin's protocol, Nakamoto ensured that the computational resources and energy expended on mining — analogous to labor in classical economics — play a pivotal role in determining Bitcoin's value. As mining difficulty increases with more miners joining the network, the cost of production rises. This mechanism reflects Ricardo's assertion that production costs influence market prices, aligning with the idea that commodities gravitate towards their natural price, defined by production expenses.

The interplay between miners' expenditures on hardware and electricity (the "toil and trouble") and the network's difficulty adjustment embodies Smith's and Ricardo's theories in a digital context.

An examination of Bitcoin's price graph in relation to its mining reward halvings suggests a correlation that aligns with Satoshi Nakamoto's perspective. Specifically, the periodic halving events — where the reward for mining new blocks is reduced by half — effectively double the cost of producing each new Bitcoin. These halvings have historically coincided with significant increases in Bitcoin's market price, implying that the rising production costs influence its valuation. This pattern supports the notion that, at least in the context of Bitcoin, production costs play a substantial role in price formation, echoing classical economic theories within a modern, digital framework.

4.1 Bitcoin's Halving Events and Price Correlation

Bitcoin's protocol includes a built-in mechanism that reduces the mining reward by half approximately every four years — a process known as "halving." This mechanism:

Increases Production Costs: After each halving, miners receive 50% fewer Bitcoins for the same amount of work, effectively doubling the cost to produce one Bitcoin if other factors remain constant.

Affects Supply Dynamics: The reduced influx of new Bitcoins slows down the growth of the total supply, contributing to scarcity.

Historically, these halving events have been followed by substantial price increases:

- 2012 Halving: Occurred in November 2012 when the block reward reduced from 50 to 25 Bitcoins. In the subsequent year, Bitcoin's price surged from around \$12 to over \$1,000.
- 2016 Halving: Occurred in July 2016, reducing the reward to 12.5 Bitcoins. By December 2017, the price soared to nearly \$20,000.
- 2020 Halving: Took place in May 2020, cutting the reward to 6.25 Bitcoins. By April 2021, Bitcoin reached an all-time high above \$60,000. [1]

Modern economic theory, particularly the Subjective Theory of Value, posits that the value of a good is determined by individual preferences and marginal utility rather than production costs. Economists like Ludwig von Mises [13] and Friedrich Hayek [7] argued that prices are set by what consumers are willing to pay, not by the costs incurred in production. It is hard to believe that solely the programmatic decrease in production is a sole factor of Bitcoin price increase over time.

Bitcoin's finite supply, capped at 21 million coins, introduces scarcity as a fundamental concept that also contributes to its value. The scarcity principle suggests that limited availability of a resource increases its value when demand is constant or growing (Mankiw, 2020) [11]. Nakamoto highlighted this aspect in the original Bitcoin whitepaper:

> "The steady addition of a constant amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended."
>
> — Satoshi Nakamoto, 2008 [15]

By comparing Bitcoin to gold, Nakamoto underscores how finite supply and the difficulty of production create scarcity, enhancing Bitcoin's value. This aligns with the Scarcity Principle in economics, which states that limited supply, coupled with high demand, leads to increased prices (Samuelson & Nordhaus, 2010) [19]. The controlled supply growth ensures that Bitcoin remains a scarce asset, potentially leading to price increases as demand grows (Böhme et al., 2015) [4].

While production difficulty and finite supply establish the foundational economic conditions for potential price appreciation, they are not sufficient on their own to drive actual market prices upward. The key element that bridges these factors is user perception and expectation, aligning with the Subjective Theory of Value.

Market Sentiment and Price Formation: Studies have shown that investor sentiment significantly impacts Bitcoin's price. Garcia et al. (2014) found that social media and word-of-mouth contribute to Bitcoin's price dynamics, indicating that user perception plays a critical role [5].

Expectations and Asset Valuation: Keynes (1936) discussed how market participants' expectations about future prices can influence current prices, a concept known as "animal spirits" [9]. In the Bitcoin market, positive expectations about future scarcity and production difficulty lead to increased current demand.

Network Effects and Adoption: Metcalfe's Law suggests that the value of a network grows with the square of its users. As more users believe in Bitcoin's value and adopt it, the utility and perceived value increase exponentially (Metcalfe, 2013) [12].

4.2 Extension to Other Cryptocurrencies and the Role of Proof of Stake

While Bitcoin's economic model, driven by production difficulty, finite supply, and user perception, has led to significant price appreciation, other blockchain technologies have not followed the same trajectory. Cryptocurrencies such as Ethereum (following its transition to Proof of Stake), Solana, Avalanche etc. have adopted Proof of Stake (PoS) consensus economic mechanisms to enhance efficiency, reduce transaction fees, and offer extended computing capabilities (Buterin, 2014 [3]; Yakovenko, 2018 [21]; Team Avalanche, 2020 [2]). Despite their technological advancements, these PoS-based cryptocurrencies have not consistently experienced the same magnitude of price increases as Bitcoin.

This discrepancy suggests that the economic drivers underlying PoS tokenomics differ from those of Bitcoin's Proof of Work (PoW) model. In PoW systems like Bitcoin, the security and issuance of new coins are directly linked to the expenditure of external resources — namely computational power and electricity — aligning with the Cost of Production Theory of Value. In contrast, PoS systems rely on validators staking their existing tokens to secure the network and validate transactions, with rewards distributed based on the amount staked (King & Nadal, 2012 [10]; Saleh, 2021 [18]). This shift means that the production cost in PoS networks is not tied to external resource expenditure in the same way, potentially diminishing the applicability of production cost as a determinant of value.

Moreover, many PoS cryptocurrencies do not have a strictly capped finite supply like Bitcoin, which reduces the scarcity effect that significantly contributes to Bitcoin's value (Mankiw, 2020) [11]. Without mechanisms such as increasing production difficulty and halving events, PoS tokens may lack the inherent economic pressures that drive price increases in the Bitcoin network. User perception also plays a critical role; the belief that Bitcoin's mining difficulty and finite supply will lead to price appreciation fuels demand, aligning with the Subjective Theory of Value (Menger, 18719; Shiller, 201710). In PoS networks, differing economic incentives and security models may influence user expectations and perceptions differently, affecting demand and price formation.

This disparity highlights that the economic drivers contributing to Bitcoin's price increases — namely production costs tied to resource expenditure and enforced scarcity — are not directly replicated in PoS cryptocurrencies. Consequently, PoS tokenomics, up to now, do not follow the same underlying economic drivers as Bitcoin for their cryptocurrency value. Understanding these differences is crucial for analyzing the valuation and investment potential of various cryptocurrencies within the broader blockchain ecosystem.

In this work we try to propose a tokenomics system for POS blockchain that would be as close to Bitcoin's POW economic model, while correcting some of the Bitcoin's shortfalls, such as transaction fees that tend to go higher as price of the native cryptocurrency of the blockchain (Bitcoin price in this case) increases.

5 Separation of Tokens

In Acki Nacki there are two tokens: a network token and a computation token.

The separation allows us to have two distinctive properties of Acki Nacki that is not possible under a one common token design.

In Proof-of-Stake systems the security of the network and the participation incentives are largely attributed to the network token price increase over time. This is achieved by bending the Supply/Demand curve in favor of Demand.

It can be done by increasing the Token Utility and Decreasing the Supply. But when there is only one token which is used for both security guarantees and network transaction fees its utility will be hampered by its increasing price, which happens with every blockchain we know. To tackle this problem Bitcoin is promoting the Lightning network, Ethereum is trying to balance the gas price and Solana is processing large amounts of transactions with very low fees. We don't believe any of these approaches work over time and we see problems with all of them: Lightning Network adoption rate is faltering, Ethereum transactions are so expensive, most of the people using L2 networks to transact Ethe and Solana can't regulate its network usage effectively leading to network stoppage and spam.

We take a different approach by introducing two interconnected tokens separately created to optimally perform each of the functions: network usage and network security.

Computation token, called **SHELL** — is designed to pay for network usage, and **NACKL** coin — designed to guarantee network security.

NACKL Coin — is used for Staking and provides a claim for a share of Shell revenues therefore will accumulate value over time.

SHELL Token — designed in such a way that its price will never increase, it can only decrease, but will eventually correct itself, as described in more details below (see Section "SHELL — Equal or Less" 14).

6 NACKL Tokenomics

6.1 Proof Of Stake

In Acki Nacki there is no predetermined Stake Interest rate. Simple and clear — there are no staking rewards. Like in Bitcoin the rewards are paid for Network Participation which comprises several activities like Block Production, Block Verification and Transaction Processing, but unlike Bitcoin all the Rewards are distributed proportionally between all Network Participants within a common epoch (see Section "Rewards" 7). If Network Participants are not performing according to current Acki Nacki Network rules or boundaries they may be excluded from the network, penalized or slashed depending on the type of rule they violate. This is according to the main idea of Proof-of-Stake Protocols [10].

6.2 Delegation

Acki Nacki is trying to avoid delegation of stakes as much as possible. There are special mechanisms in place to make it not economical or not secure to delegate NACKL Token for staking by other Block Keepers: Block Keeper Epoch contract only accepts messages signed by a Block Keeper private key, therefore making it impossible to create decentralized pools and perform staking delegation. Of course Block Keepers can run off-chain services to obtain stakes from investors, but this is no longer a network concern.

Instead there is a special mechanism to include regular participants into a protocol without a need to become a Block Keeper and have special server equipment etc. (see Section "Mobile Verifiers" 11) Yet it is important to mention that it's not based on staking pools or delegation either, as mobile verifiers perform very particular and real security verification contributing to network security guarantees.

6.3 Fairness

Acki Nacki is "fair" protocol, where fairness is defined per Pass and Shi [16]: "A blockchain protocol has n-approximate fairness if, with overwhelming probability, any honest subset controlling f fraction of the compute power is guaranteed to get at least a $(1 - n)f$ fraction of the blocks in a sufficiently long window".

The fairness in Acki Nacki is achieved by the following logical construction:

Each Validator receives proportional reward regardless of if they produced blocks or not. The reward depends solely on their honest participation in the network as described below. Thus the network participants are not rewarded specifically for producing the block but for participating in all stages of block livecycle from the creation and up to the finality.

Because in the Asynchronous transaction model the particular arrangement of incoming external transactions does not determine the execution order of subsequent internal transactions, there is no apparent calculable profit extraction (MEV) opportunity exists for a Block Producer. For example frontrunning is highly improbable and can instead result in a loss. Since the chances of such loss are high enough no rational actor should try. In Acki Nacki therefore there is simply no game to play around MEV extraction, which in turn makes the equal block rewards model possible.

Therefore in Acki Nacki the fairness model that usually applies to most of the networks does not hold true [8]. We therefore can consider Acki Nacki a "fair protocol" at least according to the above definition.

7 Rewards

The reward in Acki Nacki is divided among three groups of network participants: Block Keepers, Mobile Verifiers, and Block Managers. The reward for each Block Keeper is awarded based on individual **Epochs** of a certain duration. It is precomputed before the start of the individual epoch and is awarded at the end of that epoch. The distribution of the reward within each group of network participants is described in more detail in the subsections "Block Keeper Reward" 7.3, "Mobile Verifier Reward" 11.2, and "Block Manager Reward" 12.1.

7.1 General Reward

7.1.1 Bitcoin Analysis

- t — Time — Time since the network launch in seconds
- T_B — Bitcoin Total Supply — The total number of Bitcoins to be mined
- $R_B(t)$ — Bitcoin Block Reward — The amount of Bitcoin mined per block in the Bitcoin network. The Initial Bitcoin Block Reward denotes as $R_{B,0}$.
- $\Delta_{B,B}$ — Bitcoin Seconds per Block — The average block time in Bitcoin in seconds
- $\Delta_{B,H}$ — Bitcoin Delta Halving — The number of seconds that pass on average between halvings in the Bitcoin network, taking into account that, on average, a block is mined every $\Delta_{B,B}$ seconds.
- $M_B(t)$ — Bitcoin Total Mined Token Amount — The number of mined Bitcoins at time t
- $\tilde{M}_B(t)$ — Approximated Bitcoin Total Mined Token Amount — The approximation of $M_B(t)$ function by an exponential saturation function

Bitcoin's Total Supply $T_B = 21,000,000$. The Initial Block Reward $R_{B,0} = 50$. On average a block is mined every 10 minutes = 600 seconds $=: \Delta_{B,B}$. Every $\Delta_{B,H} = 210,000$ blocks = 126,000,000 seconds, the R_B is halved.

Thus, Bitcoin Total Mined Token Amount M_B over time t can be described by the following formula, where the first term represents the number of coins mined during the full periods up to the last halving, and the second term represents the number of coins mined after the last halving:

$$M_B(t) = \frac{R_{B,0}}{\Delta_{B,B}} \cdot \Delta_{B,H} \cdot \left(1 + \frac{1}{2} + \ldots + \frac{1}{2^{\lfloor t/\Delta_{B,H} \rfloor - 1}}\right) + \frac{R_{B,0}}{\Delta_{B,B}} \cdot \frac{1}{2^{\lfloor t/\Delta_{B,H} \rfloor}} \cdot (t - \lfloor t/\Delta_{B,H} \rfloor \cdot \Delta_{B,H}) \tag{1}$$

By simplifying the sum, we get:

$$M_B(t) = \frac{R_{B,0}}{\Delta_{B,B}} \cdot \Delta_{B,H} \cdot 2 \cdot \left(1 - \frac{1}{2^{\lfloor t/\Delta_{B,H} \rfloor}}\right) + \frac{R_{B,0}}{\Delta_{B,B}} \cdot \frac{1}{2^{\lfloor t/\Delta_{B,H} \rfloor}} \cdot (t - \lfloor t/\Delta_{B,H} \rfloor \cdot \Delta_{B,H}) \tag{2}$$

Note that if we let time t tend to infinity, we will obtain T_B:

$$\lim_{t \to +\infty} M_B = \lim_{t \to +\infty} \frac{R_{B,0}}{\Delta_{B,B}} \cdot \Delta_{B,H} \cdot 2 \cdot \left(1 - \frac{1}{2^{\lfloor t/\Delta_{B,H} \rfloor}}\right) = \frac{R_{B,0}}{\Delta_{B,B}} \cdot \Delta_{B,H} \cdot 2 = \frac{50}{600} \cdot 126000000 \cdot 2 = 21000000 = T_B \tag{3}$$

Thus, if we imagine that the reduction in block reward for mining Bitcoins happens not once every 4 years on average through halving, but continuously, with each block and by a small amount, we can approximate M_B with an exponential saturation function:

$$\tilde{M}_B(t) = T_B \cdot \left(1 - e^{-t \cdot \ln 2/\Delta_{B,H}}\right) \tag{4}$$

7.1.2 NACKL Adaptation

- t — Time — Time since the network launch in seconds
- T — Total Supply — The total number of tokens to be minted
- $M(t)$ — Expected Total Minted Token Amount — The expected number of minted tokens at time t

- $R(t)$ — Expected General Reward per Second — Expected reward for network participation per second for all network participants at time t
- τ — Total Token Minting Time — The expected time for minting the last fraction of token in seconds
- K_M — Total Minted Token Amount Function Coefficient — The parameter regulating the decay rate of the Total Minted Token Amount function
- $R_{\text{BK}}(t)$ — Expected Total Base Block Keeper Reward per Second — Fraction of the reward $R(t)$ allocated to Block Keepers
- $R_{\text{MV}}(t)$ — Expected Total Mobile Verifier Reward per Second — Fraction of the reward $R(t)$ allocated to Mobile Verifiers
- $R_{\text{BM}}(t)$ — Expected Total Block Manager Reward per Second — Fraction of the reward $R(t)$ allocated to Block Managers
- $K_{R,\text{BK}}$ — Base Block Keeper Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Keepers
- $K_{R,\text{MV}}$ — Mobile Verifier Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Mobile Verifiers
- $K_{R,\text{BM}}$ — Block Manager Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Managers
- $M_{\text{BK}}(t)$ — Expected Block Keeper Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Keepers
- $M_{\text{MV}}(t)$ — Expected Mobile Verifier Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Mobile Verifiers
- $M_{\text{BM}}(t)$ — Expected Block Manager Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Managers
- $\tilde{R}_{\text{BK}}(t)$ — Adjusted Total Base Block Keeper Reward per Second — Adjusted reward for network participation per second for all Block Keepers
- $\tilde{R}_{\text{MV}}(t)$ — Adjusted Total Mobile Verifier Reward per Second — Adjusted reward for network participation per second for all Mobile Verifiers
- $\tilde{R}_{\text{BM}}(t)$ — Adjusted Total Block Manager Reward per Second — Adjusted reward for network participation per second for all Block Managers

The curve of the total number of minted NACKL coins is an adaptation of the Bitcoin curve with some modifications.

Firstly, just like in the approximated curve of the total number of mined Bitcoins, there will be no halving at specific time intervals in Acki Nacki. Instead, the block reward will decrease continuously, allowing the total number of mined NACKL coins to be described by an exponential saturation function.

Secondly, the theoretical M_B curve differs from the real one because, in reality, Bitcoin blocks were not mined exactly every 10 minutes due to various factors, such as variations in the network hash rate, mining difficulty adjustments, and changes in miner activity. Additionally, the process of finding a hash is random, which can lead to deviations from the average hash finding time. Therefore, when selecting parameters, we will rely on the real curve of the total number of mined Bitcoin tokens.

Thirdly, even in the real curve, we will not rely on the total number of mined Bitcoins during the first year after the launch of the Bitcoin network, as the network's hash rate was quite low at that time, and the actual time to find a block took significantly longer than the usual 10 minutes we are familiar with.

Fourthly, we will slightly modify the curve form of $T_B \cdot \left(1 - e^{-t \cdot \ln 2 / \Delta_{B,H}}\right)$ so that by the time τ after the launch of the Acki Nacki network, exactly $10,400,000,000$ tokens will have been minted. Unlike Bitcoin, where with the current algorithm, the last fraction of a Bitcoin will never be fully mined. However the Bitcoin reward will decrease infinitely, eventually becoming practically equal to zero which is essentially equivalent to stopping token minting after the time τ.

Considering all the changes, we obtain the following Expected Total Minted Token Amount function for NACKL token and the following General Reward, which is calculated in seconds to eliminate dependency on blocks and thereby prevent potential spam activity:

$$u_M = -\frac{1}{\tau} \cdot \ln\left(\frac{K_M}{K_M + 1}\right) \tag{5}$$

$$M(t) = \begin{cases} T \cdot (1 + K_M) \cdot (1 - \exp(-u_M \cdot t)) & \text{if } 0 \leq t \leq \tau \\ T & \text{if } t > \tau \end{cases} \tag{6}$$

$$R(t) = \begin{cases} T \cdot (1 + K_M) \cdot (\exp(-u_M \cdot t) - \exp(-u_M \cdot (t+1))) & \text{if } 0 \leq t \leq \tau - 1 \\ 0 & \text{if } t > \tau - 1 \end{cases} \tag{7}$$

The resulting reward and total minted token amount are divided among the three groups of network participants in predetermined proportions, calculated based on each group's contribution to the network's operation.

$$\begin{gathered} K_{R,\mathrm{BK}} = 0.675 \\ K_{R,\mathrm{MV}} = 0.225 \\ K_{R,\mathrm{BM}} = 0.1 \\ K_{R,\mathrm{BK}} + K_{R,\mathrm{MV}} + K_{R,\mathrm{BM}} = 1 \end{gathered} \tag{8}$$

$$\begin{array}{ccccc} R_{\mathrm{BK}} & = & K_{R,\mathrm{BK}} \cdot R & = & 0.675 \cdot R \\ R_{\mathrm{MV}} & = & K_{R,\mathrm{MV}} \cdot R & = & 0.225 \cdot R \\ R_{\mathrm{BM}} & = & K_{R,\mathrm{BM}} \cdot R & = & 0.1 \cdot R \end{array} \tag{9}$$

$$\begin{array}{ccccc} M_{\mathrm{BK}} & = & K_{R,\mathrm{BK}} \cdot M & = & 0.675 \cdot M \\ M_{\mathrm{MV}} & = & K_{R,\mathrm{MV}} \cdot M & = & 0.225 \cdot M \\ M_{\mathrm{BM}} & = & K_{R,\mathrm{BM}} \cdot M & = & 0.1 \cdot M \end{array} \tag{10}$$

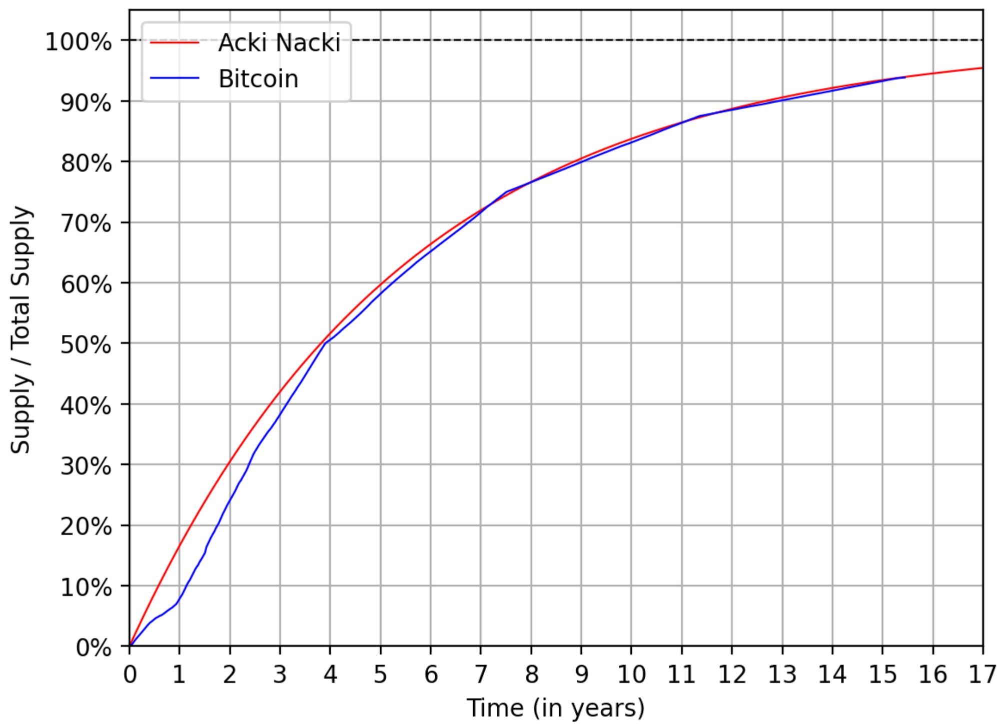

Figure 1: Comparison plot of Bitcoin and Acki Nacki NACKL token supplies per year

Up until this point, we have discussed only the **Expected** Total Minted Token Amount. The **Actual** Total Minted Token Amount will differ from the expected due to various factors, which will be explained later. To minimize the error between the actual and expected values of this parameter, we introduce a special reward adjustment mechanism (see Section "Automated Reward Adjustment" 13), which initializes and updates the adjusted reward values $\tilde{R}_{\mathrm{BK}}$, $\tilde{R}_{\mathrm{MV}}$, $\tilde{R}_{\mathrm{BM}}$. Therefore, in the following reward calculation formulas for Block Keepers (eq. (12)), Mobile Verifiers (eq. (26)), and Block Managers (eq. (36)), the adjusted parameters $\tilde{R}_{\mathrm{BK}}$, $\tilde{R}_{\mathrm{MV}}$, $\tilde{R}_{\mathrm{BM}}$ will be used instead of the expected parameters R_{BK}, R_{MV}, R_{BM}.

7.2 Reputation Coefficient

For Block Keepers, the reward they receive from R is called the Base Reward. This is because on top of the fair block reward, each Block Keeper may receive a Reputation Premium Reward called the Reputation Coefficient. This reward is calculated based on the time the particular Block Keeper, authenticated as Public Key of the cryptographic key pair, controlling the Block Keeper's wallet has continuously participated in a protocol and restaked their tokens.

The reputation multiplicator can provide much greater rewards than Base Reward, thus providing incentives for Block Keepers to keep uninterrupted network validation.

If a Block Keeper skips at least one epoch, their Reputation Coefficient is immediately reset to the minimum possible one.

- $\mathcal{R}$ — Reputation Coefficient — Additional rewards granted to a Block Keeper for continuous validation
- $t_{\mathcal{R}}$ — Block Keeper Reputation Time — The time during which the Block Keeper has been continuously running validation epochs
- $\mathcal{R}_{\min}$ — Minimal Reputation Coefficient
- $\mathcal{R}_{\max}$ — Maximal Reputation Coefficient
- $t_{\mathcal{R},\max}$ — Maximal Reputation Time — The time it takes for the Block Keeper to accumulate maximum reputation for continuous validation
- $K_{\mathcal{R}}$ — Adjustment Reputation Function Coefficient — The parameter regulating the rate of reputation growth over time

$$\mathcal{R}\left(t_{\mathcal{R}}\right) = \begin{cases} \mathcal{R}_{\min} + \dfrac{\left(\mathcal{R}_{\max} - \mathcal{R}_{\min}\right)}{1 - {K_{\mathcal{R}}}^{-1}} \cdot \left(1 - \exp\left(-\dfrac{\ln\left(K_{\mathcal{R}}\right)}{t_{\mathcal{R},\max}} \cdot t_{\mathcal{R}}\right)\right) & \text{if } t_{\mathcal{R}} < t_{\mathcal{R},\max} \\ \mathcal{R}_{\max} & \text{if } t_{\mathcal{R}} \geq t_{\mathcal{R},\max} \end{cases} \tag{11}$$

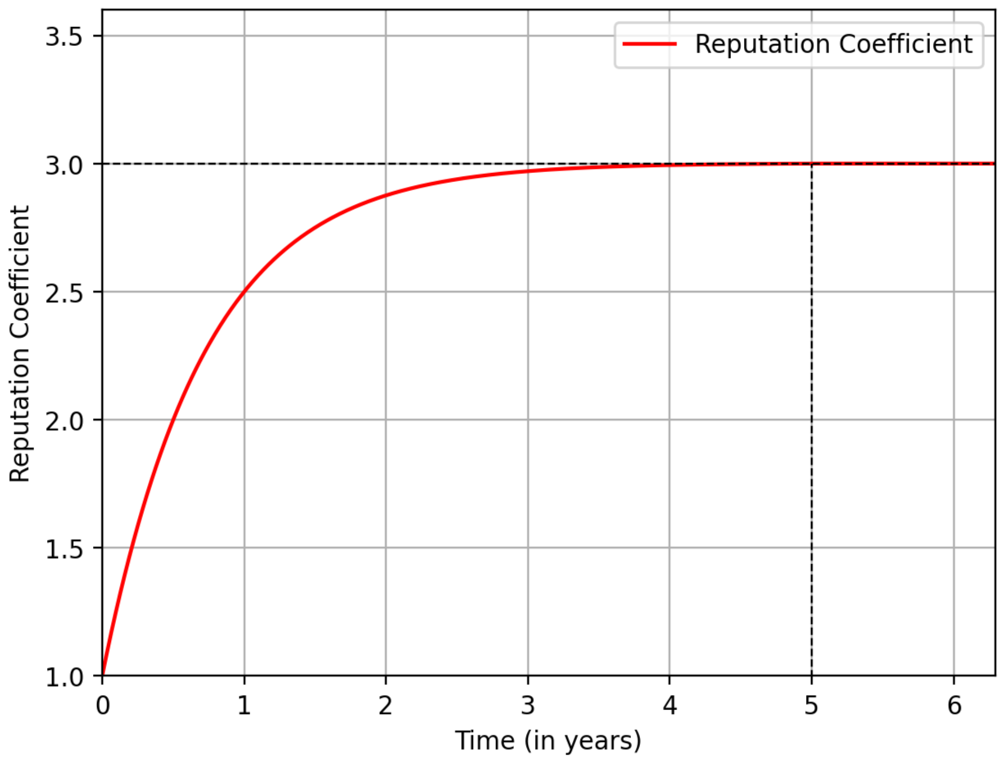

Figure 2: Plot of the Reputation Coefficient depending on the continuous validation time by a particular Block Keeper

7.3 Block Keeper Reward

As mentioned earlier, each network participant receives a reward for each epoch. For Block Keepers, we will refer to this epoch as the validation epoch.

We assume that if all network participants act honestly, the reward should be distributed fairly among them, regardless of whether the Block Keeper performs as a Block Producer, Acki-Nacki, or Block Keeper during that epoch[1]. Thus, the individual Block Keeper's reward will depend only on their stake and Reputation Coefficient.

Therefore, the Block Keeper's reward function r_{BK} will be calculated as follows:

- $r_{\text{BK}}(t)$ — Block Keeper Reward per Second — The reward earned by a Block Keeper per second of validation, depending on their stake and current Reputation Coefficient
- $\tilde{R}_{\text{BK}}(t)$ — Adjusted Total Base Block Keeper Reward per Second — Adjusted reward for network participation per second for all Block Keepers
- s_{BK} — Block Keeper Stake — The specific amount of tokens that a Block Keeper has staked the latest time in order to participate in validation
- S_{BK} — Total Block Keeper Stake — The sum of all Block Keeper stakes s_{BK} at time
- $\mathcal{R}$ — Reputation Coefficient — Additional rewards granted to a Block Keeper for continuous validation
- $t_{\mathcal{R}}$ — Block Keeper Reputation Time — The time during which the Block Keeper has been continuously running validation epochs
- t_{val} — Validation Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular validation epoch
- $r_{\text{BK},e}(t)$ — Block Keeper Reward per Validation Epoch — The reward received by a Block Keeper per one validation epoch
- $\Delta_{\text{BK},e}$ — Block Keeper Epoch Duration — The duration of one validation epoch in seconds
- $\tilde{N}_{\text{BK}}$ — Actual Block Keeper Number — The number of Block Keepers in the last block

[1]If Block Keepers are acting dishonestly their economic penalty for doing so is taken care by Slashing conditions of the consensus protocol and is outside of the scope of this paper

- $\tilde{M}_{\text{BK}}$ — Actual Block Keeper Total Minted Token Amount — The actual amount of tokens earned by all Block Keepers at time t
- T — Total Supply — The total number of tokens to be minted
- $K_{R,\text{BK}}$ — Base Block Keeper Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Keepers

$$r_{\text{BK}}\left(t,\, t_{\mathcal{R}},\, s_{\text{BK}},\, S_{\text{BK}}\right) = \tilde{R}_{\text{BK}}(t) \cdot \frac{s_{\text{BK}}|_t}{S_{\text{BK}}|_t} \cdot \mathcal{R}(t_{\mathcal{R}}) \tag{12}$$

7.3.1 Block Keeper Epoch Reward

Since a Block Keeper receives a reward at the end of each validation epoch, let us convert the reward per second of validation into a reward per epoch.

To ensure that each Block Keeper can easily calculate their reward for the Validation epoch at the start of the epoch, we lock the parameters s_{BK}, S_{BK}, and $\mathcal{R}$ at the beginning of the epoch and don't change them during the epoch. Since the number of Block Keepers in the network remains approximately constant during a Block Keeper's epoch, the case where the parameters s_{BK} and s_{BK} are fixed at the start of the epoch is practically identical to the case where these parameters are dynamically recalculated throughout the epoch. In other words, for a reasonable Block Keeper, it is disadvantageous to choose the moment when he starts an epoch to maximize their reward, as the time spent waiting will cause them to lose more reward than they could potentially earn, and they will also reset their accumulated Reputation Coefficient. Additionally, since the Maximal Reputation Coefficient accumulates over a much longer period of time than the duration of a validation epoch, it does not make practical sense to recalculate it during an epoch. It is sufficient to update the value of the Reputation Coefficient when transitioning from one epoch to the next. For the same reason, it does not make practical sense to recalculate the value of the $\tilde{R}_{\text{BK}}(t)$ function during the validation epoch.

It is also important to note that in Acki Nacki, there is no pre-distribution of tokens before the network launch, as the network is decentralized. As a result, there will be no tokens to stake during the first two epochs, because only by the beginning of the third epoch (or more precisely, around the middle of the second epoch) will the tokens exit the cooling period of the first epoch and become available to network participants for staking (see Section "Block Keeper Min Stake" 9). Therefore, it must be taken into account that the reward for the first two validation epochs will not depend on the Block Keepers' stake, but will instead depend on Block Keeper number.

Thus, let us calculate the reward per the validation epoch for a single Block Keeper:

$$r_{\text{BK},e}\left(t_{\text{val}},\, t_{\mathcal{R}},\, \Delta_{\text{BK},e},\, s_{\text{BK}},\, S_{\text{BK}},\, \tilde{N}_{\text{BK}}\right) = \begin{cases} \tilde{R}_{\text{BK}}(t_{\text{val}}) \cdot \dfrac{1}{\tilde{N}_{\text{BK}}|_{t_{\text{val}}}} \cdot \mathcal{R}(t_{\mathcal{R}}|_{t_{\text{val}}}) \cdot \Delta_{\text{BK},e} & \text{if } \tilde{M}_{\text{BK}}|_{t_{\text{val}}} = 0 \\[2ex] \tilde{R}_{\text{BK}}(t_{\text{val}}) \cdot \dfrac{s_{\text{BK}}|_{t_{\text{val}}}}{S_{\text{BK}}|_{t_{\text{val}}}} \cdot \mathcal{R}(t_{\mathcal{R}}|_{t_{\text{val}}}) \cdot \Delta_{\text{BK},e} & \text{if } 0 < \tilde{M}_{\text{BK}}|_{t_{\text{val}}} < T \cdot K_{R,\text{BK}} \\[2ex] 0 & \text{if } \tilde{M}_{\text{BK}}|_{t_{\text{val}}} \geq T \cdot K_{R,\text{BK}} \end{cases} \tag{13}$$

If a Block Keeper, for any reason, validates longer than the expected duration of a single epoch, additional time spent as a Block Keeper will be added to the parameter $\Delta_{\text{BK},e}$.

8 Free Float

Acki Nacki largely follows a well-researched Bitcoin free float model. We define Bitcoin's Free Float as the number of tokens that have been in circulation over the last year. While in Bitcoin the free float average is around 40% (see Fig. 4), Acki Nacki will theoretically experience exponential saturation growth from nearly 0 to $\frac{1}{3}$, while $(1 - F)$ of tokens (up to a maximum of $\frac{2}{3}$) will be locked in staking.

Let's construct the exponential saturation function for the Free Float (as a percentage of the total minted token amount):

- $F_{\max}$ — Maximal Free Float Fraction — Maximal fraction of Free Float of Total Minted Token Amount

- $F(t)$ — Free Float Fraction — The current fraction of Free Float of Total Minted Token Amount
- τ — Total Token Minting Time — The expected time for minting the last fraction of token
- K_F — Free Float Function Coefficient — The parameter regulating the decay rate of the F function

$$u_F = -\frac{1}{\tau} \cdot \ln\left(\frac{K_F}{1+K_F}\right) \tag{14}$$

$$F(t) = \begin{cases} F_{\max} \cdot (1+K_F) \cdot (1 - \exp(-u_F \cdot t)) & \text{if } 0 \leq t \leq \tau \\ F_{\max} & \text{if } t > \tau \end{cases} \tag{15}$$

If Block Keepers do not restake their stakes and withdraw them, thereby increasing the Free Float, the reward remains fixed. Meaning the remaining Block Keepers will start receiving more rewards, which will reduce their motivation to withdraw their stakes even if the token price decreases. Because the min stake will decrease, allowing other Block Keepers to stake their tokens if they couldn't do so before (see Section "Block Keeper Min Stake" 9).

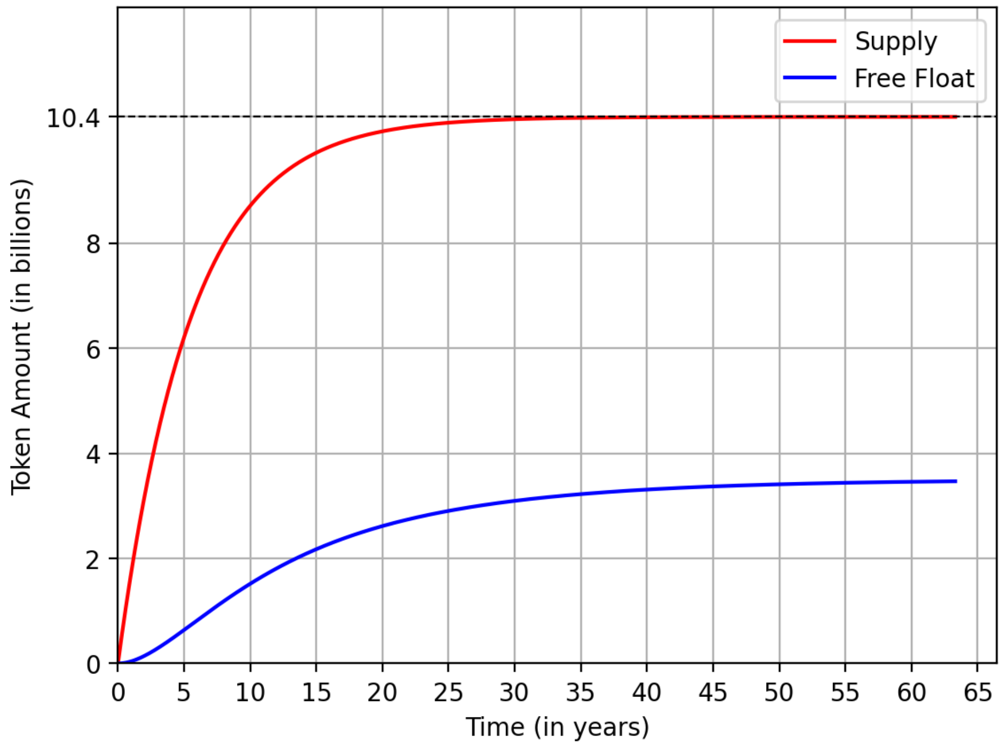

Figure 3: Plot of the total number of minted tokens and free float (in tokens) over time

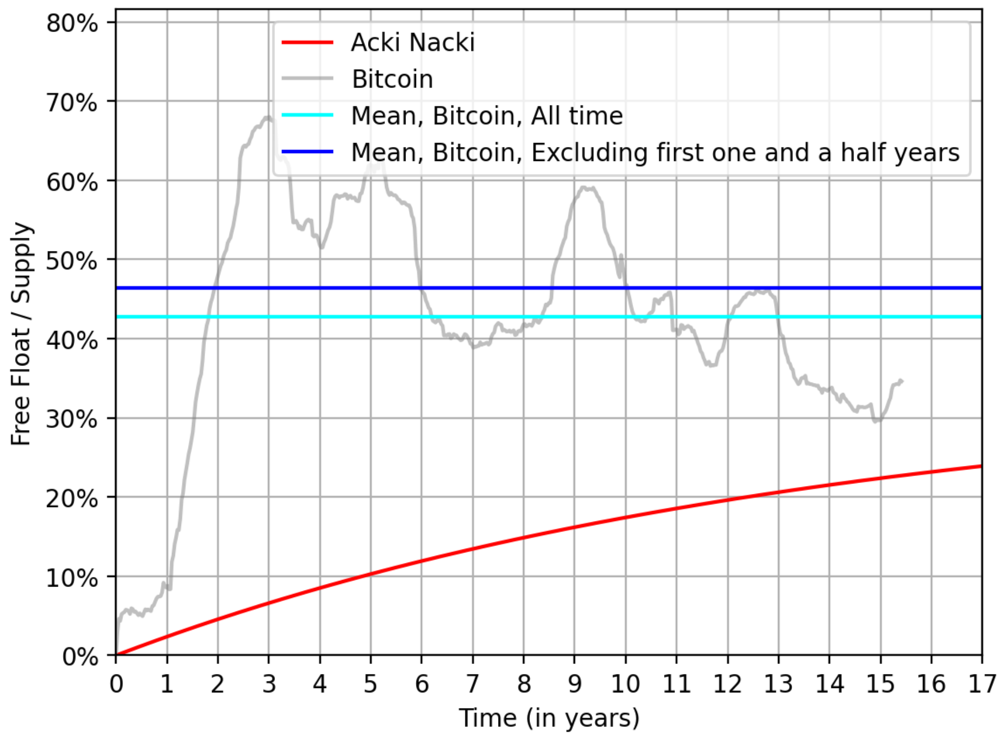

Figure 4: Comparison plot of Bitcoin and Acki Nacki NACKL Free Floats (as a percentage of the **current Supply**) per year

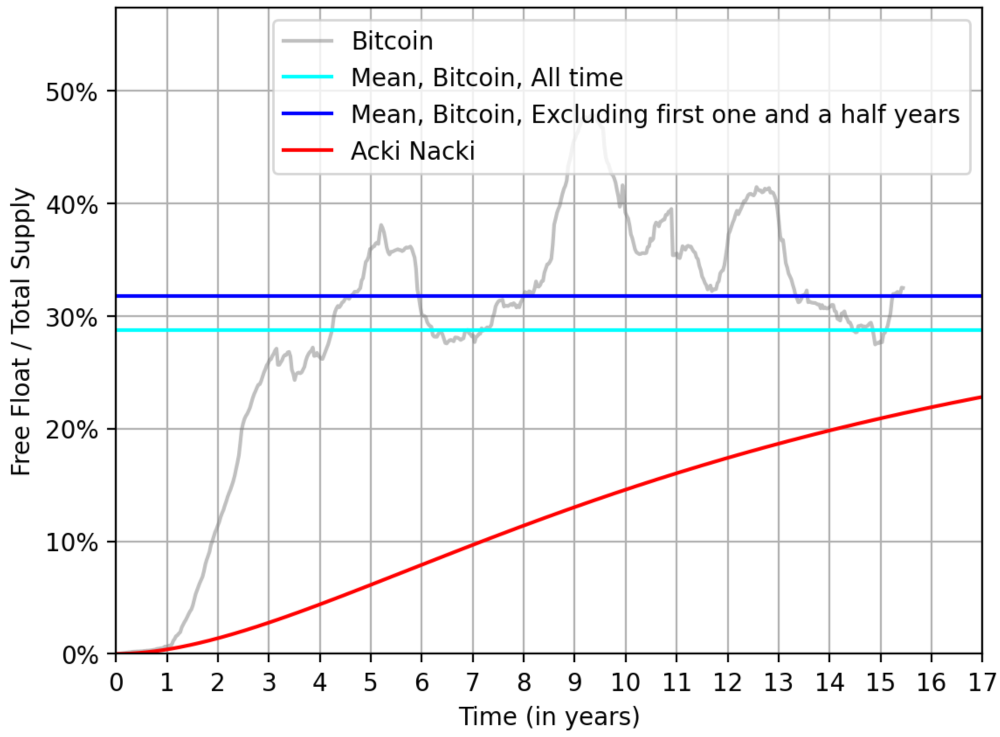

Figure 5: Comparison plot of Bitcoin and Acki Nacki NACKL Free Floats (as a percentage of the **Total Supply**) per year

9 Block Keeper Min Stake

Because Acki Nacki is a scalable computational network, the execution load parameter plays a significant role in its tokenomics.

Acki Nacki is a multithreaded execution environment. Threads grow when computation demand on the network grows, more Block Keepers are required to process the network load. Usually one would argue the rewards should grow to lure more Block Keepers into the network. But that won't work because of a “spam attack”. In the Spam Attack the Block Keeper may create spam transactions to artificially increase network load so that threads are multiplied to inflate the block rewards. And since in Acki Nacki the payment for computations (electricity) is stable or less (see Section "SHELL — stable or less coin" 14) the arbitrage between the compute expense and block reward is always beneficial to the attacker. Therefore no increase of the Block Reward is possible. Instead the minimum required stake is lowered automatically. Thus allowing lower barriers to entry for new Block Keepers to provide their computing power to participate in a slice of a block rewards. And since Reputation Coefficient plays a much greater role in the Block reward for each Block Keeper over time, it provides a lucrative opportunity for profitable network participation.

- $S(t)$ — Expected Total Staked Token Amount — The expected total number of tokens staked in the network at time t
- $M(t)$ — Expected Total Minted Token Amount — The expected number of minted tokens at time t
- $F(t)$ — Free Float Fraction — The current fraction of Free Float of Total Minted Token Amount
- $\tilde{M}_{\text{BK,av}}$ — Actual Block Keeper Total Available Token Amount — The total amount of tokens available for staking. Calculated as the difference between the current total minted token amount and the total number of tokens burned during the whole slashing.
- $s_{\text{BK,base}}(t)$ — Base Minimal Block Keeper Stake — Minimal Stake when the current number of Block Keepers equals the necessary number of Block Keepers
- $s_{\text{BK,min}}(t)$ — Minimal Block Keeper Stake — Current minimal Block Keeper stake depending on the particular difference between the current number of Block Keepers and the required number of Block Keepers
- t_{val} — Validation Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular validation epoch
- t_{stk} — Staking Time — The time in seconds that has passed from the moment the network was launched until the registration for the next epoch and setting the current stake
- $N_{\text{BK,req}}$ — Needed Block Keeper Number — The number of Block Keepers required in the network at time t according to the number of threads
- $\tilde{N}_{\text{BK}}$ — Actual Block Keeper Number — The number of Block Keepers in the last block
- K_s — Adjustment Minimal Block Keeper Stake Function Coefficient — The parameter regulating the growth of the $s_{\text{BK,min}}(t)$ function

The total number of staked tokens is easily calculated from the known total number of minted tokens and the current free float:

$$S(t) = M(t) \cdot (1 - F(t)) \tag{16}$$

Since each validation epoch for a Block Keeper requires time to verify the correctness of all Block Keepers' actions, half of the staked tokens is locked in the current validation cycle, and the other half of the staked tokens is locked in the cooling period for slashing calculation. Therefore, each Block Keeper effectively needs to have two stakes to validate.

In Acki Nacki, not only Block Keepers stake but also Mobile Verifiers and Block Managers (see Subsections "Mobile Verifier Min Stake" 11.4, "Block Manager Min Stake" 12.3). As mentioned earlier, the actual total minted token amount and the expected minted token amount will differ slightly. Additionally, the pre-determined reward distribution for each type of network participant (see eq. (8)) will also vary slightly. Therefore, let the stake placed by a Block Keeper depend on the actual amount of tokens available for staking $\tilde{M}_{\text{BK}}$ and the current Free Float Fraction parameter F.

Let's calculate $s_{\text{BK,base}}$ and $s_{\text{BK,min}}$ for Block Keepers, taking into account that the minimum stake should be calculated at the start of the validation epoch:

$$s_{\text{BK,base}}\left(t_{\text{stk}},\, N_{\text{BK,req}},\, \tilde{M}_{\text{BK,av}}\right) = \begin{cases} 0, & \text{if } \tilde{M}_{\text{BK,av}}|_{t_{\text{stk}}} = 0 \\ \dfrac{\tilde{M}_{\text{BK,av}}|_{t_{\text{stk}}} \cdot (1 - F(t_{\text{stk}}))}{2} \cdot \dfrac{1}{N_{\text{BK,req}}|_{t_{\text{stk}}}}, & \text{if } \tilde{M}_{\text{BK,av}}|_{t_{\text{stk}}} \neq 0 \end{cases} \tag{17}$$

$$u_s = -\frac{\ln\left(\frac{K_s}{K_s+1}\right)}{N_{\mathrm{BK,req}}|_{t_{\mathrm{stk}}}} \tag{18}$$

$$u_{N,\mathrm{BK}} = 2 \cdot N_{\mathrm{BK,req}}|_{t_{\mathrm{stk}}} - \tilde{N}_{\mathrm{BK}}|_{t_{\mathrm{stk}}} \tag{19}$$

$$s_{\mathrm{BK,min}}(t_{\mathrm{stk}}) := s_{\mathrm{BK,min}}(t_{\mathrm{stk}},\ \tilde{N}_{\mathrm{BK}},\ N_{\mathrm{BK,req}}) \tag{20}$$

$$s_{\mathrm{BK,min}}(t_{\mathrm{stk}}) = \begin{cases} s_{\mathrm{BK,base}}|_{t_{\mathrm{stk}}} \cdot (1+K_s) \cdot \left(1 - \exp\left(-u_s \cdot \tilde{N}_{\mathrm{BK}}|_{t_{\mathrm{stk}}}\right)\right) & \text{if } 0 \leq \tilde{N}_{\mathrm{BK}}|_{t_{\mathrm{stk}}} \leq N_{\mathrm{BK,req}}|_{t_{\mathrm{stk}}} \\ \\ s_{\mathrm{BK,base}}|_{t_{\mathrm{stk}}} \cdot \left(2 - (1+K_s) \cdot (1 - \exp(-u_s \cdot u_{N,\mathrm{BK}}))\right) & \text{if } \tilde{N}_{\mathrm{BK}}|_{t_{\mathrm{stk}}} > N_{\mathrm{BK,req}}|_{t_{\mathrm{stk}}} \end{cases} \tag{21}$$

It is worth noting that, based on the equation (21), it can be seen that the curve $s_{\mathrm{BK,min}}\left(\tilde{N}_{\mathrm{BK}}\right)$ starts at the point $(0, 0)$. This implies that the curve accounts for the potential decrease in the number of Block Keepers even down to zero. However, in reality, the network includes a parameter for the minimum number of Block Keepers, which prevents the number of Block Keepers $\tilde{N}_{\mathrm{BK}}$ from falling below a certain threshold.

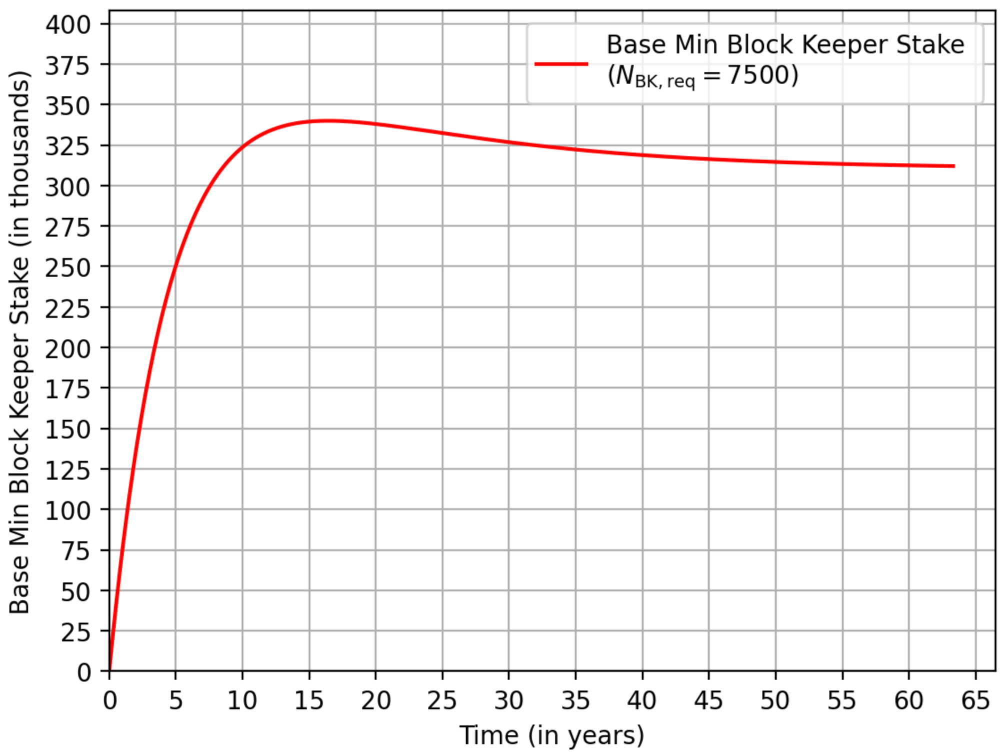

Figure 6: Plot of $s_{\mathrm{BK,base}}\,(t)$ over time since the network's launch with the necessary number of Block Keepers set to $7,500$

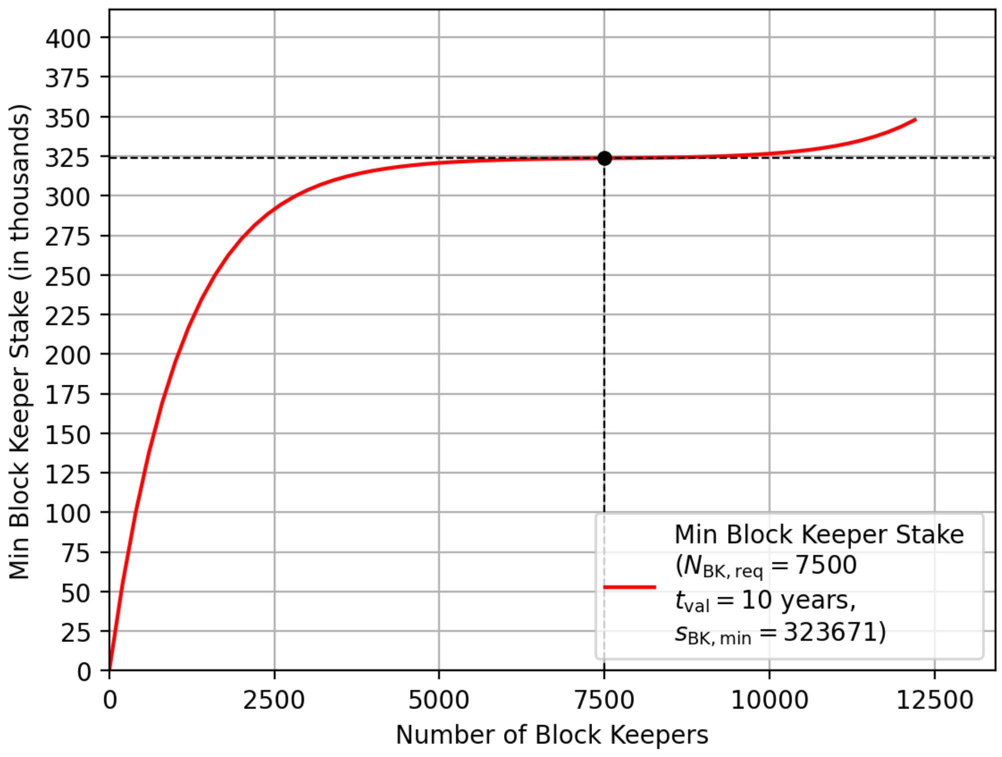

Figure 7: Plot of $s_{\mathrm{BK,min}}\left(\tilde{N}_{\mathrm{BK}}\right)$ depending on the current number of Block Keepers 10 years after the network's launch with the necessary number of Block Keepers equal to $7,500$

10 Expected APR for Block Keepers

- y — Year Number — The year number since the network launch at which we calculate the $R_{y,\%}$
- $R_{y,\%}(y)$ — Annual Percentage Reward — A measure of the yearly earnings expressed as a percentage of the average amount of tokens staked during the year
- $M_y(y)$ — Annual Minted Token Amount — An expected number of tokens minted over the year
- $S_y(y)$ — Annual Average Staked Token Amount — The average amount of tokens staked during the year
- σ_y — Seconds In Year — Average number of seconds in a year
- $M(t)$ — Expected Total Minted Token Amount — The expected number of minted tokens at time t
- $S(t)$ — Expected Total Staked Token Amount — The expected total number of tokens staked in the network at time t

While we are not keen to use terms like Annual Percentage Reward while talking about Acki Nacki staking, it is still important to provide such indicative calculations on the rewards Block Keeper receive for performing Network Participation work in comparison with NACKL Stake they provide as security bond. Please note that we omit all direct Block Keeper operation costs as they are compensated by SHELL Token as described below.

We propose the following $R_{y,\%}$ calculation. $R_{y,\%}$ is the number of tokens minted in one year divided by the average number of tokens staked during the year. y refers to the year number that closes a time period of one year. For instance, when $y = 1$, the $R_{y,\%}(y)$ will be calculated for the first 12 months since the network launch.

$$R_{y,\%}(y) = \frac{M_y(y)}{S_y(y)} = \frac{M(y \cdot \sigma_y) - M([y-1] \cdot \sigma_y)}{\dfrac{1}{\sigma_y} \displaystyle\int\limits_{[y-1]\cdot\sigma_y}^{y\cdot\sigma_y} S(t)\,\mathrm{d}t} \tag{22}$$

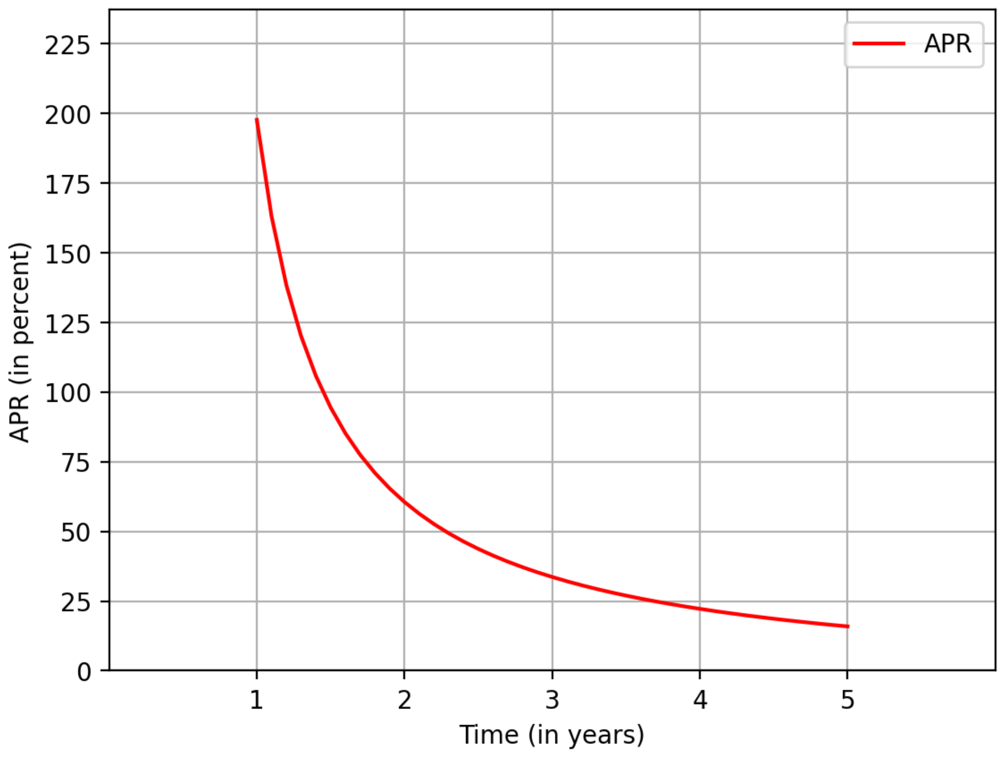

Figure 8: APR plot for the first 5 years after network launch

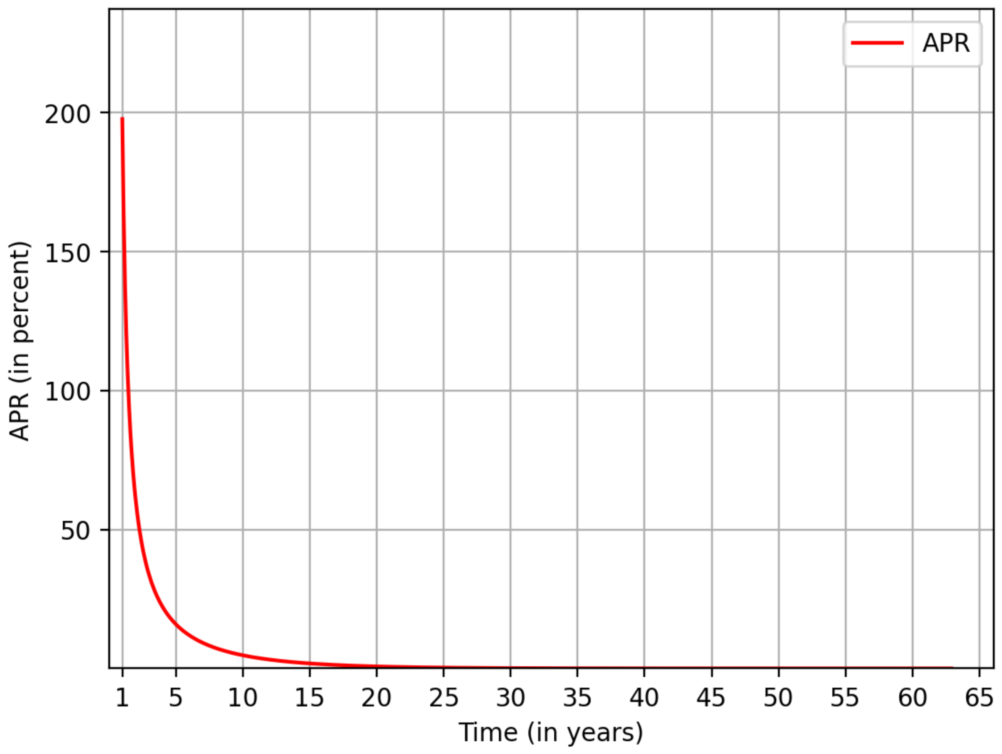

Figure 9: APR plot over time

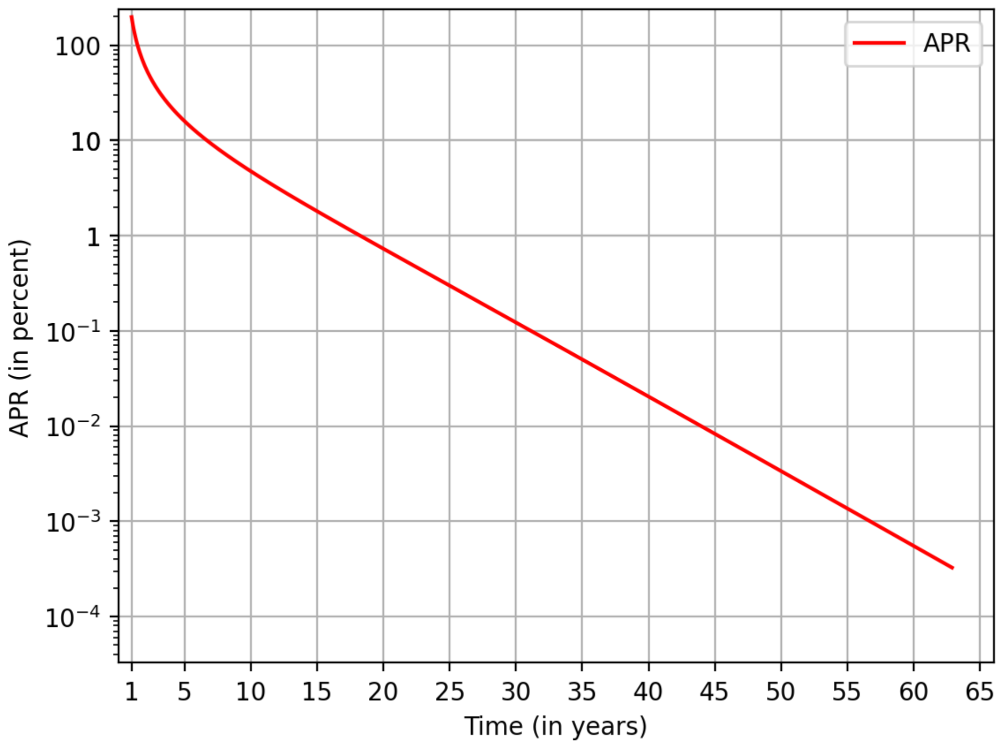

Figure 10: APR Plot with a logarithmic Y-axis over time

11 Mobile Verifiers

11.1 Motivation

Ideally, we would want a protocol that everyone can participate in without a need to run expensive server hardware. That would dramatically increase network security and decentralization. From the other side such a network would not be very performant, fast and scalable because of network and computing power limitation of mobile devices.

To solve this we introduce the Mobile Verifier role to Acki Nacki. A mobile user would not need to validate every block on the network, which would be technically impossible, but instead such a user could participate in the protocol as a Verifier by validating transactions in subtrees of accounts, occasionally. Since there is no way to know when such a user would choose to Verify and what, it would provide additional security guarantees to the network, dramatically decreasing the probability of attack on top of the already great security guarantees of the main Acki Nacki Protocol.

- $\tilde{N}_{\text{BK}}$ — Actual Block Keeper Number — The number of Block Keepers in the last block
- v — Acki-Nacki Number — The average number of Acki-Nacki per block
- A — Attestation Number — The number of attestations required for block finalization
- $N_{\text{BK,mal}}$ — Malicious Block Keeper Number — The expected number of Malicious Block Keepers
- p — Successful Attack Probability — Successful attack probability in a single attempt
- N_{MV} — Mobile Verifier Number — The number of Mobile Verifiers, who meet the reward eligibility condition
- $N_{\text{MV,mal}}$ — Malicious Mobile Verifiers Number — Expected number of Malicious mobile verifiers
- λ_{MV} — Mobile Verifier Verification Frequency — The average fraction of transactions verified by a single Mobile Verifier in a block
- p_{MV} — Mobile Verifier Successful Attack Probability — Successful attack probability in a single attempt with Mobile Verifiers
- $r_{\text{MV}}(t)$ — Mobile Verifier Reward per Second — The reward earned by a Mobile Verifier per second of verification
- $\mathcal{B}$ — Boost Coefficient — A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier based on their position in the sorted in ascending order list of all Mobile Verifiers by the number of Boosts. The sum of $\mathcal{B}$ for all Mobile Verifiers equals 1.

- G — Epoch Game Score — The score of a Mobile Verifier in the online game for one epoch
- $\mathcal{B}_i^{\mathrm{cl}}$ — Cluster Boost Coefficient — A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier cluster i based on their position in the sorted in ascending order list of all Mobile Verifiers clusters
- G_i^{cl} — Epoch Cluster Game Score — The score of all Mobile Verifiers belonging the cluster i in the online game for one epoch
- N_{cl} — Cluster Number — The current number of clusters consisting of Mobile Verifiers
- $R_{\mathrm{MV},i}^{\mathrm{cl}}(t)$ — Cluster Reward per Second — The reward allocated to the cluster i at time t
- $\tilde{R}_{\mathrm{MV}}(t)$ — Adjusted Total Mobile Verifier Reward per Second — Adjusted reward for network participation per second for all Mobile Verifiers
- $\Delta_{\mathrm{MV}}^{\mathrm{cl},i}$ — Mobile Verifier Cluster Number — The number of Mobile Verifiers belonging cluster i

Referring to "ACKI NACKI: a Probabilistic Proof-of-Stake consensus protocol with fast finality and parallelization" [6], in the case where only Block Keepers are present, the probability of an attack on a single block is expressed by the following formula:

$$p\left(\tilde{N}_{\mathrm{BK}},\, v,\, A,\, N_{\mathrm{BK,mal}}\right) = \left(1 - \frac{v}{\tilde{N}_{\mathrm{BK}} - 1}\right)^{A - N_{\mathrm{BK,mal}}} \tag{23}$$

For the reference, next Figs. are illustrating the successful attack probability from a number of Malicious network participants for Bitcoin, pBFT, and Acki Nacki protocols with a total of 1000 Block Keepers.

To calculate the successful attack probability in Bitcoin, we use the commonly accepted number of blocks for probabilistic 'finality', which is 6. For calculating the successful attack probability in Acki Nacki, we use the number of Acki-Nacki set to 40 and the number of Attestations set to 80.

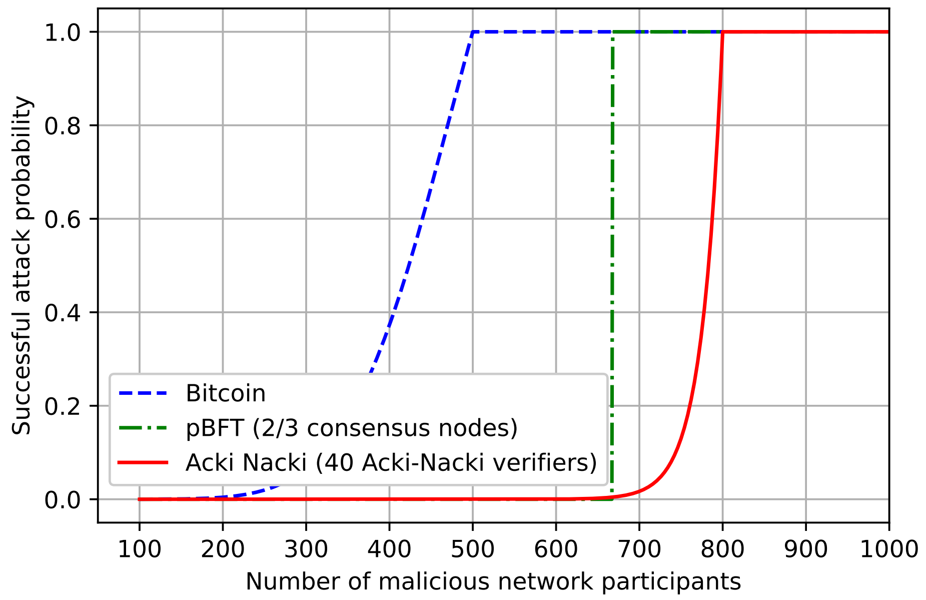

Figure 11: Comparison of successful attack probabilities in Bitcoin, pBFT and Acki Nacki

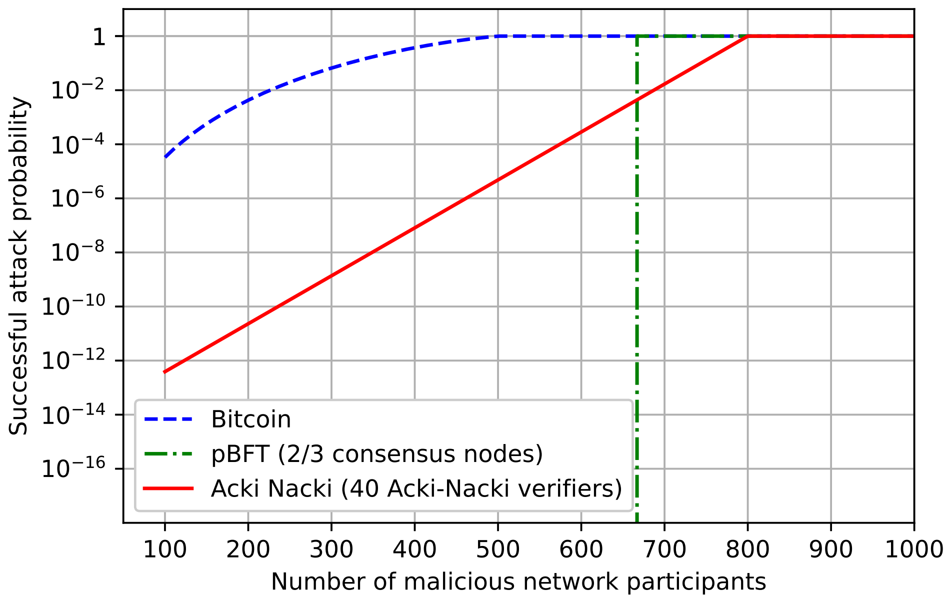

Figure 12: Fig. 11 with log-scaled y-axis

If we denote by λ_{MV} the average fraction of transactions verified by a single Mobile Verifier in a block, the probability of an attack on a block consists of the following logical conjunction of events: "no honest surviving BK has become Acki-Nacki" [6] and "no honest Mobile Verifier has verified a single Malicious transaction in the block." In this logical conjunction, we consider the case of a single Malicious transaction in the block, as it represents the optimal approach for Malicious network participants to attack the block.

Thus, the probability of the second event in the logical conjunction is equal to $(1-\lambda_{\text{MV}})^{N_{\text{MV}}-N_{\text{MV,mal}}}$, and accordingly the probability of an attack on the block, taking into account the presence of Mobile Verifiers, is expressed by the following formula:

$$p_{\text{MV}}\left(\tilde{N}_{\text{BK}},\, v,\, A,\, N_{\text{BK,mal}},\, \lambda_{\text{MV}},\, N_{\text{MV}},\, N_{\text{MV,mal}}\right) = \left(1-\frac{v}{\tilde{N}_{\text{BK}}-1}\right)^{A-N_{\text{BK,mal}}} \cdot (1-\lambda_{\text{MV}})^{N_{\text{MV}}-N_{\text{MV,mal}}} \tag{24}$$

11.2 Mobile Verifier Reward

Mobile Verifiers will compete in an online game that involves scoring points during an epoch and earning Boosts, which together determine the fraction of the block reward they will receive.

A Mobile Verifier receives a reward only if their wallet holds at least $s_{\text{MV,min}}$ tokens (see Subsection "Mobile Verifier Min Stake" 11.4).

Earning Boosts comes down to securing a position in an ordered list of Mobile Verifiers, ranked based on the number of their Boosts. Each place is then converted into a Boost Coefficient $\mathcal{B}$. There will be many ways for using Boosts in different open mining mechanics. Therefore, we need the sum of Boost Coefficients across all Mobile Verifiers equals 1.

Ideally, to distribute the entire reward per second among all Mobile Verifiers, we would use the following formula for the reward per second of Mobile Verifier i: $\dfrac{G_i \cdot \mathcal{B}_i}{\sum\limits_{j=1}^{N_{\text{MV}}} G_j \cdot \mathcal{B}_j}$. However, it is impractical to store the parameters G and $\mathcal{B}$ for each Mobile Verifier due to their large quantity. Therefore, we use clustering of Mobile Verifiers based on the number of Boosts they have earned, assign each cluster i a Boost Coefficient $\mathcal{B}_i^{\text{cl}}$ and the total score earned by the cluster G_i^{cl}. Then, we allocate the reward per second for a specific Mobile Verifier as a fraction of the reward for the cluster to which it belongs.

The reward per second of cluster i is expressed by the following formula:

$$R^{\mathrm{cl}}_{\mathrm{MV},i}(t) = \tilde{R}_{\mathrm{MV}}(t) \cdot \frac{G^{\mathrm{cl}}_i \cdot \mathcal{B}^{\mathrm{cl}}_i}{\sum\limits_{j=1}^{N_{\mathrm{cl}}} G^{\mathrm{cl}}_j \cdot \mathcal{B}^{\mathrm{cl}}_j} \tag{25}$$

The reward per second of a Mobile Verifier belonging to cluster i is expressed by the following formula:

$$r_{\mathrm{MV}}(t) = R^{\mathrm{cl}}_{\mathrm{MV},i}(t) \cdot \frac{G}{G^{\mathrm{cl}}_i} \tag{26}$$

11.3 Boost Coefficient

Our task will be to determine $\mathcal{B}^{\mathrm{cl}}$ for each Mobile Verifier cluster. To do this, we will construct the function $f_{\mathcal{B}}(x)$, whose integral from 0 to 1 equals 1. Each cluster will have a Boost Coefficient equal to the integral of this function over the interval determined by their quantile in the list of all Mobile Verifier clusters sorted in ascending order by the number of Boosts and the number of Mobile Verifiers in this cluster. The function will be an exponential curve consisting of several sub-curves such that:

1. The domain of the curve will be $\mathrm{Dom}(f) = [0, 1]$, allowing us to distribute the Total Mobile Verifiers Reward regardless of the number of Mobile Verifiers.
2. The integral over the entire domain of the curve equals 1, so we can divide the Total Mobile Verifiers Reward among all Mobile Verifiers.
3. - The first 30% of Mobile Verifiers will receive almost no reward.
   - The middle 40% will receive 30% of the total reward.
   - The last 30% with the most Boosts will receive 70% of the total reward.

11.3.1 Form of the Exponential Curve

An exponential curve with a growth coefficient k, passing through the points (x_1, y_1) and (x_2, y_2), is defined as follows:

$$f(x) = y_1 + \frac{(y_2 - y_1)\left(\exp\left(k \cdot \dfrac{x - x_1}{x_2 - x_1}\right) - 1\right)}{\exp(k) - 1} \tag{27}$$

Thus, we obtain a function with the following input parameters:

- Dot 1 $= (x_1, y_1)$ — The leftmost point of the first sub-curve
- Dot 2 $= (x_2, y_2)$ — The connection point between the first and second sub-curves
- Dot 3 $= (x_3, y_3)$ — The connection point between the second and third sub-curves
- Dot 4 $= (x_4, y_4)$ — The rightmost point of the third sub-curve
- k_1 — the growth coefficient of the first sub-curve
- k_2 — the growth coefficient of the second sub-curve
- k_3 — the growth coefficient of the third sub-curve

$$f_{\mathcal{B}}\left(x,\,x_1,\,y_1,\,x_2,\,y_2,\,x_3,\,y_3,\,x_4,\,y_4,\,k_1,\,k_2,\,k_3\right)=\begin{cases} y_1+(y_2-y_1)\cdot\dfrac{\left(\exp\left(k_1\cdot\dfrac{x-x_1}{x_2-x_1}\right)-1\right)}{\exp\left(k_1\right)-1} & \text{if } x_1\le x\le x_2\\ y_2+(y_3-y_2)\cdot\dfrac{\left(\exp\left(k_2\cdot\dfrac{x-x_2}{x_3-x_2}\right)-1\right)}{\exp\left(k_2\right)-1} & \text{if } x_2\le x\le x_3\\ y_3+(y_4-y_3)\cdot\dfrac{\left(\exp\left(k_3\cdot\dfrac{x-x_3}{x_4-x_3}\right)-1\right)}{\exp\left(k_3\right)-1} & \text{if } x_3\le x\le x_4\end{cases}\tag{28}$$

Let us denote these sub-curves as I, II, and III.

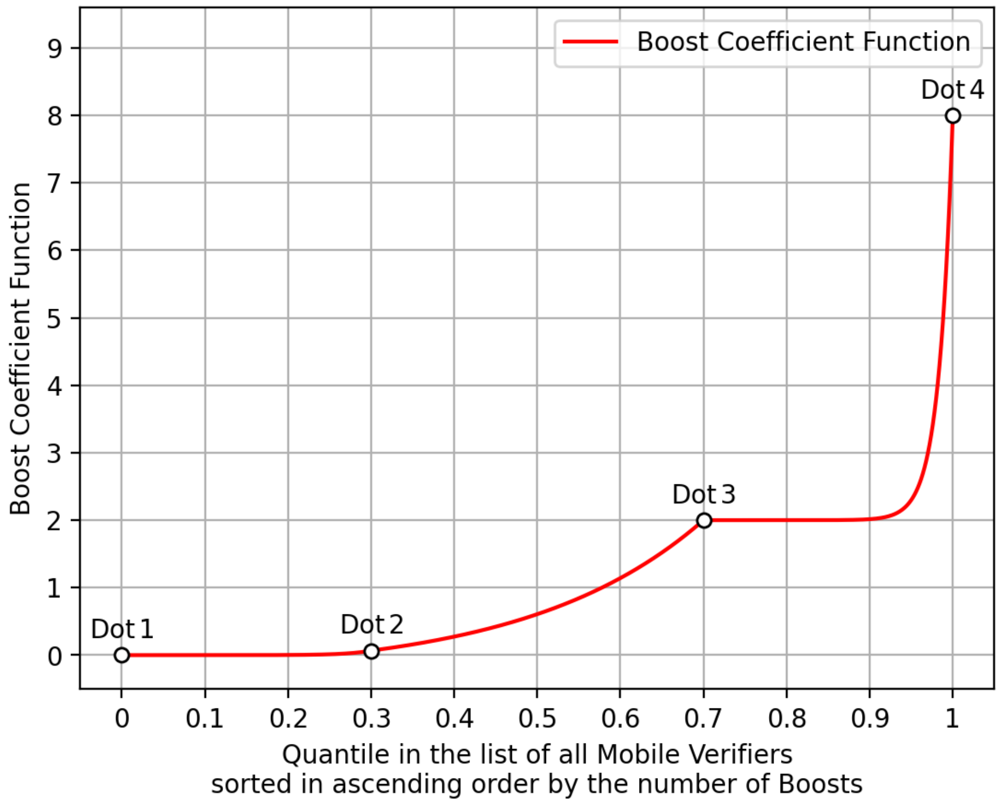

Figure 13: Boost Coefficient Function

11.3.2 Calculation of Parameters for the Piecewise Exponential Curve

As mentioned earlier, let $x_1=0$, $x_2=0.3$, $x_3=0.7$, and $x_4=1$.

We will empirically choose the following parameters for the curves: $k_1=10$, $y_3=2$, $y_4=8$.

The curve starts at the point $y_1=0$. This means that a Mobile Verifier with the fewest Boosts receives almost no reward.

Now we need to find the parameters y_2, k_2, and k_3. To do this, we will calculate the integral for each sub-curve and, based on point 3, equate these integrals to the following values:

- $\int\limits_{x_1}^{x_2} \mathrm{I}\,\mathrm{d}x=q_1:=0.002$, meaning the first 30% of Mobile Verifiers will collectively receive 0.2% of the total reward.
- $\int\limits_{x_2}^{x_3} \mathrm{II}\,\mathrm{d}x=q_2:=0.298$, meaning the middle 40% of Mobile Verifiers will collectively receive 29.8% of the total

reward.

- $\int\limits_{x_3}^{x_4} \mathrm{III}\,\mathrm{d}x = q_3 := 0.7$, meaning the top 30% of Mobile Verifiers with the most Boosts will collectively receive 70% of the total reward.

Let's calculate these definite integrals:

$$\int\limits_{x_1}^{x_2} \mathrm{I}\,\mathrm{d}x = \int\limits_{x_1}^{x_2} y_1 + (y_2 - y_1) \cdot \frac{\left(\exp\left(k_1 \cdot \dfrac{x - x_1}{x_2 - x_1}\right) - 1\right)}{\exp\left(k_1\right) - 1}\,\mathrm{d}x = x_2 y_1 - x_1 y_1 + \frac{(\exp(k_1) - 1 - k_1)\,(x_2 - x_1)\,(y_2 - y_1)}{(\exp(k_1) - 1)\,k_1} = q_1$$

$$\int\limits_{x_2}^{x_3} \mathrm{II}\,\mathrm{d}x = \int\limits_{x_2}^{x_3} y_2 + (y_3 - y_2) \cdot \frac{\left(\exp\left(k_2 \cdot \dfrac{x - x_2}{x_3 - x_2}\right) - 1\right)}{\exp\left(k_2\right) - 1}\,\mathrm{d}x = x_3 y_2 - x_2 y_2 + \frac{(\exp(k_2) - 1 - k_2)\,(x_3 - x_2)\,(y_3 - y_2)}{(\exp(k_2) - 1)\,k_2} = q_2$$

$$\int\limits_{x_3}^{x_4} \mathrm{III}\,\mathrm{d}x = \int\limits_{x_3}^{x_4} y_3 + (y_4 - y_3) \cdot \frac{\left(\exp\left(k_3 \cdot \dfrac{x - x_3}{x_4 - x_3}\right) - 1\right)}{\exp\left(k_3\right) - 1}\,\mathrm{d}x = x_4 y_3 - x_3 y_3 + \frac{(\exp(k_3) - 1 - k_3)\,(x_4 - x_3)\,(y_4 - y_3)}{(\exp(k_3) - 1)\,k_3} = q_3 \tag{29}$$

Let's construct a system of three equations for the three unknowns y_2, k_2, and k_3:

$$\begin{cases} x_2 y_1 - x_1 y_1 + \dfrac{(\exp(k_1) - 1 - k_1)\,(x_2 - x_1)\,(y_2 - y_1)}{(\exp(k_1) - 1)\,k_1} = q_1 \\[2ex] x_3 y_2 - x_2 y_2 + \dfrac{(\exp(k_2) - 1 - k_2)\,(x_3 - x_2)\,(y_3 - y_2)}{(\exp(k_2) - 1)\,k_2} = q_2 \\[2ex] x_4 y_3 - x_3 y_3 + \dfrac{(\exp(k_3) - 1 - k_3)\,(x_4 - x_3)\,(y_4 - y_3)}{(\exp(k_3) - 1)\,k_3} = q_3 \end{cases} \tag{30}$$

An analytical solution to this system of equations is impossible. Therefore, we substitute the values of the known parameters x_1, x_2, x_3, x_4, y_1, y_3, y_4, k_1, q_1, q_2, q_3 and obtain the numerical solution for the following values y_2, k_2, k_3:

$$\begin{aligned} y_2 &\approx 0.066696948409 \\ k_2 &\approx 1.894163612445 \\ k_3 &\approx 17.999995065464 \end{aligned} \tag{31}$$

Thus, we have obtained the curve with all known parameters.

11.3.3 Calculation of the Boost Coefficient for Different Clusters

Now, we need to calculate $\mathcal{B}_i^{\mathrm{cl}}$ from the known function $f_{\mathcal{B}}$. For this, we introduce the list containing the number of Mobile Verifiers in each cluster: $\Delta_{\mathrm{MV}}^{\mathrm{cl}} = \left\{\Delta_{\mathrm{MV}}^{\mathrm{cl},1}, \ldots, \Delta_{\mathrm{MV}}^{\mathrm{cl},N_{\mathrm{cl}}}\right\}$.

The Boost Coefficient of a Mobile Verifier cluster, who is in the i-th position in the list sorted in ascending order by the number of Boosts, will be calculated as the integral over the subinterval corresponding to this Mobile Verifier cluster.

$$\mathcal{B}_i^{\mathrm{cl}} = \int_{\sum_{j=1}^{i-1} \Delta_{\mathrm{MV}}^{\mathrm{cl},j}}^{\sum_{j=1}^{i} \Delta_{\mathrm{MV}}^{\mathrm{cl},j}} f_{\mathcal{B}}\left(x,\, x_1,\, y_1,\, x_2,\, y_2,\, x_3,\, y_3,\, x_4,\, y_4,\, k_1,\, k_2,\, k_3\right) \mathrm{d}x, \text{ where } i \in [1, N_{\mathrm{cl}}] \tag{32}$$

11.4 Mobile Verifier Min Stake

- t_{verf} — Verification Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Mobile Verifiers
- $s_{\mathrm{MV,min}}(t)$ — Minimal Mobile Verifier Stake — Current minimal particular Mobile Verifier stake
- $S(t)$ — Expected Total Staked Token Amount — The expected total number of tokens staked in the network at time t
- $M(t)$— Expected Total Minted Token Amount — The expected number of minted tokens at time t
- $r_{\mathrm{MV},\Sigma}$ — Mobile Verifier Reward History — Amount of tokens that Mobile Verifier have earned during their entire participation in the network
- $F(t)$ — Free Float Fraction — The current fraction of Free Float of Total Supply

The size of the Mobile Verifier's stake does not affect their reward (see eq. (26)), only the presence of the Min Stake on the Mobile Verifier's wallet matters. For Mobile Verifiers, there is no point in dynamically adjusting the stake based on the current number of Mobile Verifiers (as is done for Block Keepers (see Section "Block Keeper Min Stake" 9)), since it is impossible to determine the required number of Mobile Verifiers. Therefore, each Mobile Verifier's Min Stake will be unique and depend solely on the amount of tokens $r_{\mathrm{MV},\Sigma}$ they have earned during their entire participation in the network. The Min Stake of a Mobile Verifier will be a fraction of the $r_{\mathrm{MV},\Sigma}$ parameter, just as the Total Staked Token Amount $S(t)$ is a fraction of the Expected Total Minted Token Amount $M(t)$ (see eq. (16)), provided that the Mobile Verifier must have two stakes for the same reasons as for Block Keepers (see Section "Block Keeper Min Stake" 9).

For Mobile Verifiers (and Block Managers), just like for Block Keepers, the first two epochs do not require a stake. The difference lies in the fact that for Block Keepers, the first two epochs are counted from the network launch, while for Mobile Verifiers, the two epochs are counted from the moment the Mobile Verifier starts participating in the protocol. And only those epochs in which the Mobile Verifier performed their duties are taken into account. For instance, if a Mobile Verifier begins participating in the common Mobile Verifier epoch a year after the network launch, their first two epochs will not require a stake, as the parameter $r_{\mathrm{MV},\Sigma}$ will be zero.

$$s_{\mathrm{MV,min}}\left(t_{\mathrm{verf}},\, r_{\mathrm{MV},\Sigma}\right) = r_{\mathrm{MV},\Sigma}\big|_{t_{\mathrm{verf}}} \cdot \frac{(1 - F(t_{\mathrm{verf}}))}{2} \tag{33}$$

If a Mobile Verifier does not have enough tokens in their wallet to place the stake for the next epoch, they will stop receiving rewards and cease to be a Mobile Verifier. Becoming a Mobile Verifier again will require going through the process anew, but the description of this process is beyond the scope of this document.

11.5 Mobile Verifier Epoch Reward

- $r_{\mathrm{MV},e}(t)$ — Mobile Verifier Reward per Epoch — The reward received by a Mobile Verifier per one verification epoch
- t_{verf} — Verification Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Mobile Verifiers
- $\mathcal{B}$ — Boost Coefficient — A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier based on their position in the sorted in ascending order list of all Mobile Verifiers, who meet the reward eligibility condition, by the number of Boosts. The sum of $\mathcal{B}$ for all Mobile Verifiers equals 1.
- G — Epoch Game Score — The score of a Mobile Verifier in the online game for one epoch
- $\mathcal{B}_i^{\mathrm{cl}}$ — Cluster Boost Coefficient — A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier cluster i based on their position in the sorted in ascending order list of all Mobile Verifiers clusters

- G_i^{cl} — Epoch Cluster Game Score — The score of all Mobile Verifiers belonging the cluster i in the online game for one epoch
- N_{cl} — Cluster Number — The current number of clusters consisting of Mobile Verifiers
- $\tilde{R}_{\mathrm{MV}}(t)$ — Adjusted Total Mobile Verifier Reward per Second — Adjusted reward for network participation per second for all Mobile Verifiers
- $\Delta_{\mathrm{MV},e}$ — Mobile Verifier Epoch Duration — The duration of one epoch of Mobile Verifiers in seconds
- $\tilde{M}_{\mathrm{MV}}$ — Actual Mobile Verifier Total Minted Token Amount — The actual amount of tokens earned by all Mobile Verifiers at time t
- T — Total Supply — The total number of tokens to be minted
- $K_{R,\mathrm{MV}}$ — Mobile Verifier Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Mobile Verifiers

The epoch of Mobile Verifiers, unlike the epochs of Block Keepers, is common for all Mobile Verifiers. The first epoch starts when the network is launched. The current reward of the cluster i is updated after the epoch ends to prevent unnecessary continuous calculations, as the number of Boosts of Mobile Verifiers changes with high frequency and the scored points reset after the epoch finishing.

The reward per epoch for Mobile Verifier belonging cluster i is expressed by the following formula:

$$u_{\mathrm{MV},e} = \frac{G|_{t_{\mathrm{verf}}+\Delta_{\mathrm{MV,e}}} \cdot B_i^{\mathrm{cl}}|_{t_{\mathrm{verf}}+\Delta_{\mathrm{MV,e}}}}{\sum\limits_{j=1}^{N_{\mathrm{cl}}} G_j^{\mathrm{cl}}|_{t_{\mathrm{verf}}+\Delta_{\mathrm{MV,e}}} \cdot B_j^{\mathrm{cl}}|_{t_{\mathrm{verf}}+\Delta_{\mathrm{MV,e}}}} \tag{34}$$

$$r_{\mathrm{MV},e}\left(t_{\mathrm{verf}},\, \Delta_{\mathrm{MV},e},\, u_{\mathrm{MV},e}\right) = \begin{cases} \Delta_{\mathrm{MV},e} \cdot \tilde{R}_{MV}(t_{\mathrm{verf}} + \Delta_{\mathrm{MV},e}) \cdot u_{\mathrm{MV},e} & \text{if } 0 \leq \tilde{M}_{\mathrm{MV}}|_{t_{\mathrm{verf}}} < T \cdot K_{R,\mathrm{MV}} \\ 0 & \text{if } \tilde{M}_{\mathrm{MV}}|_{t_{\mathrm{verf}}} \geq T \cdot K_{R,\mathrm{MV}} \end{cases} \tag{35}$$

12 Block Managers

The primary function of Block Managers is to provide the user with a blockchain database and to process external messages. Block Managers receive a portion of the total block reward based on the number of external messages they process. The reward distribution is structured in such a way that spamming the network with external messages to increase rewards is not practically beneficial. This is because generating spam external messages requires certain resources, and the reward increase will slow down significantly if the number of external messages processed by a specific Block Manager exceeds the average number of processed messages across all Block Managers.

12.1 Block Manager Reward

Block Managers reward depends only on the number of external messages they processed.

- $r_{\mathrm{BM}}(t)$ — Block Manager Reward per Second — The reward earned by a Block Manager per second of management
- $\tilde{R}_{\mathrm{BM}}(t)$ — Adjusted Total Block Manager Reward per Second — Adjusted reward for network participation per second for all Block Managers
- $\mathcal{E}$ — External Messages Coefficient — the coefficient that determines the fraction of the reward allocated to a particular Block Manager based on their position in the sorted in ascending order list of all Block Managers by the number of processed external messages. The sum of $\mathcal{E}$ for all Block Managers equals 1.
- N_{BM} — Block Manager Number — The current number of Block Managers in the network

The reward for Block Managers is calculated using the following formula:

$$r_{\mathrm{BM}}\left(t,\, \mathcal{E}\right) = \tilde{R}_{\mathrm{BM}}(t) \cdot \mathcal{E}|_t \tag{36}$$

12.2 External Messages Coefficient

Let's determine $\mathcal{E}$ for each Block Manager. By analogy with the Boost Coefficient for Mobile Verifiers (see Subsection "Boost Coefficient" 11.3), we will construct the function $f_{\mathcal{E}}(x)$, whose integral from 0 to 1 equals 1. Each Block Manager will have an External Messages Coefficient equal to the integral of this function over the interval. Now, the interval is determined by their quantile in the list of all Block Managers sorted in ascending order by the number of processed external messages during the Block Manager epoch. The function will be a complex curve consisting of several sub-curves such that:

1. The domain of the curve will be $\mathrm{Dom}(f) = [0,1]$, allowing us to distribute the Total Block Managers Reward regardless of the number of Block Managers.
2. The integral over the entire domain of the curve equals 1, so we can divide the Total Block Managers Reward among all Block Managers.
3. - The first 10% of Block Managers will receive almost no reward.
   - The top 90% will receive almost the entire reward.

12.2.1 Form of the Curve

If we establish a direct proportionality between the reward received by a Block Manager and the number of transactions they process, some Block Managers may be incentivized to carry out a spam attack on the network with fake transactions to receive the entire reward. To prevent this, we designed the following $f_{\mathcal{E}}$ curve, based on the model we had developed for Mobile Verifiers.

- Dot 1 $= (x_1, y_1)$ — The leftmost point of the first sub-curve
- Dot 2 $= (x_2, y_2)$ — The connection point between the first and second sub-curves
- Dot 3 $= (x_3, y_3)$ — The rightmost point of the second sub-curve
- k_1 — the growth coefficient of the first sub-curve

$$f_{\mathcal{E}}\left(x,\, x_1,\, y_1,\, x_2,\, y_2,\, x_3,\, y_3,\, k_1\right) = \begin{cases} y_1 + (y_2 - y_1) \cdot \dfrac{\left(\exp\left(k_1 \cdot \dfrac{x - x_1}{x_2 - x_1}\right) - 1\right)}{\exp\left(k_1\right) - 1}, & x_1 \le x \le x_2 \\[2ex] \dfrac{y_3 - y_2}{x_3 - x_2} \cdot (x - x_2) + y_2, & x_2 \le x \le x_3 \end{cases} \tag{37}$$

Let us denote these sub-curves as I, II.

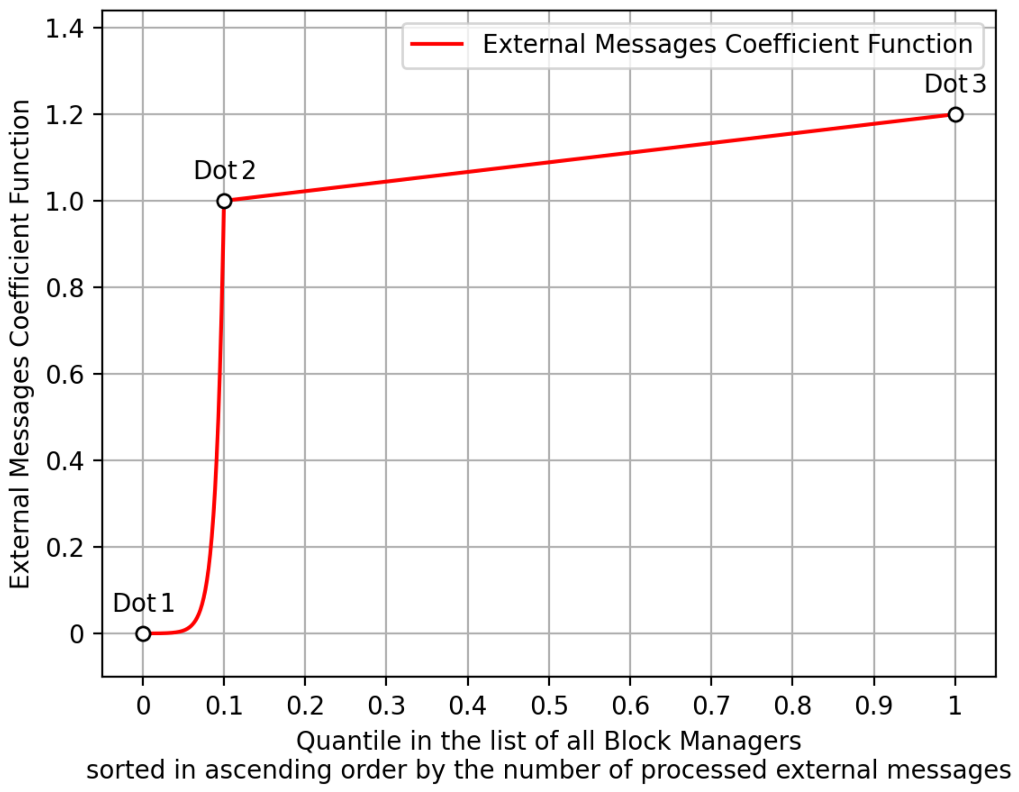

Figure 14: External Messages Coefficient Function

12.2.2 Calculation of Parameters for the Piecewise Curve

As mentioned earlier, let $x_1 = 0$, $x_2 = 0.1$, $x_3 = 1$.

To remove the incentive for a spam attack on the network, we analyzed the curve and empirically chose the following parameter for the curve: $y_3 = 1.2$.

The curve starts at the point $y_1 = 0$. This means that a Block Manager with the smallest number of processed external messages will receive almost no reward.

Now we need to find the parameters y_2, k_1. To do this, we will calculate the integral for each sub-curve and, based on point 3, equate these integrals to the following values:

- $\int\limits_{x_1}^{x_2} \mathrm{I}\,\mathrm{d}x = q_1 := 0.01$, meaning the first 10% of Block Managers will collectively receive 1% of the total reward.
- $\int\limits_{x_2}^{x_3} \mathrm{II}\,\mathrm{d}x = q_2 := 0.99$, meaning the top 90% of Block Managers will collectively receive 99% of the total reward.

Let's calculate these definite integrals:

$$\int\limits_{x_1}^{x_2} \mathrm{I}\,\mathrm{d}x = \int\limits_{x_1}^{x_2} y_1 + (y_2 - y_1) \cdot \frac{\left(\exp\left(k_1 \cdot \dfrac{x - x_1}{x_2 - x_1}\right) - 1\right)}{\exp(k_1) - 1}\,\mathrm{d}x = x_2 y_1 - x_1 y_1 + \frac{(\exp(k_1) - 1 - k_1)(x_2 - x_1)(y_2 - y_1)}{(\exp(k_1) - 1)\,k_1} = q_1$$

$$\int\limits_{x_2}^{x_3} \mathrm{II}\,\mathrm{d}x = \int\limits_{x_2}^{x_3} \left(\frac{y_3 - y_2}{x_3 - x_2} \cdot (x - x_2) + y_2\right)\,\mathrm{d}x = \frac{1}{2}(x_3 - x_2)(y_2 + y_3) = q_2 \tag{38}$$

Let's construct a system of two equations for the two unknowns y_2, k_1:

$$\begin{cases} x_2 y_1 - x_1 y_1 + \dfrac{\left(\exp(k_1) - 1 - k_1\right)(x_2 - x_1)(y_2 - y_1)}{\left(\exp(k_1) - 1\right) k_1} = q_1 \\ \\ \dfrac{1}{2}(x_3 - x_2)(y_2 + y_3) = q_2 \end{cases} \tag{39}$$

For simplicity, we directly substitute the known parameter values x_1, x_2, x_3, y_1, y_3, q_1, q_2 and obtain the following values for y_2, k_1:

$$\begin{aligned} y_2 &= 1 \\ k_1 &\approx 9.99544113 \end{aligned} \tag{40}$$

Thus, we have obtained the curve with all known parameters.

12.2.3 Calculation of the External Messages Coefficient for Different Numbers of Block Managers

Now, we need to calculate $\mathcal{E}$ from the known function $f_{\mathcal{E}}$. For this, we introduce the parameter for the number of Block Managers N_{BM}.

By analogy with the Mobile Verifiers, we define the External Messages Coefficient for the Block Manager who is in the i-th position in the list sorted in ascending order by the number of processed external messages, will be calculated as the integral over the subinterval corresponding to this Block Manager. In other words, we will divide the interval $[0, 1]$ into N_{BM} parts, and the i-th Manager will correspond to the interval $\left[\dfrac{i-1}{N_{\text{BM}}}, \dfrac{i}{N_{\text{BM}}}\right]$.

That is,

$$\mathcal{E}_i = \int\limits_{(i-1)/N_{\text{BM}}}^{i/N_{\text{BM}}} f_{\mathcal{E}}\left(x, x_1, y_1, x_2, y_2, x_3, y_3, k_1\right) \mathrm{d}x, \text{ where } i \in [1, N_{\text{BM}}] \tag{41}$$

12.3 Block Manager Min Stake

- t_{mng} — Management Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Block Managers
- $r_{\text{BM},\Sigma}$ — Block Manager Reward History — Amount of tokens that Block Manager have earned during their entire participation in the network
- $s_{\text{BM,min}}(t)$ — Minimal Block Manager Stake — Current minimal particular Block Manager stake
- $F(t)$ — Free Float Fraction — The current fraction of Free Float of Total Supply

Similarly to Mobile Verifiers (see Subsection "Mobile Verifier Min Stake" 11.4), the reward of a Block Manager does not depend on the amount of stake they place but only on the presence of the Min Stake. The Min Stake of each Block Manager is unique and depends solely on the amount of tokens they have earned during their participation in the network $r_{\text{BM},\Sigma}$. As with Block Keepers and Mobile Verifiers, two stakes are required to continuously participate in the network:

$$s_{\text{BM,min}}\left(t_{\text{mng}}, r_{\text{BM},\Sigma}\right) = r_{\text{BM},\Sigma}|_{t_{\text{mng}}} \cdot \frac{\left(1 - F(t_{\text{mng}})\right)}{2} \tag{42}$$

12.4 Block Manager Epoch Reward

- $r_{\text{BM},e}(t)$ — Block Manager Reward per Epoch — The reward received by a Block Manager per one management epoch
- t_{mng} — Management Epoch Start Time — The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Block Managers
- $\Delta_{\text{BM},e}$ — Block Manager Epoch Duration — The duration of one epoch of Block Managers in seconds

- $\mathcal{E}$ — External Messages Coefficient — the coefficient that determines the fraction of the reward allocated to a particular Block Manager based on their position in the sorted in ascending order list of all Block Managers by the number of processed external messages. The sum of $\mathcal{E}$ for all Block Managers equals 1.
- $\tilde{R}_{\text{BM}}(t)$ — Adjusted Total Block Manager Reward per Second — Adjusted reward for network participation per second for all Block Managers
- $\tilde{M}_{\text{BM}}$ — Actual Block Manager Total Minted Token Amount — The actual amount of tokens earned by all Block Managers at time t
- T — Total Supply — The total number of tokens to be minted
- $K_{R,\text{BM}}$ — Block Manager Reward Function Coefficient — The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Managers

Similarly to Mobile Verifiers (see Subsection "Mobile Verifier Epoch Reward" 11.5), the epoch of Block Managers is common for all Block Managers. $\mathcal{E}$ is calculated each time at the end of the epoch and reset after epoch finishing. This means that the reward of a Block Manager depends only on their position in the sorted in ascending order list of all Block Managers by the number of processed external messages at the end of the epoch.

$$r_{\text{BM},e}\left(t_{\text{mng}},\, \mathcal{E},\, \Delta_{\text{BM},e}\right) = \begin{cases} \tilde{R}_{\text{BM}}(t_{\text{mng}}) \cdot \mathcal{E}|_{t_{\text{mng}}+\Delta_{\text{BM},e}} \cdot \Delta_{\text{BM},e} & \text{if } 0 \leq \tilde{M}_{\text{BM}}|_{t_{\text{mng}}} < T \cdot K_{R,\text{BM}} \\ 0 & \text{if } \tilde{M}_{\text{BM}}|_{t_{\text{mng}}} \geq T \cdot K_{R,\text{BM}} \end{cases} \tag{43}$$

13 Automated Reward Adjustment

- $\tilde{R}_{\text{BK}}(t)$ — Adjusted Total Base Block Keeper Reward per Second — Adjusted reward for network participation per second for all Block Keepers
- $\tilde{R}_{\text{MV}}(t)$ — Adjusted Total Mobile Verifier Reward per Second — Adjusted reward for network participation per second for all Mobile Verifiers
- $\tilde{R}_{\text{BM}}(t)$ — Adjusted Total Block Manager Reward per Second — Adjusted reward for network participation per second for all Block Managers
- $M_{\text{BK}}(t)$ — Expected Block Keeper Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Keepers
- $\tilde{M}_{\text{BK}}$ — Actual Block Keeper Total Minted Token Amount — The actual amount of tokens earned by all Block Keepers at time t
- $\tilde{N}_{\text{BK}}$ — Actual Block Keeper Number — The number of Block Keepers in the last block
- $\mathcal{R}$ — Reputation Coefficient — Additional rewards granted to a Block Keeper for continuous validation
- s_{BK} — Block Keeper Stake — The specific amount of tokens that a Block Keeper has staked the latest time in order to participate in validation
- S_{BK} — Total Block Keeper Stake — The sum of all Block Keeper stakes s_{BK} at time
- $\mathcal{R}_{\text{avg},s}(t)$ — Stake Weighted Average Reputation Coefficient — The stake-weighted average value of the reputation coefficient across all Block Keepers in the network at time t
- $\Delta_{\tilde{R}_{\text{BK}},\text{min}}$ — Minimal Adjusted Total Base Block Keeper Reward Update Time — The minimum time between updates of the $\tilde{R}_{\text{BK}}$ parameter
- $\Delta_{\tilde{R}_{\text{BK}},\text{avg}}$ — Average Adjusted Total Base Block Keeper Reward Update Time — The average time elapsed between updates of $\tilde{R}_{\text{BK}}$ parameter since the network launch
- $\tilde{R}_{\text{BK},\text{min}}$ — Minimal Adjusted Total Base Block Keeper Reward per Second
- $\tilde{R}_{\text{BK},\text{max}}$ — Maximal Adjusted Total Base Block Keeper Reward per Second
- $M_{\text{MV}}(t)$ — Expected Mobile Verifier Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Mobile Verifiers
- $\tilde{M}_{\text{MV}}$ — Actual Mobile Verifier Total Minted Token Amount — The actual amount of tokens earned by all Mobile Verifiers at time t

- $\Delta_{\tilde{R}_{\mathrm{MV}},\min}$ — Minimal Adjusted Total Mobile Verifier Reward Update Time — The minimum time between updates of the $\tilde{R}_{\mathrm{MV}}$ parameter
- $\Delta_{\tilde{R}_{\mathrm{MV}},\mathrm{avg}}$ — Average Adjusted Total Mobile Verifier Reward Update Time — The average time elapsed between updates of $\tilde{R}_{\mathrm{MV}}$ parameter since the network launch
- $\tilde{R}_{\mathrm{MV},\min}$ — Minimal Adjusted Total Mobile Verifier Reward per Second
- $\tilde{R}_{\mathrm{MV},\max}$ — Maximal Adjusted Total Mobile Verifier Reward per Second
- $M_{\mathrm{BM}}(t)$ — Expected Block Manager Total Minted Token Amount — Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Managers
- $\tilde{M}_{\mathrm{BM}}$ — Actual Block Manager Total Minted Token Amount — The actual amount of tokens earned by all Block Managers at time t
- $\Delta_{\tilde{R}_{\mathrm{BM}},\min}$ — Minimal Adjusted Total Block Manager Reward Update Time — The minimum time between updates of the $\tilde{R}_{\mathrm{BM}}$ parameter
- $\Delta_{\tilde{R}_{\mathrm{BM}},\mathrm{avg}}$ — Average Adjusted Total Block Manager Reward Update Time — The average time elapsed between updates of $\tilde{R}_{\mathrm{BM}}$ parameter since the network launch
- $\tilde{R}_{\mathrm{BM},\min}$ — Minimal Adjusted Total Block Manager Reward per Second
- $\tilde{R}_{\mathrm{BM},\max}$ — Maximal Adjusted Total Block Manager Reward per Second

If no additional adjustments are made to the reward per second, the actual number of minted tokens may significantly deviate from the theoretical one, as Block Keepers have a reputation coefficient parameter. This parameter depends on the individual decisions of each Block Keeper, making the reputation coefficient unpredictable in advance. Furthermore, there are some simplifications in the reward calculations per epoch, which, although causing small deviations from Expected Total Minted Token Amount curve, can accumulate over time. Moreover, it is important to consider not only the Expected Total Minted Token Amount at a specific point in time but also its distribution among the three types of network participants: Block Keeper, Mobile Verifier, and Block Manager. This distribution is predefined and should not change significantly over the course since the network started. For these reasons, we introduce the Automated Reward Adjustment.

This algorithm involves the independent updating of three parameters: $\tilde{R}_{\mathrm{BK}}$, $\tilde{R}_{\mathrm{MV}}$, and $\tilde{R}_{\mathrm{BM}}$, which are included in the reward calculation formulas per epoch (see eq. (45), eq. (35), eq. (43)).

Let's consider the parameter $\tilde{R}_{\mathrm{BK}}$ in detail, as the approach to $\tilde{R}_{\mathrm{MV}}$ and $\tilde{R}_{\mathrm{BM}}$ will follow a similar structure.

Let's assume we are at time t and want to update the reward in such a way as to minimize the error between the theoretical M_{BK} and actual $\tilde{M}_{\mathrm{BK}}$ after some time interval. Let us denote this interval as Δt. Let this interval represent the time until the next reward update. Since $\tilde{R}_{\mathrm{BK}}$ is defined as the reward per second, then to minimize the error the adjusted reward value must compensate for the difference between the theoretical value M_{BK} after the interval Δt and the current actual value $\tilde{M}_{\mathrm{BK}}$.

Obtaining an initial assumption for the $\tilde{R}_{\mathrm{BK}}$ function:

$$\tilde{M}_{\mathrm{BK}}|_t + \tilde{R}_{\mathrm{BK}}(t) \cdot \Delta t = M_{\mathrm{BK}}\left(t + \Delta t\right) \tag{44}$$

However, each Block Keeper has a reputation coefficient, which, under this approach, results in an increased value of $\tilde{M}_{\mathrm{BK}}$ at time $t + \Delta t$. Since the epochs of all Block Keepers almost always last the same amount of time, only two variable factors remain in the formula for calculating the epoch reward (see eq. (45)): the reputation coefficient and the proportion of the stake provided by a specific Block Keeper relative to the total stake. Therefore, at the moment of reward adjustment, we calculate the current stake-weighted average reputation coefficient, which reflects the average reputation coefficient in the network, and use it to correct the adjusted reward.

$$\mathcal{R}_{\mathrm{avg},s}\left(t\right) = \begin{cases} \mathcal{R}_{\min} & \text{if } \tilde{M}_{\mathrm{BK}}|_t = 0 \\ \\ \displaystyle\sum_{i=1}^{\tilde{N}_{\mathrm{BK}}} \mathcal{R}_i|_t \cdot \frac{s_{\mathrm{BK},i}|_t}{S_{\mathrm{BK}}|_t} & \text{if } 0 < \tilde{M}_{\mathrm{BK}}|_t < T \cdot K_{R,\mathrm{BK}} \end{cases}, \tag{45}$$

where $\mathcal{R}_i|_t$ — the reputation coefficient of i-th Block Keeper at time t, $s_{\mathrm{BK},i}|_t$ — the stake provided by i-th Block Keeper at time t.

After introducing the $\mathcal{R}_{\text{avg},s}$, the compensation with the adjusted reward value will be as follows:

$$\tilde{M}_{\text{BK}}|_t + \tilde{R}_{\text{BK}}(t) \cdot \Delta t \cdot \mathcal{R}_{\text{avg},s}(t) = M_{\text{BK}}\left(t + \Delta t\right) \tag{46}$$

From which:

$$\tilde{R}_{\text{BK}}\left(t,\, \Delta t,\, \tilde{M}_{\text{BK}}\right) = \frac{M_{\text{BK}}(t + \Delta t) - \tilde{M}_{\text{BK}}|_t}{\Delta t \cdot \mathcal{R}_{\text{avg},s}(t)} \tag{47}$$

Regarding Δt, since it is impossible to guarantee that a responsible entity in a decentralized system will update the $\tilde{R}_{\text{BK}}$ parameter at a consistent, fixed time interval Δt, we introduce the parameter $\Delta_{\tilde{R}_{\text{BK}},\text{min}}$, which represents the minimum time between updates of the $\tilde{R}_{\text{BK}}$ parameter, and the parameter $\Delta_{\tilde{R}_{\text{BK}},\text{avg}}$, which represents the actual average time elapsed between updates of $\tilde{R}_{\text{BK}}$ since the network launch. Thus, the responsible entity will update the $\tilde{R}_{\text{BK}}$ parameter no more frequently than once every $\Delta_{\tilde{R}_{\text{BK}},\text{min}}$ seconds.

Therefore, Δt can be approximated as $\Delta_{\tilde{R}_{\text{BK}},\text{avg}}|_t$ at time t. At the moment $t = 0$, the parameter $\Delta_{\tilde{R}_{\text{BK}},\text{avg}}$ is set equal to the parameter $\Delta_{\tilde{R}_{\text{BK}},\text{min}}$.

In addition, the $\tilde{R}_{\text{BK}}$ parameter cannot be lower than $\tilde{R}_{\text{BK},\text{min}}$ or higher than $\tilde{R}_{\text{BK},\text{max}}$ to prevent sharp fluctuations in the reward per second during the update of the $\tilde{R}_{\text{BK}}$ parameter. Moreover, $\tilde{R}_{\text{BK},\text{max}}\left(t\right) = \tilde{R}_{\text{BK}}\left(t - \Delta_{\tilde{R}_{\text{BK}},\text{avg}}\right)$. That is, the new updated value of $\tilde{R}_{\text{BK}}$ cannot be greater than the previous one. This way, we protect against potential manipulations with the epoch start time by Block Keepers, aimed at timing the moment when the reward would be excessively high.

Obtaining the final formula for the adjusted reward $\tilde{R}_{\text{BK}}$:

$$\tilde{R}_{\text{BK}}\left(t,\, \Delta_{\tilde{R}_{\text{BK}},\text{avg}},\, \tilde{M}_{\text{BK}},\, \tilde{R}_{\text{BK},\text{min}},\, \tilde{R}_{\text{BK},\text{max}}\right) = \min\left(\max\left(\frac{M_{\text{BK}}(t + \Delta_{\tilde{R}_{\text{BK}},\text{avg}}|_t) - \tilde{M}_{\text{BK}}|_t}{\Delta_{\tilde{R}_{\text{BK}},\text{avg}}|_t \cdot \mathcal{R}_{\text{avg},s}(t)},\, \tilde{R}_{\text{BK},\text{min}}\right),\, \tilde{R}_{\text{BK},\text{max}}\right) \tag{48}$$

For Mobile Verifiers and Block Managers, the reward-per-second formulas lack a factor, similar to the reputation coefficient, that adjusts a participant's reward strictly individually (see eq. (26), eq. (36)).

Instead, coefficients $(K_{\mathcal{B}} \cdot \mathcal{B} + K_G \cdot G_{\text{norm}})$ and $\mathcal{E}$ are used, each of which, summed across all network participants, is equal to 1 at any given time. This ensures that the total number of tokens distributed to all Mobile Verifiers and Block Managers per second of participation does not depend on their individual decisions.

Therefore, the following simplified formulas can be used to adjust the rewards for Mobile Verifiers and Block Managers:

$$\tilde{R}_{\text{MV}}\left(t,\, \Delta_{\tilde{R}_{\text{MV}},\text{avg}},\, \tilde{M}_{\text{MV}},\, \tilde{R}_{\text{MV},\text{min}},\, \tilde{R}_{\text{MV},\text{max}}\right) = \min\left(\max\left(\frac{M_{\text{MV}}(t + \Delta_{\tilde{R}_{\text{MV}},\text{avg}}|_t) - \tilde{M}_{\text{MV}}|_t}{\Delta_{\tilde{R}_{\text{MV}},\text{avg}}|_t},\, \tilde{R}_{\text{MV},\text{min}}\right),\, \tilde{R}_{\text{MV},\text{max}}\right) \tag{49}$$

$$\tilde{R}_{\text{BM}}\left(t,\, \Delta_{\tilde{R}_{\text{BM}},\text{avg}},\, \tilde{M}_{\text{BM}},\, \tilde{R}_{\text{BM},\text{min}},\, \tilde{R}_{\text{BM},\text{max}}\right) = \min\left(\max\left(\frac{M_{\text{BM}}(t + \Delta_{\tilde{R}_{\text{BM}},\text{avg}}|_t) - \tilde{M}_{\text{BM}}|_t}{\Delta_{\tilde{R}_{\text{BM}},\text{avg}}|_t},\, \tilde{R}_{\text{BM},\text{min}}\right),\, \tilde{R}_{\text{BM},\text{max}}\right) \tag{50}$$

14 SHELL — Equal or Less

SHELL is a network usage token, designed to provide compensation for Block Keepers for their computing resources. Anyone who wishes to execute a transaction on Acki Nacki needs to pay Block Keepers for their computing resources and storage. Since main expenses for running a Block Keeper are electricity and network traffic costs and server amortization (wherever hardware or lease costs), and all of them are paid in fiat currency the SHELL price should try to reflect those.

SHELL Tokens will be sold via a System Pool in exchange for any currency Block Keepers decided to accept. Block Keepers will provide liquidity for such exchange and set up a SHELL minting rate for that pair, which will constitute their collective vote on current conversation price for a particular pair. Any SHELL holder may decide to sell their unused SHELL tokens which will be placed in the pool setting the price lower, respectively until the supply is not sold. Therefore the SHELL Token can be sold at the price Block Keepers set up in the Pool, or less. Hence — equal or less.

All the payments collected for SHELL tokens are then directed into an Accumulator Contract where they are locked. Any NACKL token holder has a proportional right to the content of the Accumulator Contract. At any time NACKL holder can decide to burn their tokens and receive the proportional amount locked in Accumulator Contract.

A NACKL holder would rarely (or never) use such a mechanism because most of the time the open market price of NACKL will be higher than revenues divided by tokens outstanding because of future revenues expectations and decreasing supply mechanism built into the market price of NACKL.

Since all SHELL revenues go to the Accumulator Contract, the amount of Revenue divided by the amount of NACKL Tokens will constitute the "intrinsic" or a "floor" value of the NACKL. This intrinsic value will always rise while the NACKL supply will always decrease.

A Glossary

Here is a table with all the variables used in the article. The table contains the columns 'Name', 'Notation', 'Definition', and 'Value'. If the value of a variable is constant and does not change after the network starts, its value is provided in the 'Value' column; otherwise, it is marked as '-'.

Name	Notation	Description	Value
Time	t	Time since the network launch in seconds	-
Bitcoin Total Supply	T_B	The total number of Bitcoins to be mined	$21,000,000$
Bitcoin Block Reward	$R_B(t)$	The amount of Bitcoin mined per block in the Bitcoin network. The Initial Bitcoin Block Reward denotes as $R_{B,0}$.	-
Bitcoin Seconds per Block	$\Delta_{B,B}$	The average block time in Bitcoin in seconds	600
Bitcoin Delta Halving	$\Delta_{B,H}$	The number of seconds that pass on average between halvings in the Bitcoin network, taking into account that, on average, a block is mined every $\Delta_{B,B}$ seconds	$126,000,000$
Bitcoin Total Mined Token Amount	$M_B(t)$	The number of mined Bitcoins at time t	-
Approximated Bitcoin Total Mined Token Amount	$\tilde{M}_B(t)$	The approximation of $M_B(t)$ function by an exponential saturation function	-
Total Supply	T	The total number of tokens to be minted	$10,400,000,000$
Expected Total Minted Token Amount	$M(t)$	The expected number of minted tokens at time t	-
Expected General Reward per Second	$R(t)$	Expected reward for network participation per second for all network participants at time t	-
Total Token Minting Time	τ	The expected time for minting the last fraction of token in seconds	$2,000,000,000$
Total Minted Token Amount Function Coefficient	K_M	The parameter regulating the decay rate of the Total Minted Token Amount function	0.00001
Expected Total Base Block Keeper Reward per Second	$R_{\text{BK}}(t)$	Fraction of the reward $R(t)$ allocated to Block Keepers	-
Expected Total Mobile Verifier Reward per Second	$R_{\text{MV}}(t)$	Fraction of the reward $R(t)$ allocated to Mobile Verifiers	-
Expected Total Block Manager Reward per Second	$R_{\text{BM}}(t)$	Fraction of the reward $R(t)$ allocated to Block Managers	-

Name	Notation	Description	Value
Base Block Keeper Reward Function Coefficient	$K_{R,\mathrm{BK}}$	The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Keepers	0.675
Mobile Verifier Reward Function Coefficient	$K_{R,\mathrm{MV}}$	The coefficient that determines the fraction of the reward $R(t)$ allocated to Mobile Verifiers	0.225
Block Manager Reward Function Coefficient	$K_{R,\mathrm{BM}}$	The coefficient that determines the fraction of the reward $R(t)$ allocated to Block Managers	0.1
Expected Block Keeper Total Minted Token Amount	$M_{\mathrm{BK}}(t)$	Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Keepers	-
Expected Mobile Verifier Total Minted Token Amount	$M_{\mathrm{MV}}(t)$	Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Mobile Verifiers	-
Expected Block Manager Total Minted Token Amount	$M_{\mathrm{BM}}(t)$	Fraction of the Expected Total Minted Token Amount $M(t)$ allocated to Block Managers	-
Adjusted Total Base Block Keeper Reward per Second	$\tilde{R}_{\mathrm{BK}}(t)$	Adjusted reward for network participation per second for all Block Keepers	-
Adjusted Total Mobile Verifier Reward per Second	$\tilde{R}_{\mathrm{MV}}(t)$	Adjusted reward for network participation per second for all Mobile Verifiers	-
Adjusted Total Block Manager Reward per Second	$\tilde{R}_{\mathrm{BM}}(t)$	Adjusted reward for network participation per second for all Block Managers	-
Reputation Coefficient	$\mathcal{R}$	Additional rewards granted to a Block Keeper for continuous validation	-
Block Keeper Reputation Time	$t_{\mathcal{R}}$	The time during which the Block Keeper has been continuously running validation epochs	-
Minimal Reputation Coefficient	$\mathcal{R}_{\min}$	Minimal Reputation Coefficient	1
Maximal Reputation Coefficient	$\mathcal{R}_{\max}$	Maximal Reputation Coefficient	3
Maximal Reputation Time	$t_{\mathcal{R},\max}$	The time it takes for the Block Keeper to accumulate maximum reputation for continuous validation in seconds	$157,766,400$
Adjustment Reputation Function Coefficient	$K_{\mathcal{R}}$	The parameter regulating the rate of reputation growth over time	$1,000$
Block Keeper Reward per Second	$r_{\mathrm{BK}}(t)$	The reward earned by a Block Keeper per second of validation, depending on their stake and current Reputation Coefficient	-

Name	**Notation**	**Description**	**Value**
Block Keeper Stake	s_{BK}	The specific amount of tokens that a Block Keeper has staked the latest time in order to participate in validation	-
Total Block Keeper Stake	S_{BK}	The sum of all Block Keeper stakes s_{BK} at time	-
Validation Epoch Start Time	t_{val}	The time in seconds that has passed from the moment the network was launched until the start of a particular validation epoch	-
Block Keeper Reward per Validation Epoch	$r_{\text{BK},e}(t)$	The reward received by a Block Keeper per one validation epoch	-
Block Keeper Epoch Duration	$\Delta_{\text{BK},e}$	The duration of one validation epoch in seconds	-
Actual Block Keeper Number	$\tilde{N}_{\text{BK}}$	The number of Block Keepers in the last produced block	-
Actual Block Keeper Total Minted Token Amount	$\tilde{M}_{\text{BK}}$	The actual amount of tokens earned by all Block Keepers at time t	-
Maximal Free Float Fraction	$F_{\max}$	Maximal fraction of Free Float of Total Minted Token Amount	1/3
Free Float Fraction	$F(t)$	The current fraction of Free Float of Total Minted Token Amount	-
Free Float Function Coefficient	K_F	The parameter regulating the decay rate of the F function	0.01
Expected Total Staked Token Amount	$S(t)$	The expected total number of tokens staked in the network at time t	-
Actual Block Keeper Total Available Token Amount	$\tilde{M}_{\text{BK,av}}$	The total amount of tokens available for staking. Calculated as the difference between the current total minted token amount and the total number of tokens burned during the whole slashing.	-
Staking Time	t_{stk}	The time in seconds that has passed from the moment the network was launched until the registration for the next epoch and setting the current stake	-
Base Minimal Block Keeper Stake	$s_{\text{BK,base}}(t)$	Minimal Stake when the current number of Block Keepers equals the necessary number of Block Keepers	-
Minimal Block Keeper Stake	$s_{\text{BK,min}}(t)$	Current minimal Block Keeper stake depending on the particular difference between the current number of Block Keepers and the required number of Block Keepers	-
Needed Block Keeper Number	$N_{\text{BK,req}}$	The number of Block Keepers required in the network at time t according to the number of threads	-

Name	Notation	Description	Value
Adjustment Minimal Block Keeper Stake Function Coefficient	K_s	The parameter regulating the growth of the $s_{\text{BK,min}}(t)$ function	0.001
Year Number	y	The year number since the network launch at which we calculate the $R_{y,\%}$	-
Annual Percentage Reward	$R_{y,\%}\left(y\right)$	A measure of the yearly earnings expressed as a percentage of the average amount of tokens staked during the year	-
Annual Minted Token Amount	$M_y\left(y\right)$	An expected number of tokens minted over the year	-
Annual Average Staked Token Amount	$S_y\left(y\right)$	The average amount of tokens staked during the year	-
Seconds In Year	σ_y	Average number of seconds in a year	$31,557,600$
Acki-Nacki Number	v	The average number of Acki-Nacki per block	-
Attestation Number	A	The number of attestations required for block finalization	-
Malicious Block Keeper Number	$N_{\text{BK,mal}}$	The expected number of Malicious Block Keepers	-
Successful Attack Probability	p	Successful attack probability in a single attempt	-
Mobile Verifier Number	N_{MV}	The number of Mobile Verifiers, who meet the reward eligibility condition	-
Malicious Mobile Verifier Number	$N_{\text{MV,mal}}$	Expected number of Malicious Mobile Verifiers	-
Mobile Verifier Verification Frequency	λ_{MV}	Fraction of blocks verified by Mobile Verifiers	-
Mobile Verifier Successful Attack Probability	p_{MV}	Successful attack probability in a single attempt with Mobile Verifiers	-
Mobile Verifier Reward per Second	$r_{\text{MV}}(t)$	The reward earned by a Mobile Verifier per second of verification	-
Boost Coefficient	$\mathcal{B}$	A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier based on their position in the sorted in ascending order list of all Mobile Verifiers, who meet the reward eligibility condition, by the number of Boosts	-
Epoch Game Score	G	The score of a Mobile Verifier in the online game for one epoch	-

Name	**Notation**	**Description**	**Value**
Cluster Boost Coefficient	$\mathcal{B}_i^{\text{cl}}$	A coefficient that determines the fraction of the reward allocated to a particular Mobile Verifier cluster i based on their position in the sorted in ascending order list of all Mobile Verifiers clusters	-
Epoch Cluster Game Score	G_i^{cl}	The score of all Mobile Verifiers belonging the cluster i in the online game for one epoch	-
Cluster Number	N_{cl}	The current number of clusters consisting of Mobile Verifiers	-
Cluster Reward per Second	$R_{\text{MV},i}^{\text{cl}}(t)$	The reward allocated to the cluster i at time t	-
Mobile Verifier Cluster Number	$\Delta_{\text{MV}}^{\text{cl},i}$	The number of Mobile Verifiers belonging cluster i	-
Verification Epoch Start Time	t_{verf}	The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Mobile Verifiers	-
Minimal Mobile Verifier Stake	$s_{\text{MV,min}}(t)$	Current minimal particular Mobile Verifier stake	-
Mobile Verifier Reward History	$r_{\text{MV},\Sigma}$	Amount of tokens that Mobile Verifier have earned during their entire participation in the network	-
Mobile Verifier Reward per Epoch	$r_{\text{MV},e}(t)$	The reward received by a Mobile Verifier per one verification epoch	-
Mobile Verifier Epoch Duration	$\Delta_{\text{MV},e}$	The duration of one epoch of Mobile Verifiers in seconds	-
Actual Mobile Verifier Total Minted Token Amount	$\tilde{M}_{\text{MV}}$	The actual amount of tokens earned by all Mobile Verifiers at time t	-
Block Manager Reward per Second	$r_{\text{BM}}(t)$	The reward earned by a Block Manager per second of management	-
External Messages Coefficient	$\mathcal{E}$	The coefficient that determines the fraction of the reward allocated to a particular Block Manager based on their position in the sorted in ascending order list of all Block Managers by the number of processed external messages	-
Block Manager Number	N_{BM}	The current number of Block Managers in the network	-
Management Epoch Start Time	t_{mng}	The time in seconds that has passed from the moment the network was launched until the start of a particular epoch of Block Managers	-
Block Manager Reward History	$r_{\text{BM},\Sigma}$	Amount of tokens that Block Manager have earned during their entire participation in the network	-
Minimal Block Manager Stake	$s_{\text{BM,min}}(t)$	Current minimal particular Block Manager stake	-

Name	Notation	Description	Value
Block Manager Reward per Epoch	$r_{\text{BM},e}$	The reward received by a Block Manager per one management epoch	-
Block Manager Epoch Duration	$\Delta_{\text{BM},e}(t)$	The duration of one epoch of Block Manager in seconds	-
Actual Block Manager Total Minted Token Amount	$\tilde{M}_{\text{BM}}$	The actual amount of tokens earned by all Block Managers at time t	-
Stake Weighted Average Reputation Coefficient	$\mathcal{R}_{\text{avg},s}(t)$	The stake-weighted average value of the reputation coefficient across all Block Keepers in the network at time t	-
Minimal Adjusted Total Base Block Keeper Reward Update Time	$\Delta_{\tilde{R}_{\text{BK}},\text{min}}$	The minimum time between updates of the $\tilde{R}_{\text{BK}}$ parameter	-
Average Adjusted Total Base Block Keeper Reward Update Time	$\Delta_{\tilde{R}_{\text{BK}},\text{avg}}$	The average time elapsed between updates of $\tilde{R}_{\text{BK}}$ parameter since the network launch	-
Minimal Adjusted Total Base Block Keeper Reward per Second	$\tilde{R}_{\text{BK,min}}$	Minimal Adjusted Total Base Block Keeper Reward per Second	-
Maximal Adjusted Total Base Block Keeper Reward per Second	$\tilde{R}_{\text{BK,max}}$	Maximal Adjusted Total Base Block Keeper Reward per Second	-
Minimal Adjusted Total Mobile Verifier Reward Update Time	$\Delta_{\tilde{R}_{\text{MV}},\text{min}}$	The minimum time between updates of the $\tilde{R}_{\text{MV}}$ parameter	-
Average Adjusted Total Mobile Verifier Reward Update Time	$\Delta_{\tilde{R}_{\text{MV}},\text{avg}}$	The average time elapsed between updates of $\tilde{R}_{\text{MV}}$ parameter since the network launch	-
Minimal Adjusted Total Mobile Verifier Reward per Second	$\tilde{R}_{\text{MV,min}}$	Minimal Adjusted Total Mobile Verifier Reward per Second	-
Maximal Adjusted Total Mobile Verifier Reward per Second	$\tilde{R}_{\text{MV,max}}$	Maximal Adjusted Total Mobile Verifier Reward per Second	-
Minimal Adjusted Total Block Manager Reward Update Time	$\Delta_{\tilde{R}_{\text{BM}},\text{min}}$	The minimum time between updates of the $\tilde{R}_{\text{BM}}$ parameter	-
Average Adjusted Total Block Manager Reward Update Time	$\Delta_{\tilde{R}_{\text{BM}},\text{avg}}$	The average time elapsed between updates of $\tilde{R}_{\text{BM}}$ parameter since the network launch	-

Name	Notation	Description	Value
Minimal Adjusted Total Block Manager Reward per Second	$\tilde{R}_{\mathrm{BM,min}}$	Minimal Adjusted Total Block Manager Reward per Second	-
Maximal Adjusted Total Block Manager Reward per Second	$\tilde{R}_{\mathrm{BM,max}}$	Maximal Adjusted Total Block Manager Reward per Second	-

References

[1] Bitcoin historical data. https://coinmarketcap.com/currencies/bitcoin/ (2023), coinMarketCap

[2] Avalanche, T.: Avalanche consensus: Performance through network synchronization. Ava Labs White Paper (2020), https://www.avalabs.org/whitepapers, retrieved from https://www.avalabs.org/whitepapers

[3] Buterin, V.: A next-generation smart contract and decentralized application platform. Ethereum White Paper (2014), https://ethereum.org/en/whitepaper/, retrieved from https://ethereum.org/en/whitepaper/

[4] Böhme, R., Christin, N., Edelman, B., Moore, T.: Bitcoin: Economics, technology, and governance. Journal of Economic Perspectives **29**(2), 213–238 (2015)

[5] Garcia, D., Tessone, C.J., Mavrodiev, P., Perony, N.: The digital traces of bubbles: Feedback cycles between socio-economic signals in the bitcoin economy. Journal of the Royal Society Interface **11**(99), 20140623 (2014). https://doi.org/10.1098/rsif.2014.0623, https://doi.org/10.1098/rsif.2014.0623

[6] Goroshevsky, M., Sattarov, N., Trepacheva, A.: Acki Nacki: A Probabilistic Proof-of-Stake Consensus Protocol with Fast Finality and Parallelisation. In: International Conference on Applied Cryptography and Network Security. pp. 43–62. Springer (2024), https://link.springer.com/chapter/10.1007/978-3-031-61486-6_4

[7] Hayek, F.A.: The use of knowledge in society. The American Economic Review **35**(4), 519–530 (1945)

[8] Jain, A., et al.: We might walk together, but I run faster: Network Fairness and Scalability in Blockchains. arXiv preprint arXiv:2102.04326 (2021), https://arxiv.org/pdf/2102.04326

[9] Keynes, J.M.: The General Theory of Employment, Interest and Money. Macmillan (1936)

[10] King, S., Nadal, S.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake (Aug 19 2012), https://decred.org/research/king2012.pdf

[11] Mankiw, N.G.: Principles of Economics. Cengage Learning, 9th edn. (2020)

[12] Metcalfe, B.: Metcalfe's law after 40 years of ethernet. Computer **46**(12), 26–31 (2013). https://doi.org/10.1109/MC.2013.374, https://doi.org/10.1109/MC.2013.374

[13] Mises, L.v.: Human Action: A Treatise on Economics. Yale University Press (1949)

[14] Nakamoto, S.: Re: Bitcoin mining is now worth less than cost of electricity. https://bitcointalk.org/index.php?topic=721.msg8114#msg8114 (February 18 2010), [Online forum post]

[15] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf (2008)

[16] Pass, R., Shi, E.: Fruitchains: A fair blockchain. In: Proceedings of the ACM Symposium on Principles of Distributed Computing. pp. 315–324. ACM (2017), https://dl.acm.org/doi/10.1145/3087801.3087809

[17] Ricardo, D.: On the Principles of Political Economy and Taxation. John Murray (1817)

[18] Saleh, F.: Blockchain without waste: Proof-of-stake. The Review of Financial Studies **34**(3), 1156–1190 (2021). https://doi.org/10.1093/rfs/hhaa075, https://doi.org/10.1093/rfs/hhaa075

[19] Samuelson, P.A., Nordhaus, W.D.: Economics. McGraw-Hill, 19th edn. (2010)

[20] Smith, A.: An Inquiry into the Nature and Causes of the Wealth of Nations. W. Strahan and T. Cadell (1776)

[21] Yakovenko, A.: Solana: A new architecture for a high performance blockchain. Solana White Paper (2018), https://solana.com/solana-whitepaper.pdf, retrieved from https://solana.com/solana-whitepaper.pdf