HoneSTake: A Dynamic, Centralization Resistant, Reputation Driven Staking Protocol for Enhanced Security and Governance in Permissionless Peer-to-Peer Blockchain Networks

By: Andrew Nicholas Smith
Title: Andrew Nicholas Smith, Founder
Organization: Versatus Labs
Email: Andrew Nicholas Smith <as@versatus.io>
Organization Email: info@versatus.io
Website: versatus.io

Abstract

The HoneSTake protocol emerges as a pioneering solution to the challenges of security and centralization in permissionless, peer-to-peer blockchain networks. Distancing itself from traditional stake-weighted or hash power-driven models, which often gravitate towards centralization, HoneSTake champions a "1 node, 1 vote" ethos. This is achieved by ingeniously intertwining node reputation scores with dynamic staking requirements. At its heart, the protocol employs a trustless decentralized node reputation management system, harmonizing both local and global perspectives of node reputations. Through mechanisms like threshold approximate agreement and efficient global state updates, the system ensures that nodes with superior reputations benefit from reduced staking demands. This structure not only elevates the cost and complexity of malicious attempts to control the network but also inherently fosters decentralization. When paired with punitive measures like jailing and slashing, either automated or socially consensus-driven, HoneSTake creates a robust economic alignment between node operators and the overarching network protocol. The culmination of these features offers a promising pathway to hyper-decentralized networks, enhancing resistance to adversarial actors and bolstering overall network security.

Introduction

In the rapidly advancing realm of permissionless peer-to-peer networks, especially blockchains, the pursuit of robust security presents multifaceted challenges[1]. At the heart of these challenges lies the imperative to structure economic incentives such that potential adversarial actions are not merely costly but, in an ideal scenario, infeasible. Over time, the blockchain community has observed a transition from the resource-intensive Proof-of-Work (PoW) paradigm to the more capital-focused Proof-of-Stake (PoS) framework[2].

Nevertheless, both these paradigms harbor intrinsic vulnerabilities, potentially leading to centralization[3]. Such centralization risks compromise the foundational decentralized ethos of blockchain. In this paper, we introduce a dynamic staking model, underpinned by node reputation inspired by the HonestPeer algorithm[4]. This model, integrated with the egalitarian principle of "1 Node 1 Vote," presents a promising direction for enhancing security and governance in permissionless blockchain networks.

The Centralization Conundrum in Conventional Paradigms

Proof-of-Stake (PoS) Centralization Perils:

- **Whale Dominance:** A limited number of affluent entities, by virtue of holding a substantial stake, can exert disproportionate influence, leading to power centralization.
- **Network Usurpation:** A coalition of such affluent entities, if they strategize collaboratively, might achieve control over more than half of the total stake, thereby jeopardizing the network's decentralized spirit.
- **Stagnation:** The escalating rewards for dominant stakers can deter newcomers, resulting in a static or diminishing validator pool.
- **Economic Drives:** The predominant incentive being economic can inadvertently promote centralization, as resource-rich entities gain ascendancy.

Proof-of-Work (PoW) Centralization Perils:

- **Hardware Hegemony:** Entities equipped with cutting-edge mining hardware can potentially centralize both decision-making and reward distribution.
- **Economies of Scale:** Expansive mining operations, due to their scale, can optimize electricity expenses, further concentrating mining prowess.
- **Mining Consortiums:** Such collectives consolidate hashing power. Although initially dependent on power delegation, the dynamics can pivot if they secure direct control over the mining apparatus, either via capital investments or even acquisition.

- **Ecological Implications:** The energy-intensive character of PoW raises environmental apprehensions, potentially inviting regulatory interventions favoring miners with sustainable energy access.

Introducing HoneSTake: A Decentralization-Affirmative, Reputation-Centric Staking Protocol

HoneSTake pioneers a fresh perspective on staking, emphasizing sustained, genuine participation over mere capital prowess. Central to HoneSTake is the dynamic staking criterion, anchored to a node's reputation, ascertained by an algorithm reminiscent of HonestPeer, albeit with certain refinements. HoneSTake significantly mitigates the centralization pitfalls inherent to traditional PoS models:

- Equitable Power Dispersion: HoneSTake's "1 node, 1 vote" tenet ensures a balanced power distribution, championing genuine decentralization.
- Reputation Supersedes Capital: Linking staking prerequisites to node reputation, HoneSTake celebrates enduring, genuine engagement over sheer capital aggregation.
- Augmented Security: The escalating expense associated with network power acquisition deters malevolent intrusions, rendering them cost-prohibitive.

In the ensuing sections, we will discuss and elaborate on the theoretical foundations of HoneSTake, underscoring its potential to redefine blockchain security and governance by thwarting centralization. We commence by detailing a probabilistic, scalable, and trustless methodology for node reputation monitoring, encompassing the algorithmic and structural mechanisms for reputation management. Subsequently, we explore the collaborative dynamics for stake requirement computation.

A Scalable, Trustless Peer Reputation Algorithm

Central to the HoneSTake protocol is its innovative reputation system[5]. This system is indispensable for tracking, updating, and achieving a consensus on a node's reputation. Without it, the dynamic staking calculations, which determine a node's eligibility as a validator and governing entity, would be unattainable.

Interestingly, the design does not necessitate a uniform, synchronized reputation across the entire network. On the contrary, a diversified view is more desirable. A uniform reputation perspective would mandate the integration of the reputation algorithm with the network's consensus mechanism. Such an integration could introduce substantial overhead, potentially hindering the network's primary function: processing user transactions. To circumvent this, we employ a consensus approximation to ascertain node reputation.

Before delving into the intricacies of node stake requirements, it's pivotal to understand the algorithm employed by the node client to refresh peer reputation scores and the data structure preserving these scores.

Within the HoneSTake framework, every node possesses an individual perspective on its peer's reputation[6]. The collective reputation of a node in the network is a synthesis of its interactions with other nodes and the feedback from those nodes[7]. This system's architecture hinges on two assessments of a node's reputation:

Local Reputation:

- This score mirrors the satisfaction derived from the interactions between the receiving and sending peers. Factors like responsiveness and adherence to the network protocol play a role. Subsequently, this local perspective influences the weighting of third-party interactions.

Global Reputation:

- Representing an aggregate, this score is a summation of the interactions between other nodes, weighted by local reputation. It excludes interactions involving the local node. Crucially, this metric dictates the stake requirements, influencing a node's eligibility as a validator and governing entity.

Pseudocode Local Reputation Update Algorithm

```
def update_local_reputation(local_reputation, peer, satisfaction, value):
  if satisfaction == "positive":
    local_reputation[peer] += value
  else if satisfaction == "negative":
    local_reputation[peer] -= value
```

Pseudocode Global Reputation Update Algorithm

```
def update_global_reputation(
  local_reputation, global_reputation, reporting_peer, interaction_peer, satisfaction, value
):
  weight = local_reputation[reporting_peer]
```

```
if satisfaction == "positive":
    global_reputation[interaction_peer] += (weight * value)
else if satisfaction == "negative":
    global_reputation[interaction_peer] -= (weight * value)
```

In the algorithms provided, the interaction's value is contingent on the specific network protocol where the staking protocol is deployed. Different networks might prioritize certain interactions over others, assigning varying values to both positive and negative interactions. The network protocol's objectives, the significance of specific interactions, and other factors should be considered when determining the value of interactions influencing nodes' reputation.

Peer Rotation and Hysteresis in Peer-to-Peer Networks

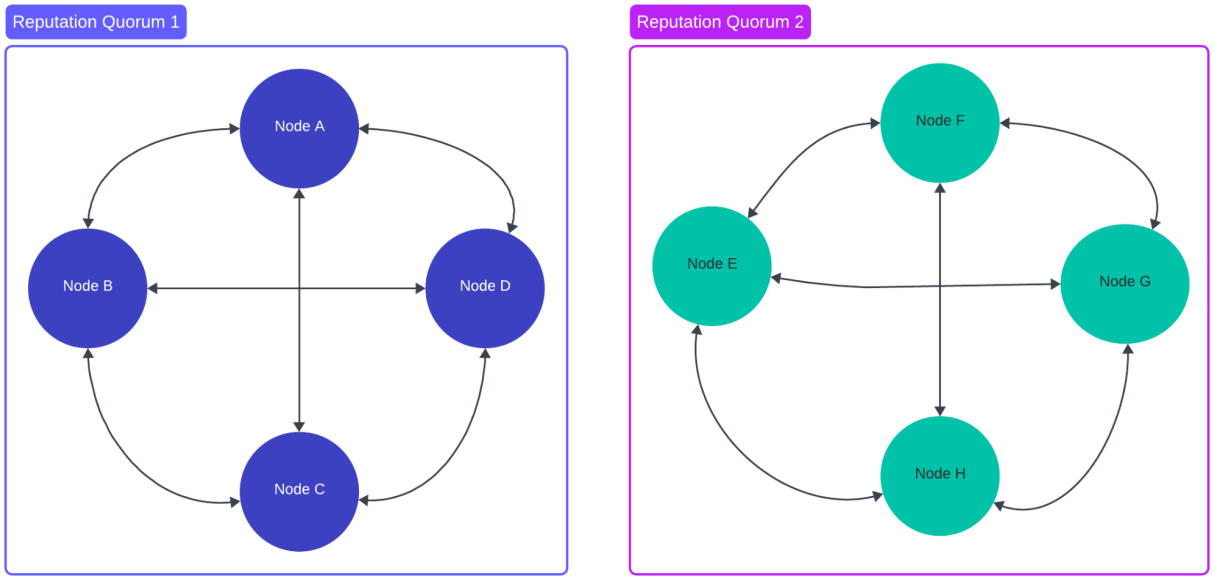
In the realm of peer-to-peer networks, the dynamic management of peers is crucial. This involves pruning redundant connections and reorganizing peers into efficient subsets, ensuring rapid information dissemination with minimal hops[8]. While a comprehensive discussion of peer-to-peer network structures is beyond this paper's purview, the concept of peer rotation holds significance in the context of the HoneSTake protocol.

For the HoneSTake system, it's imperative to frequently and randomly rotate peers that evaluate each other's reputation[9]. This rotation strategy aims to mitigate the risks associated with static peer groups, which could potentially harbor colluding nodes. By constantly shuffling peer associations, the system disrupts any emerging trust or understanding among potentially malicious nodes. The absence of consistent peer interactions makes malicious endeavors riskier. A node, after engaging in dubious activities, might find itself amidst a group of honest peers, who could then report its misbehavior.

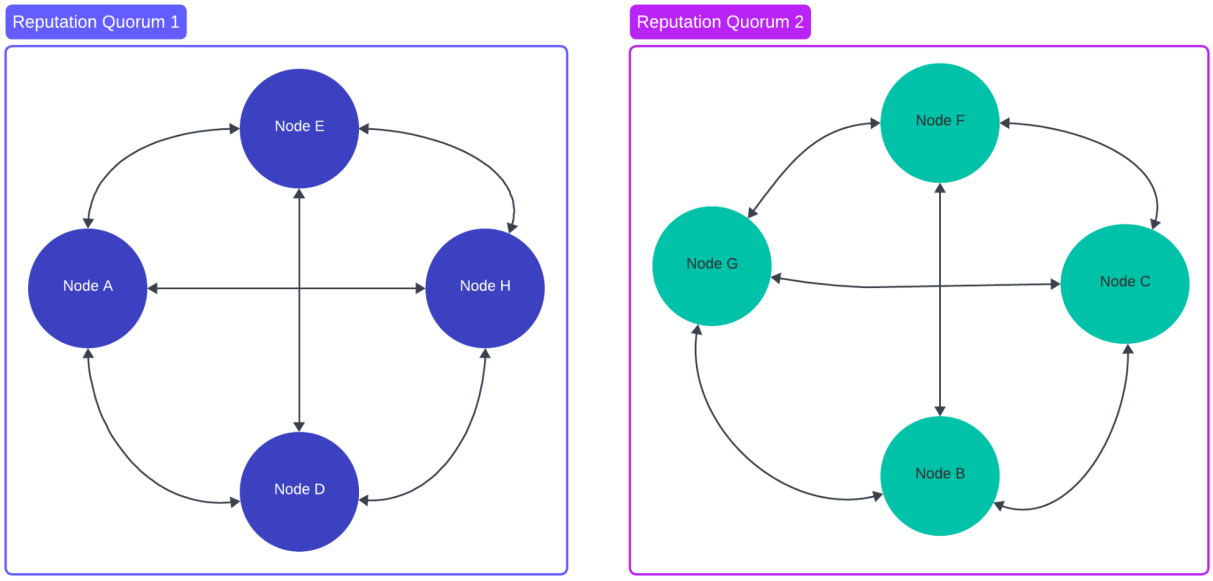
Introducing peer rotation adds layers of complexity to node coordination. Nodes must perpetually recalibrate their strategies, adapting to an ever-evolving peer landscape. This dynamic environment discourages malicious nodes from investing in intricate attack strategies, especially against the reputation system, as the associated costs and risks escalate.

To bolster the peer rotation mechanism's efficacy, the protocol incorporates a Hysteresis or "cool-down" phase. If a cluster of colluding nodes does form, their subsequent rotation ensures they can't easily regroup with their former colluding partners. This design choice significantly hampers sustained, harmful collusion, rendering it nearly unfeasible[10][11].

Starting Reputation Cluster



Peer Rotation



Nodes A & D, C & B cannot all 4 be in the same cluster again for a period of time

Efficient Updates and Synchronization of Reputation Scores

The HoneSTake protocol's reputation update mechanism, if executed after every interaction, could introduce significant network bottlenecks and strain bandwidth[12]. To address this, the protocol adopts a batched reporting approach for interactions, updating the network's node reputations efficiently. Instead of immediately reporting post-interaction, nodes batch their peer interactions and request global view updates at randomized intervals. This randomness complicates any attempts by sybil or colluding nodes to exploit the update intervals. Given the diverse reputation perspectives across nodes, it's crucial that any consensus leading to synchronized reputation scores aligns with the system's objectives.

Cryptographically Secure Batched Updates

Rather than continuously reporting each peer interaction, nodes, at randomized intervals, send batched updates of interactions since their last report[13]. This batching minimizes network congestion and ensures the primary network functions, like transaction processing, remain unhindered. Nodes maintain a local reputation view, caching interactions by adjusting their local perspective of a node's reputation. During reporting, nodes transmit the difference between the current and previously reported local views.

Each batch update is authenticated using the reporting node’s ECDSA signature, ensuring data integrity. This cryptographic approach negates any potential for mid-report data manipulation. If any discrepancy arises between the signed and received data, the recipient node can reject the update and alert the network of potential interference.

Verifiably Random Reporting Intervals

Randomized reporting intervals offer several advantages. While they help distribute network load and reduce bandwidth and computational overhead, their unpredictability is their primary asset[14]. Predictable intervals could expose the system to timing attacks or collusion. Random intervals thwart such attempts, making coordinated attacks or manipulations challenging. If colluding nodes cannot predict when their counterparts will report, inconsistencies arise, potentially flagging malicious activity.

Reputation Decay: Ensuring Continuous Positive Behavior

HoneSTake introduces a systematic, epoch-based reputation score decay. This decay mechanism encourages nodes to consistently exhibit positive behavior[15]. Nodes can't merely capitalize on past positive actions; they must remain active and adhere to the protocol. This approach prevents nodes from becoming dormant and later benefiting from their historical reputation. Such a decay

system levels the playing field for newcomers, ensuring they can establish themselves without being overshadowed by dormant nodes with historical reputation.

Furthermore, reputation decay acts as a corrective mechanism. If a node is penalized for misconduct and becomes inactive, its reputation score diminishes, leading to escalating stake requirements. This decay provides an economic impetus for node operators to promptly address issues, ensuring their node operates honestly and efficiently.

Conclusion

HoneSTake's reputation management system, encompassing cryptographically verifiable batch updates, randomized reporting intervals, and systematic reputation decay, is designed for efficiency and security. While batch updates and randomized intervals optimize network performance and deter malicious activities, reputation decay ensures continuous positive participation, fostering a balanced environment for both established and new nodes.

Staggered Threshold Approximate Agreement Protocol for Reputation State Updates

In decentralized systems like HoneSTake, node reputation scores can vary among peers. Achieving consensus on a singular "correct" reputation score for any node is challenging. To address this, we introduce the Staggered Threshold Approximate Agreement protocol, designed to efficiently update node reputation scores[16].

Staggered Updates: Balancing Frequency and Efficiency

Rather than updating a node's reputation score in the global state after every interaction, which could strain the network, we advocate for staggered updates. By updating different node reputation scores at varying times, the protocol ensures that updates are neither too frequent nor too sparse. This approach allows the reputation score, and consequently the stake requirement, to be part of the global consensus without compromising network speed or the protocol's security features. Randomizing the timing of these updates further safeguards against potential collusion or timing attacks[17].

Threshold Approximate Agreement: Converging Diverse Views

The Threshold Approximate Agreement protocol ensures nodes converge to a shared value, even when starting with differing initial values. In decentralized, permissionless blockchains, nodes might develop divergent views on data points, such as node reputation, especially without regular state updates. This protocol aims to reconcile these varying perspectives into a consensus value. The threshold aspect ensures that the agreed-upon value is representative of a significant node subset, making its acceptance logical and robust. For the reputation system, we've set a

lower-bound threshold at 60%. This means the consensus value for a node's reputation score is the lowest score within the top 60% of ratings[18].

Let's denote the set of reputation scores by different peers for a given node as:

$$R = \{r_1, r_2, r_3, \dots, r_n\}$$

where

$$n = \textit{the number of peers reporting the reputation score for the given node}$$

Given the threshold criterion of considering the lowest reputation score from the top 60% of the ratings:

1. We determine the number of top ratings to consider

   $$k = \lceil 0.6 \times n \rceil$$

2. Identify the index for the approximate agreement value

   $$index = n - k + 1$$

3. Determine the approximate agreement value

   $$T = r_{index}$$

With this, we can now look at an example to provide greater insight into the Threshold Approximate Agreement mechanism:

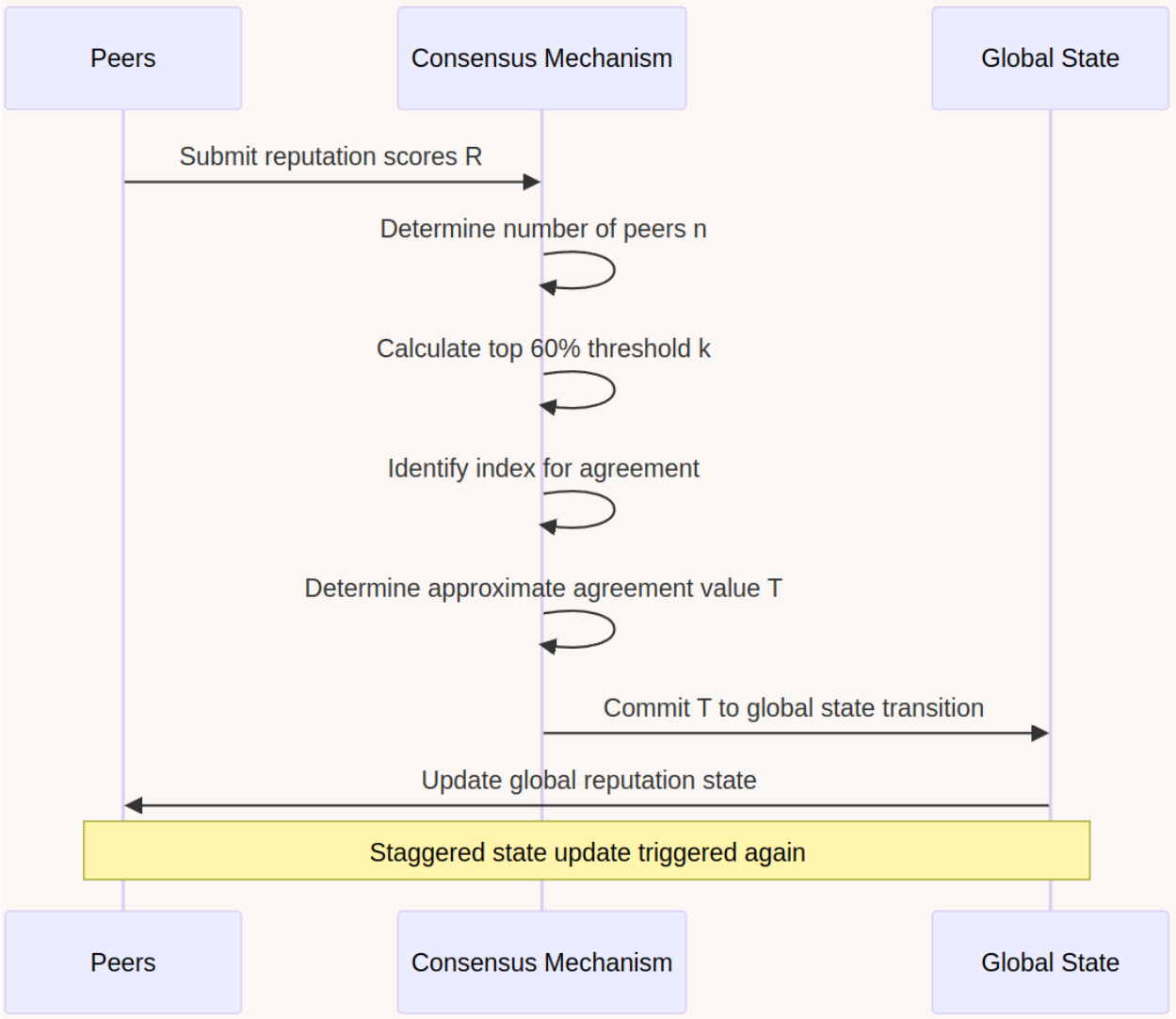
$$R = \{170, 180, 225, 240, 245, 260, 300, 400, 700\}$$

$$k = \lceil 0.6 \times 10 \rceil = 6$$

$$index = 10 - 6 + 1 = 5$$

$$T = 245$$

In the above scenario, the validators responsible for validating a given node's reputation score for a state update would take all the valid peer views of a given node's reputation, include them in a set, take the top 60% and then use the lowest score in the top 60%, or in this case, 245. Once an agreement is reached on the proper value to adopt for a given node's reputation, it is committed to be included in a global state transition. Once the global state transition is completed, all node tracking reputation updates their copy of the global reputation for the updated node to match the consensus value. The process of updating node reputation continues from here, until the randomized, staggered state update for the given node's reputation score is triggered again.



Stake Requirement Calculation

The HoneSTake protocol, as the name suggests, leverages reputation scores to drive a dynamic staking mechanism. This approach aims to bolster decentralization and security in permissionless, peer-to-peer blockchains. While reputation serves as a democratic, one-node-one-vote mechanism, staking remains the primary security measure.

By tying staking to reputation, the protocol ensures that nodes wishing to amplify their influence must launch additional nodes[19]. However, these new nodes start with no reputation, making their initiation costlier and yielding less. In chains that support parallel execution and horizontal scalability, a single operator launching multiple nodes can enhance network throughput.

Although the performance of a chain implementing HoneSTake is beyond this paper's scope, certain protocol designs may offer additional performance benefits alongside the security and decentralization advantages of HoneSTake.

Newcomers, despite facing higher staking requirements than established nodes, still find joining the network appealing. If they anticipate honest participation, the prospect of gradually reducing their stake, thereby increasing their yield, serves as a strong incentive[20].

At its core, the system mandates nodes to stake native network tokens inversely proportional to their reputation score. After establishing a node's reputation score through the threshold approximate agreement process, the next step is to calculate its staking requirement[21].

HoneSTake employs a configurable reputation bucketization combined with a progressive staking requirement. This is determined using a geometric sum of series formula:

$$range(R_{min},\ R_{max});\ where\ R\ =\ Reputation\ Score$$

$$range(S_{min},\ S_{max});\ where\ S\ =\ Stake$$

Given the above, we can define the buckets as:

$$Bucket_{range}\ =\ (R_{max}\ -\ R_{min})\ /\ n$$

$$where$$

$$n\ =\ number\ of\ buckets\ configured$$

Once we have defined the range of the bucket, we can determine r, the common ratio for the geometric series:

$$r = \left(S_{max}/S_{min}\right)^{1/n-1}$$

We can then calculate the increase of stake required for each bucket with:

$$S_{change,\, j=n-i} = S_{i-1} - (S_{max} \times r^{-i})$$

And finally, the required stake for each bucket with:

$$S_j = S_{i+1} + S_{change,\, n-i}$$

Using this formula, we can evaluate the stake requirements for buckets given:

$$R_{min} = 0$$
$$R_{max} = 1,000$$
$$S_{min} = 10,000$$
$$S_{max} = 100,000$$
$$n = 10$$

$$Bucket_{range} = (1,000 - 0) / 10 = 100$$

$$r = (100,000 / 10,000)^{1/9} \approx 1.2915$$

Bucket i	$S_{change,j,\, n-1}$	S_{i+1}	S_j
0	2,945	$S_1 = 98,055$	$S_{max} = 100{,}000$

1	3,759	$S_2 = 94,296$	98,055
2	4,857	$S_3 = 89,439$	94,296
3	6,275	$S_4 = 83,164$	89,439
4	9,102	$S_5 = 74,062$	83,164
5	10,477	$S_6 = 63,585$	74,062
6	13,501	$S_7 = 50,084$	63,585
7	17,510	$S_8 = 32,574$	50,084
8	22,574	$S_9 = 10,000$	32,574
9	0	$S_{min} = 10,000$	$S_{min} = 10,000$

*To retain the upper bound of the maximum stake requirement, $Bucket_0 = S_{max}$

Conclusion

The provided deterministic formula, given the configuration of $range(S_{min}, S_{max})$, $range(R_{min}, R_{max})$ and $Bucket_n$ allows for the precise calculation of each node's stake requirement. This is based on the Threshold Approximate Agreement Protocol and the node's reputation score as reflected in the global state. Consequently, nodes can reach a consensus on eligibility based on their current and required stake across the network.

With the deterministic formula, given a configuration, we can deterministically calculate the stake requirement of each node based on the Threshold Approximate Agreement Protocol and the node's reputation score as reflected in the global state. This ensures that node eligibility based on their current and required stake can be agreed upon throughout the network.

Whistleblower Protocol

In decentralized networks, economic incentives, primarily through staking, often play a pivotal role in deterring malicious activities. However, solely relying on these incentives might not be sufficient to ensure the security and integrity of the system. Therefore, introducing additional layers of security, such as a Whistleblower mechanism, can further bolster the network's resilience against adversarial actions. Historically, implementing whistleblower mechanisms in decentralized, trustless environments has posed challenges[22]. Many existing blockchain networks have refrained from integrating automated whistleblower systems, opting instead for social consensus-driven slashing methods. While these methods have their merits, an automated approach can offer more immediate and objective responses to malicious behaviors. Though the detailed design and implementation of an automated whistleblower mechanism fall outside the purview of this paper, it's essential to acknowledge its potential benefits. Such a system could provide an additional layer of security, ensuring that malicious actors are promptly identified and penalized. Moreover, specific policies related to node jailing and slashing, while crucial, are not discussed in this context.

Jailing & Slashing

The necessity of jailing and slashing in staking-based consensus mechanisms remains a debated topic[23]. Notably, some protocols, like Cardano, have chosen not to implement slashing mechanisms. While certain tokenomic designs theoretically allow for a system that's initially resistant to economic attacks, the dynamics change when the native staking token is freely exchanged.

Over time, as stakers accumulate more tokens, the risk of attack grows unless the network is somewhat centralized by a set of trusted nodes. Fully centralized networks, while immune to the Byzantine Generals Problem, are exposed to risks from trusted counterparties.

For optimal security, we advocate for the potential of significant economic penalties against nodes acting against the network's interest. Nodes attempting to validate invalid transactions, especially those with discrepancies like invalid signatures or double spends, should face penalties to ensure network security[24].

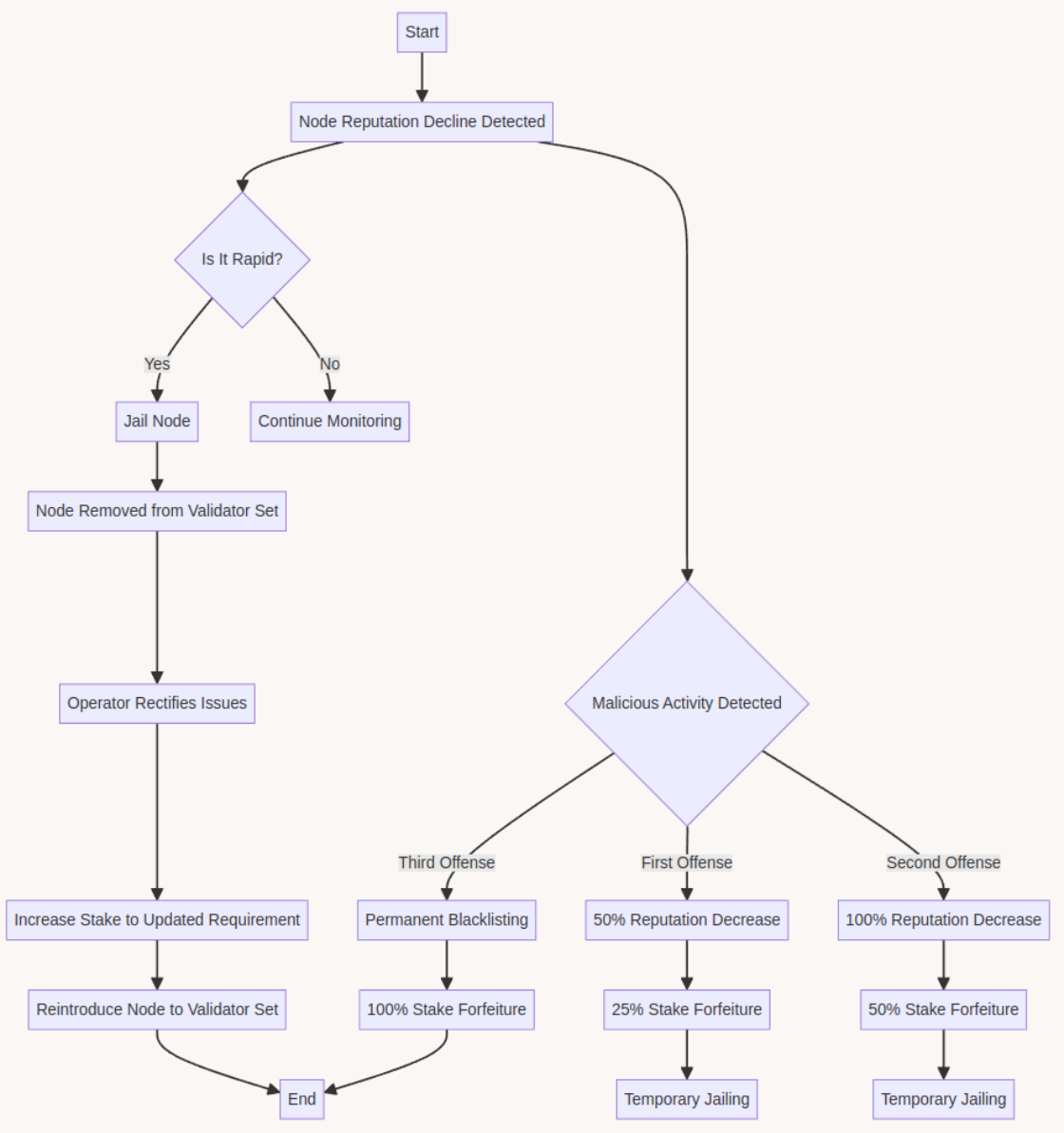
In the HoneSTake protocol, we propose a two-tiered punishment system:

> **Jailing:** Nodes experiencing a rapid decline in reputation scores should be jailed. This acts as a preventive measure against nodes that might have been compromised or are showing signs of malfunction. Jailing removes the node from the validator set temporarily, allowing the node operator to rectify issues and increase the stake to meet their updated required stake based on the decline in reputation.

Slashing: Any malicious activity, especially those attempting to alter the network's state with invalid transactions, should trigger immediate penalties. Such activities, unless caused by a node client bug, typically indicate network attacks. Depending on the severity, these nodes should face slashing penalties[25].

The HoneSTake protocol recommends:

- First offense: 50% reputation decrease, 25% stake forfeiture, and temporary jailing.
- Second offense: 100% reputation decrease, 50% stake forfeiture, and temporary jailing.
- Third offense: Permanent blacklisting and 100% stake forfeiture.



While blacklisting can be challenging in permissionless networks, the HoneSTake protocol's unique design requires new nodes to start with the maximum stake and always only have $1/n$ vote. As the network grows, attacking it becomes increasingly difficult and economically unviable. Potential attackers will likely divert their resources to more vulnerable networks, recognizing the robustness of the HoneSTake protocol.

Conclusion

The HoneSTake protocol introduces a novel approach to security in peer to peer, permissionless blockchain networks. Instead of relying on stake weighted or hash power driven approaches that have a tendency toward centralization, in the most generous interpretation of their design, HoneSTake enables a "1 node, 1 vote" protocol by leveraging the combination of node reputation scores and dynamic staking. The core of the HoneSTake system revolves around a trustless decentralized node reputation management system, that uses local and global views of node reputations, threshold approximate agreement, and efficient global state updates to node reputation. This reputation score tracking mechanism is combined with dynamic staking, in which nodes with higher reputations can reduce the minimum stake required to be eligible as a validator and governing node.

This mechanism makes it more difficult, and expensive, to take control of the network. The nature of this mechanism is such that it resists centralization. Wherein traditional Proof of Stake networks enable nodes to accumulate stake, both through token purchases and through delegation, and Proof of Work networks enable nodes to acquire more hash power, either by pooling hardware (similar to delegation in PoS), or outright purchasing it, HoneSTake grants no extra power to nodes with stakes beyond the minimum required of the given node based on its reputation. To achieve more power in the network, assuming all else equal, the operator would have to launch another node, which would enter the network with no reputation, and be required to stake the maximum amount.

Combining HoneSTake with a punitive jailing and slashing protocol, whether automated via a Whistleblower protocol, or subject to social consensus, leads to a powerful economic incentive to align operators with the network protocol. Further, the HoneSTake mechanism makes it possible for the network to become hyper-decentralized, and as a result, more resistant to non-economic actors that may attempt to attack the network. This protocol can be used to determine validator eligibility, as well as for on-chain governance related voting, ensuring the democratic principle of 1 node, 1 vote, is realized, centralization is resisted, and security of the network is enhanced.

[1] Singh, Hosen & Yoon, “Blockchain Security Attacks, Challenges and Solutions for the Future Distributed IoT Network”, 2021
[2] Qureshi, “A Survey of the Concept of Blockchain security Challenges and Risks”, *International Journal of Computer Trends and Technology*, 2019
[3] Wang, Ge & Liu, “On the Security of Permissionless Blockchain Systems: Challenges and Research Perspective”, 2021
[4] Kurdi, “HonestPeer: An enhanced EigenTrust algorithm for reputation management in P2P systems”, 2014
[5] Battah, Iraqi & Damiani, “Blockchain-Based Reputation Systems: Implementation Challenges and Mitigation”, 2021
[6] Bellini, Iraqi & Damiani, “Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey”, 2020
[7] Zhang, et. al., “A Reputation-Based Mechanism for Transaction Processing in Blockchain Systems”, 2022
[8] Kamvar, Schlosser & Garcia-Molina, “The Eigentrust algorithm for reputation management in P2P networks”, 2003
[9] Wu, “A distributed trust management model for mobile P2P networks”, 2012
[10] Hu, et. al, “A Reputation Based Attack Resistant Distributed Trust Management Model in P2P Networks”, 2010
[11] Hu, Wu, & Zhou, “RBTrust: A Recommendation Belief Based Distributed Trust Management Model for P2P Networks”, 2008
[12] Singh, et. al., “Distributed Trust and Reputation Management for Future Wireless Systems”, 2022
[13] Shen, et. al., “A P2P-Based Infrastructure for Adaptive Trustworthy and Efficient Communication in Wide-Area Distributed Systems”, 2014
[14] Paschke & Alnemr, “The Rule Responder Distributed Reputation Management System for the Semantic Web”, 2010
[15] Hatzivasilis & Manifavas, “Building Trust in Ad Hoc Distributed Resource-Sharing Networks Using Reputation-Based Systems”, 2012
[16] Takao, Sugiura & Ishikawa, “Approximate Fault Tolerance for Sensor Stream Processing”, 2020
[17] Freitas, Kuznetsov & Tonkikh, “Distributed Randomness from Approximate Agreement”, 2022
[18] Nowak & Rybicki, “Byzantine Approximate Agreement on Graphs”, 2019
[19] Avyukt, Ramchandran & Krishnamachari, “A Decentralized Review System for Data Marketplaces”, 2021
[20] Sun, et. al, “Voting-Based Decentralized Consensus Design for Improving the Efficiency and Security of Consortium Blockchain”, 2021
[21] Bu, Wu & Li, “RepShardChain: A Reputation-Based Sharding Blockchain System in Smart City”, 2022
[22] Daian, Pass & Shi, “Snow White: Robustly Reconfigurable Consensus and Applications to Provably Secure Proof of Stake”, 2019
[23] Bhudia, et. al., “Game Theoretic Modelling of a Ransom and Extortion Attack on Ethereum Validators”, 2023
[24] Stone, “Delayed Blockchain Protocols”, 2018
[25] Amoussou-Guenou, et. al. “Committee-Based Blockchains as Games Between Opportunistic Players and Adversaries”, 2023