GAUNTLET

Gauntlet Research Report

Protocol Design Assessment

An analysis of the THORChain Incentive Pendulum and Staking Economics

THORCHAIN

June 2020

Contents

Disclaimer

This report is for informational purposes only and does not constitute an offer to sell, a solicitation to buy, or a recommendation for any security, nor does it constitute an offer to provide investment advisory or other services by Gauntlet Networks Inc. No reference to any specific security constitutes a recommendation to buy, sell or hold that security or any other security. Nothing in this report shall be considered a solicitation or offer to buy or sell any security, future, option or other financial instrument or to offer or provide any investment advice or service to any person in any jurisdiction. Nothing contained in this report constitutes investment advice or offers any opinion with respect to the suitability of any security, and the views expressed in this report should not be taken as advice to buy, sell or hold any security. The information in this report should not be relied upon for the purpose of investing. In preparing the information contained in this report, we have not taken into account the investment needs, objectives and financial circumstances of any particular investor. This information has no regard to the specific investment objectives, financial situation and particular needs of any specific recipient of this information and investments discussed may not be suitable for all investors. Any views expressed in this report by us were prepared based upon the information available to us at the time such views were written. Changed or additional information could cause such views to change. All information is subject to possible correction. Information may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

Part I

Background

1 Overview

THORChain is a decentralized liquidity protocol that enables cross-chain swaps. These swaps allow for users to perform cross-chain token exchanges without the need of a trusted third-party, such as a centralized exchange or custodian. In order to settle cross-chain swaps, the network uses a staked native asset that serves as collateral for validators providing cross-chain proofs. The network's native asset, known as RUNE, serves as a settlement asset for each of the continuous liquidity pools (CLPs) and also as collateral posted by nodes ensuring network security. These continuous liquidity pools resemble those pioneered by Uniswap, albeit with the unique feature that they facilitate cross-chain transaction rather than transactions between ERC-20 tokens. THORChain's native asset code is written using the Cosmos SDK, which allows for inter-blockchain communication (IBC) with other Cosmos SDK chains.[1] Users can swap RUNE into any other asset in a hub and spoke fashion or directly swap from asset to asset but unlike Uniswap V2 instead of a fixed fee, a slippage based fee is collected. This fee is estimated based on pool liquidity and denominated in the *output token*. Collecting the fee in the output token is different from the policies of other constant function market makers, such as Uniswap [1, 2]. THORChain incentivizes RUNE holders to be bonders who validate transactions by operating nodes or act as stakers who provide liquidity to the pools in the form of external and native assets. Bonders must have a minimum quantity of 1MM RUNE that can be staked to validate transactions, akin to the minimum staking size limits of Ethereum 2.0, Tezos, and Cosmos. The proportion of overall system rewards allocated between bonding and staking is continuously dictated by a protocol mechanism called the incentive pendulum.

As this system has a number of principal agents with unique incentives, we turn to economic simulation to estimate the safety and financial properties of the THORChain system.

1.1 Simulation Analysis

1.1.1 Background on Agent-Based Simulation

The main tool that we use to analyze THORChain's protocol is agent-based simulation (ABS). ABS has been used in a variety of contexts in quantitative finance, including to estimate censorship in cryptocurrency protocols [4], detect fraudulent trading activity in CFTC exchanges [5], and in stress testing frameworks from the European Central Bank [6, 7] and the Federal Reserve [8, 9]. These models can provide invaluable information on the behavior of complex systems. This has made ABS widely used in industries such as algorithmic trading and self-driving car deployment.

1.1.2 Gauntlet Simulation Environment

The Gauntlet platform, which was used for all simulations and results in this report, provides a modular, generic ABS interface for running simulations directly against Ethereum smart contracts as well as Python representations of contracts. Here we use the latter as the Solidity contracts are still in development. In this system, the agent models are specified via a Python domain-specific language (DSL), akin to Facebook's PyTorch[10]. Agents can also interact with non-blockchain modules, such as historical or synthetic market data and/or other off-chain systems. The DSL hides the blockchain-level details from the analyst, allowing the end-user to develop

[1] More info on Cosmos and the Tendermint protocol can be found on the Cosmos website

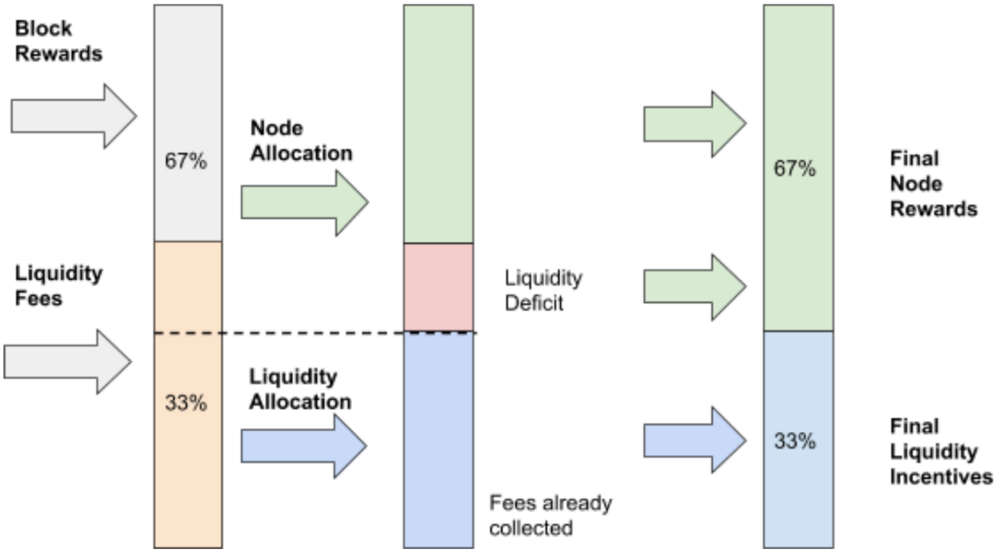


Figure 1: Incentive Pendulum [3]

strategies that can migrate from one smart contract to another, should they have similar interfaces. Most of the platform's design is inspired by similar platforms in algorithmic trading that allow for quantitative researchers to develop strategies that execute over multiple exchanges (with varying order books, wire protocols, slippage models, etc.) without having to know these low-level details. Moreover, the non-blockchain portions of the simulation are analogous to trading back-testing environments,[11] so that agents are interacting with realistic order books and financial data.

1.2 Goals of the Analysis

Bonding and staking are both critical to the growth of the THORChain ecosystem. Staking provides the primary source of liquidity to THORChain, which serves as the core driver of network activity by providing a service and collecting external fees added back into the network. Bonding is critical to ensure that the system is secure and all stakers can be confident that their assets in the pools are safe from attack.

The main goal of this analysis is to understand how the incentive pendulum drives bonder and staker behavior in the THORChain protocol, along with key parameter inputs that will influence overall protocol health and efficiency. Our analysis aims to stress test the bonding and staking incentives under a wide range of initialization conditions, market contexts and agent assumptions to ensure that the current parameter choices lead to a secure and a robust system.

The THORChain protocol is at risk when the ratio of bonded to staked capital drops below 1 [3], at which point a set of attackers can realize gains by coordinating to steal assets. Another risk is if large amounts of liquidity is withdrawn since if the pools dry up system activity grinds to a halt. THORChain mitigates both of these by heavily shifting rewards towards bonding in the former and towards staking in the latter. Another focus of the analysis is to assess the protocol's wealth distribution and utilization, especially how to balance the trade-offs of rewarding early adopters versus enticing new actors to participate.

Part II

Staking & Bonding Analysis

2 Incentives

THORChain liquidity network uses Proof of Stake (PoS) at the networking layer, along with Continuous Liquidity Pools (CLP) to facilitate asset swaps. Participants of THORChain earn rewards by either bonding or staking RUNE. A bonder can spin up a THORChain node and bond RUNE to sign transactions and provide network security. Stakers provide CLP liquidity by staking RUNE and external asset pairs to the pools. THORChain proposes a novel mechanism, called the incentive pendulum, to redistribute the total system income composed of the RUNE inflation reward and liquidity fees to participants, as a way to incentivize them to balance the network capital distribution. In particular the proportion x of system income distributed to stakers is given by

$$x = \frac{b - s}{b + s} \tag{1}$$

where b and s denote the total bonded and total staked capital, with the remainder given to bonders.

2.1 Staking

2.1.1 Returns to Staking RUNE

THORChain is a cross-chain CLP network that relies on participants to stake tokens to provide liquidity. Participants stake equivalent value of the external asset and RUNE to the liquidity pools, and the traders pay liquidity fee for asset swaps. The incentive pendulum collects system income from both inflation rewards and liquidity fees, and directly distributes the adjusted staking reward into pools. Stakers can claim the staking reward according to their shares of the liquidity in the pools. Upon staking, they receive a number of staker units determined by

$$x = \frac{(R + A)(rA + aR)}{4RA} \tag{2}$$

where R and A are the pool RUNE and asset balances including the staked amounts of RUNE and asset r and a. Upon unstaking, the pool pays out prorata relative to the total number of staking units outstanding. Calculating the P&L for staking is difficult for the stakers as they need to factor in the pool size, and the system's overall staking and bonding RUNE for the incentive pendulum reward adjustment. In particular a larger pool likely reduces the slip-based liquidity fee per trade but this in turn is better for system liquidity and should lead to an increase in trade volume.

2.1.2 RUNE/USD Returns

Staking requires the agent to bear the price risk of RUNE for the duration that one has assets staked in the protocol. The market price of RUNE is modeled using Merton's jump diffusion model with a jump process composed of log normal jumps driven by a Poisson process and a diffusion process defined by geometric Brownian motion. We chose this model as it is the standard model used by the Federal Reserve for defaultable securities [12]. Assessing THORChain's security relies on stress testing what happens to RUNE when there is capital flight and/or a 'bank run' (e.g. bonders simultaneously unbonding). Therefore, our model must consider default conditions and cannot use standard drift-diffusion models like geometric Brownian motion or Ornstein-Uhlenbeck processes. The

Merton jump-diffusion model is well-understood and analytically tractable, using this model also allows for us to compute well-defined empirical statistics in simulation. Formally, the jump diffusion price process at time t, S_t, is defined by

$$\frac{dS_t}{S_t} = (\mu - \lambda\mu_k)dt + \sigma dW_t + kdX_t \tag{3}$$

where W_t is the Wiener process, μ and σ denote the percentage drift and volatility, X_t is a compound Poisson process with intensity λ and jump magnitude k which is parameterized by a log normal with mean γ and variance δ^2 in

$$\ln(1+k) \sim \mathcal{N}(\gamma, \delta^2) \tag{4}$$

and $\mu_k = e^{\gamma+\delta^2/2} - 1.$

Agents try to infer the value of these parameters based on recent observations of the "in-sim" market prices. Analogous to traditional market making, stakers achieve ideal low-risk returns when the price is mean-reverting around a stable level.

2.1.3 Calculating Staking Returns

The main factors that go into computing the expected return on investment (ROI) of staking at an agent level are the expected growth rate of liquidity fees detailed in Section 5.5, the incentive pendulum reward readjustment from Equation 1 and impermanent loss.

Ignoring gains from liquidity fees and assuming constant pool share, impermanent loss [13][14] is a term that refers to the change in portfolio value of a staker's pool share relative to simply holding tokens in wallets or centralized exchanges when prices vary from the original staking price. We note that impermanent loss has been studied in traditional finance under the guise of volatility harvesting [15]. From the XYK formula underlying the pool computations if the pool has quantities X and Y of assets x and y respectively then the product $XY = K$ is invariant for some constant K through a transaction (though over time this grows as fees are added). As an example suppose x denotes RUNE, y denotes the USD Stablecoin Dai (DAI), p denotes the price of RUNE in DAI and assume that the pool is efficiently priced so $X/Y = p$ then originally

$$X = \sqrt{\frac{K}{p}} \quad Y = \sqrt{Kp} \tag{5}$$

Thus the overall pool value V is

$$V = p_x X + p_y Y = p\sqrt{\frac{K}{p}} + 1\sqrt{Kp} = 2\sqrt{Kp} \tag{6}$$

Suppose that the price moves by a factor of $c > 0$ so the new price $p' = cp$. Then the new pool value V' is

$$V' = p'_x X' + p'_y Y' = cp\sqrt{\frac{K}{cp}} + 1\sqrt{Kcp} = 2\sqrt{Kcp} \tag{7}$$

Using the constant pool share assumption, if a staker had instead held the assets instead of staking their portfolio share scaled to the entire pool W would be worth

$$W = p'_x X + p_y Y' = cp\sqrt{\frac{K}{p}} + 1\sqrt{Kp} = (c+1)\sqrt{Kp} \tag{8}$$

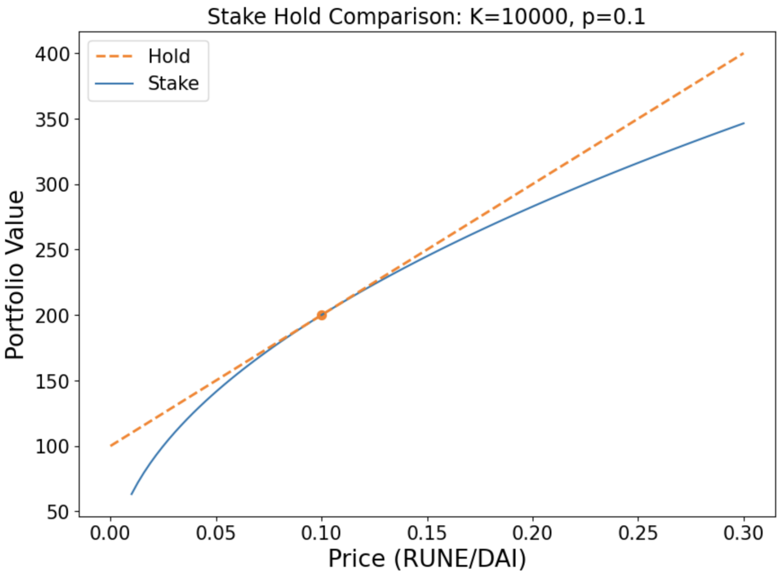


Figure 2: Impermanent Loss

Note that $c+1 \geq 2\sqrt{c}$ by the AM-GM inequality so there is always a relative drop in value in this no fee constant pool share context. Some intuition for this comes from

$$\frac{\mathrm{d}X}{\mathrm{d}p} = \frac{-1}{2}\sqrt{\frac{K}{p^3}} < 0 \quad \frac{\mathrm{d}Y}{\mathrm{d}p} = \frac{1}{2}\sqrt{\frac{K}{p}} > 0, \tag{9}$$

so as p increases so does Y while X decreases and as p decreases Y does as well while X increases. More generally, changes to p increase the delta to the weaker performing asset since initial stake and decrease the delta to the stronger asset.

$$\frac{\mathrm{d}^2V}{\mathrm{d}p^2} = \frac{\mathrm{d}}{\mathrm{d}p}\left(\frac{\mathrm{d}V}{\mathrm{d}p}\right) = \frac{\mathrm{d}}{\mathrm{d}p}\left(\sqrt{\frac{K}{p}}\right) = \frac{-1}{2}\sqrt{\frac{K}{p^3}} < 0 \tag{10}$$

shows that V is concave and the portfolio delta is inversely related to price, akin to the gamma of a short option position. Holding the assets has a linear delta to price and the holding value W is tangent to the pool value V at the staking price and from concavity it follows that $W \geq V$. This is seen in Figure 2. More formally, a replicating portfolio for a constant product market involving a short option position can be found at [16] and [17].

In the simulation we specify the impermanent loss adjusted staking ROI $r_{\text{staking adj}}$ as a function of the agent staking ROI estimating liquidity fees and incentive pendulum swings r_{staking} and the RUNE/USD ROI $r_{\text{RUNE/USD}}$ and setting the price change factor $c = 1 + r_{\text{RUNE/USD}}$ we get

$$r_{\text{staking adj}} = \frac{2\sqrt{1 + r_{\text{RUNE/USD}}}}{2 + r_{\text{RUNE/USD}}}(1 + r_{\text{staking}}) - 1 \tag{11}$$

2.2 Bonding

2.2.1 Returns to bonding RUNE

THORChain, as a PoS network, relies on nodes bonding RUNE to provide its network security. The bonding nodes earn yield via the inflation curve set by the protocol. The inflation curve is set to emit 1/6th of the remaining RUNE reserve $R_{reserve}$ each year, and the inflation rate will be reduced to 2% after ten years. Assuming the average block time is five seconds, the number of blocks per year is 6311390, the inflation reward per block i_{block} can be represented as:

$$i_{block} = \frac{R_{reserve}}{6 \cdot 6311390} \tag{12}$$

The inflation mechanism incentivizes early adopters since they will be rewarded due to a higher amount of inflation reward. Since it is desirable to be an early bonder the selection process involves competitive bidding in the form of a standby queue sorted by RUNE amount where the node that offers to bond the most RUNE is first in line. Every 3 days the oldest node is relieved and the top two nodes in the standby queue are added until the system is at max capacity at which time the process becomes two nodes out, two nodes in. This is referred to as a churn. For every block bonded nodes receive a bond unit and upon a churn or an unbond rewards are paid out prorata relative to the total number of outstanding bond units. This further entices early bonders since they receive a higher proportion of the rewards when the number of bonded nodes is small along with an incentive for bonders sitting on RUNE to participate when the system is short on validators.

Computing the returns on bonding exactly is difficult due to the high dimensional evaluation space stemming from its game theoretic nature. While bond units are paid out uniformly regardless of how much a node is bonding, nodes with more capital can jump the standby queue and re-enter quickly upon a churn. A rich bonder perhaps would want to determine an optimal capital split amongst several nodes to maximize a payoff that is constantly changing. For simplicity we only allow each agent to run one node though we leave it open as a point for future exploration. With that constraint, the return is computed from the estimated number of nodes over the course of bonding from the current number of nodes bonding and in the standby queue, incentive pendulum adjustments, bonding costs etc.

2.2.2 Cost of Bonding

The cost of hosting a node will vary depending on the specific infrastructure that each bonding node utilizes. In addition, the electricity cost and other cost factors involved in bonding should also be taken into consideration. We assume the expected cost of running a bonding node is around $1000 per month [3].

3 Key Protocol Variables

3.1 Minimum Bonding Limit

The minimum bonding limit specifies the minimum amount of RUNE required to bond for running a THORChain node. In the protocol this is currently set to 1MM RUNE which restricts bonding to a small set of participants. A low minimum bonding limit will encourage more RUNE holders to bond, increase the bonder diversification and help the system to maintain a safe bonded-to-staked ratio. However, it will also increase the network communication cost. We run simulations that explore the minimum bonding limit variable, to demonstrate the likely effects on the safety of the network.

3.2 Total RUNE Supply

The total circulating RUNE supply is important relative to the minimum bonding limit as together their quotient defines an overall percentage holding required to be a bonding. Currently this is roughly 160MM RUNE so each bonder must hold at least 0.6% of the total supply.

Part III
Agent-Based Simulation

4 Agent Types

4.1 Liquidity Provider

A liquidity provider (LP) who owns RUNE and external assets (ETH, BTC, BNB, etc) is eligible to add to the CLPs. They evaluate the returns for staking and decide to stake if the returns are better than the risk-free rate. Staking is an all or nothing decision at any given point; they will unstake if the returns for staking dip below the risk-free rate.

4.2 Bonder

A RUNE holder who owns more than the minimum bonding limit is qualified to be a bonder and run a THORChain node. The decision process for a bonder is more complex than a staker, since a bonder can bond RUNE, stake RUNE and also reinvest the bonding reward. Here's a list of some of the factors a bonder needs to take into account: bonding ROI, staking ROI, risk-free returns, staked RUNE, bonded RUNE, bonding rewards, bonding costs. Bonders chooses the strategy that maximizes the return according to their current bonded and staked positions.

4.3 Staker

A staker is a bonder whose initial allocation is below the minimum bonding limit and acts as an LP until they accrue enough rewards to run a node. Their liquidity providing decision making is additionally affected by their personal incentive alignment with the protocol, a model parameter we define as bonder altruism.

5 Key Model Parameters

5.1 Bonder Altruism

Because bonders by requirement must hold a significant portion of RUNE they have a significant long term financial incentive to keep THORChain secure and help it grow in order to realize the subsequent substantial RUNE appreciation. In the simulation we make this explicit by defining the altruism adjusted bonding ROI $r_{\text{bonding adj}}$ as a function of the agent bonding ROI r_{bonding}, the bonding cost ROI $r_{\text{bonding cost}}$ and a protocol health objective function h

$$r_{\text{bonding adj}} = r_{\text{bonding}} - r_{\text{bonding cost}} + \varphi \ln\left(\frac{h_{t+1}}{h_t}\right) \tag{13}$$

where φ is a scalar denoting altruism and h is defined as

$$h_t = c_t s_t \tag{14}$$

where s_t is the total value staked at time t and c_t is defined as

$$c_t = \begin{cases} \frac{e^{r_t - 1} - 1}{e - 1} & r_t < 2 \\ \frac{1}{r_t - 1} & r_t \geq 2 \end{cases} \tag{15}$$

and $r_t = \frac{b_t}{s_t}$ is the ratio of the total bonded and staked capital at time t. We choose c_t to harshly penalize a decreases in security from underbonding and more minorly deduct inefficiency from overbonding. $c_t = 1$ is maximized at the optimal ratio $r_t = 2$, while $c_t = 0$ when the system is insecure at $r_t = 1$ and exponentially increasing as $r_t \to 2$ and more gradually decreasing but still convex for an overbonded state where $r_t > 2$ and $c_t \to 0$ as $r_t \to \infty$. The $\ln\left(\frac{h_{t+1}}{h_t}\right)$ term is chosen to incentivize actions that increase the objective and hinders those that decrease it. The magnitude is weighted to the proportional change while keeping bonder decision Markovian for simplicity.

5.2 RUNE/USD Price Drift

A key factor that determine the market price trajectory is the drift μ from Equation 3. It affects the decisions of liquidity providers through changes in liquidity fees and impermanent loss and bonders in their cost calculations. In practice bonders are likely to be more altruistic in higher drift regimes and we leave this as a potential avenue for future work. The drift cannot be controlled by any entity and is treated as an exogenous term in our simulations. We explore a wide range of values for RUNE/USD annualized drift, or price return to simulate the market demand for RUNE.

5.3 RUNE/USD Price Volatility

Like RUNE price drift, RUNE price volatility σ from Equation 3 is another exogenous factor that will influence the decisions of agents. Spikes in volatility often come with an increase in trading demand and liquidity rewards and cause disutility to large RUNE holders such as bonders. We explore a wide range of values for RUNE/USD annualized volatility (standard deviation of the logarithmic returns) to simulate the price fluctuations for RUNE.

5.4 RUNE Initial Allocation

The Pareto distribution is commonly used by practitioners in modeling the long-tail wealth or income distribution. To produce a realistic RUNE distribution at simulation initialization, we incorporate the RUNE distribution data from the May 2020 THORChain network. Since the largest RUNE holder is fairly bounded we use a truncated Pareto distribution with probability density function f defined by

$$f(x) = \frac{\alpha L^{\alpha} x^{-\alpha-1}}{1 - \left(\frac{L}{H}\right)^{\alpha}} \tag{16}$$

Given the circulating supply of 160MM RUNE, the largest holder having 4MM, the top 100 holders owning 72% of RUNE of which roughly 45 exceeding the 1MM minimum bonding limit with a combined 82MM capital available for bonding, we choose $\alpha = 0.1$, $L = 1$ and $H = 25$.

5.5 Liquidity Fees

In traditional liquid financial markets, higher drift and volatility are correlated with an increase in trading demand. We aim to model this effect with historical data on Uniswap trades but to remain flexible to the rapidly evolving nature of decentralized exchanges we parameterize this assumption where the annualized growth rate G of the liquidity pools is defined as

$$G = g \max(\min(\beta_\mu \mu_{RUNE/USD} + \beta_\sigma \sigma^2_{RUNE/USD}, m), 1) \tag{17}$$

where g is the base growth rate, $\mu_{RUNE/USD}$ and $\sigma^2_{RUNE/USD}$ are the annualized RUNE/USD ROI and volatility and β_μ, β_σ and m are parameterized constants.

5.6 Bonding Costs

The costs of running a node also include the possibility of loss of assets from being the target of an attack. Since these are more variable than electricity and infrastructure costs this is also a parameter of interest. Additionally when the RUNE price dips and approaches 0 the bonding costs in RUNE approach infinity so we define a cap for this at 10% annualized ROI.

6 Measure of Protocol Success

6.1 Bonded to Staked Ratio

THORChain is safe when the value of the bonded capital is higher than the staked capital. If the value of the staked capital is greater than the bonded capital, bonders will have the incentive to steal the staked assets. The explanation of this attack is described in more detail in [3].

The goal of the incentive pendulum is to continually allocate the staking and bonding rewards such that the system can converge to the optimal 200% bonded to staked ratio [3]. The bonded to staked ratio is defined as the amount of the bonded RUNE divided by the staked RUNE. We average c_t from Section 5.1 to measure the performance of the incentive pendulum. We also calculate the duration that the bonded to staked ratio is below 100% to quantify the risk of assets being stolen by bonders.

6.2 Total Value Staked

Total assets staked are a good measure of the overall strength of the protocol. The mean and the variance of total value staked are both measured in the simulation since a large mean is informative of better system capacity while a large variance is a negative indicator since health is likely concave in staked value with liquidity shortages being especially damaging. Another measure is based on averaging h_t defined in Section 5.1. We also calculate the duration that there are no assets staked in the pools to quantify the risk of liquidity failure.

6.3 RUNE Utilization

The system is efficient when a high percentage of RUNE is being used towards either staking or bonding instead of sitting in wallets. We measure this with the average capital utilization of all agents throughout the simulation. In the current distribution with 80MM RUNE available for bonding and 160MM in circulation there is a trade-off between security and efficiency since at full utilization the bonded to staked ratio is at most 1.

6.4 RUNE Token Distribution

A PoS consensus protocol requires a heterogeneous set of bonders to secure the network. If there are either too few bonders or only a small number of bonders who control the large majority of the bonded tokens, the network could devolve into an effectively centralized system. We want to ensure that the CLPs provided by THORChain are truly in a decentralized state.

The Gini coefficient is a common measure of statistical dispersion among a set of quantities. When the Gini coefficient is 0, the network is fully decentralized and the distribution of RUNE is uniform, whereas when it is 1 the network is purely centralized (e.g. a single entity controls the whole network). Formally, given a RUNE quantity R_i for the ith agent in the system and the system has n agents. The Gini coefficient $G(R_1, \ldots, R_n)$ is defined as:

$$G(R_1, \ldots, R_n) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |R_i - R_j|}{2n \sum_{i=1}^{n} R_i} \tag{18}$$

7 Simulation Initialization

To bootstrap the system at initialization 115MM RUNE is allocated according to a truncated Pareto distribution to 100 bonders with those with less than 1MM tagged as stakers. Another 45MM RUNE is distributed to 100 liquidity providers uniformly at random. The remainder of the total available RUNE supply is the reserve set at 340MM which is distributed over time through the inflation curve as block rewards.

To ensure the system has a stable start, staking only begins after 12 nodes are active with the network having a minimum capacity of 4 nodes and a max of 99. The risk free rate is set to 1% annualized and the base liquidity pool growth rate is set to 5%. The length of an individual simulation is 2 years with a step size of 3 days.

Part IV

Simulation Results and Analysis

8 THORChain Initialization

8.1 RUNE Initial Allocation

Following the parameterization described in Section 5.4, the initial distribution of RUNE prior to launch is a key determinant of the decentralization of the protocol. As such, we run a total of 900 simulations across a search space of

$$\begin{aligned} &\text{Asset Truncated Pareto Alpha} \in \{0.1, 0.2, 0.4, 0.6, 0.8, 1\} \\ &\text{Asset Truncated Pareto Max} \in \{5, 10, 25, 50, 100, 200\} \end{aligned}$$

In addition to measuring RUNE disparity among the holders, a very uneven allocation could destabilize the protocol by way of dependence on the discrete actions of the richest agents. For context, α determines the shape and lower values create a more gradually sloping distribution with a fatter tail while the max M fixes the maximum ratio between the holdings of the richest and poorest bonding or staking agents so higher values make the distribution less uniform. This can be seen more clearly in Figure 3.

In the right heatmap of Figure 4 we see that the bonded to staked ratio objective is increasing with the truncated Pareto max and decreasing with the truncated Pareto alpha. This is indicative of the effect where a

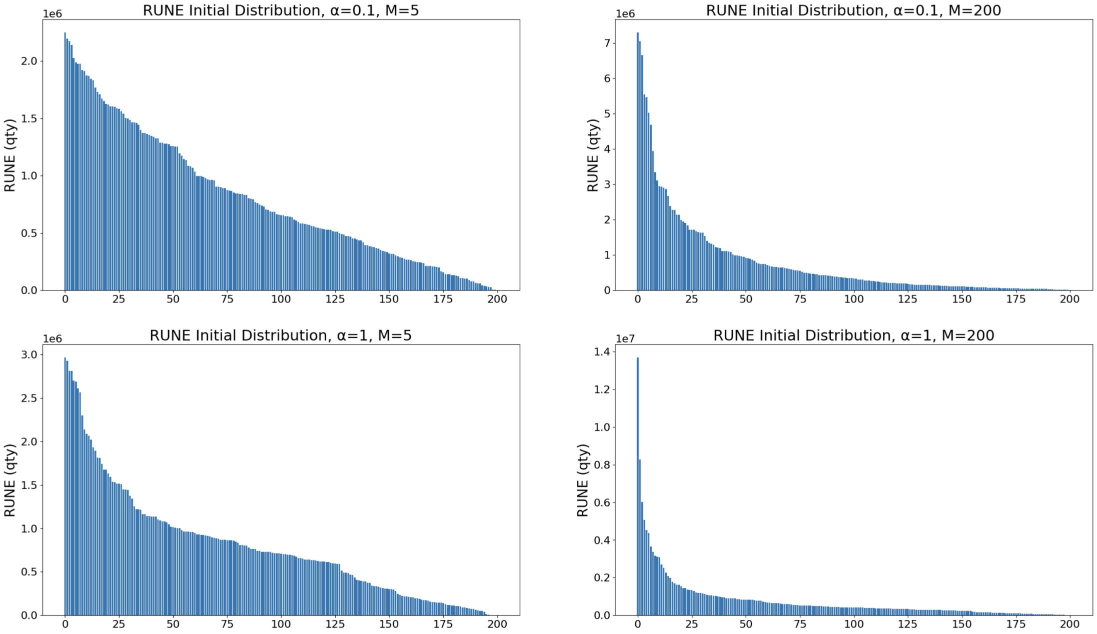


Figure 3: Truncated Pareto Initial Distributions

larger M increases the total amount of available bonded capital and additionally the smaller α spreads the capital among more bonders. Since the default setting is of 115 MM RUNE distributed across 100 agents, higher M is actually more favorable in concentrating capital in holders above the minimum bonding limit since the average agent barely has more than the limit. For the same reason smaller α leads to a slight increase in bonding RUNE but more prominently the top heavy shape produced by higher α consolidates the additional bonding RUNE in a select few bonders, causing the system to experience sizable drops in security when rich nodes are churned.

On the left side we see the terminal Gini coefficient largely increasing with M but also peaking around $\alpha = 0.5$. As seen in Figure 3, increases in M strongly skew the initial RUNE distribution towards inequality which carries over to the terminal distribution. The rise in Gini as α increases to 0.5 is also similarly explained though the effect is weaker. The decrease in Gini as α ranges from 0.5 to 1 touches upon an interesting effect at initialization. Since the system starts with a bottleneck on bonding capital, the incentive pendulum swings past its target ratio of 2 to give income to bonders. This in turn lengthens the time needed for stakers to accumulate enough RUNE to become bonders and correct the pendulum so the longer this cycle persists the more phenomenal the returns early bonders receive compared to early LPs. Since bonders have sole access to the superior initial ROI stream there is a natural tendency for RUNE towards greater inequality. In particular, bonders with just above the minimum limit are heavily rewarded since they collect the same bond units as the top holder while bonding and don't have to wait too long in standby due to the bonder shortage. In this context the Gini for a plutocracy emerges higher than for an autocracy as the disparity grows between bonders as first class citizens and LPs as their second class counterparts. A final note is that since the system is dependent on some degree of altruism it's easier to make assumptions about a benevolent autocrat due to their large individual stake but this becomes harder when spread to an entire group of moderately wealthy oligarchs.

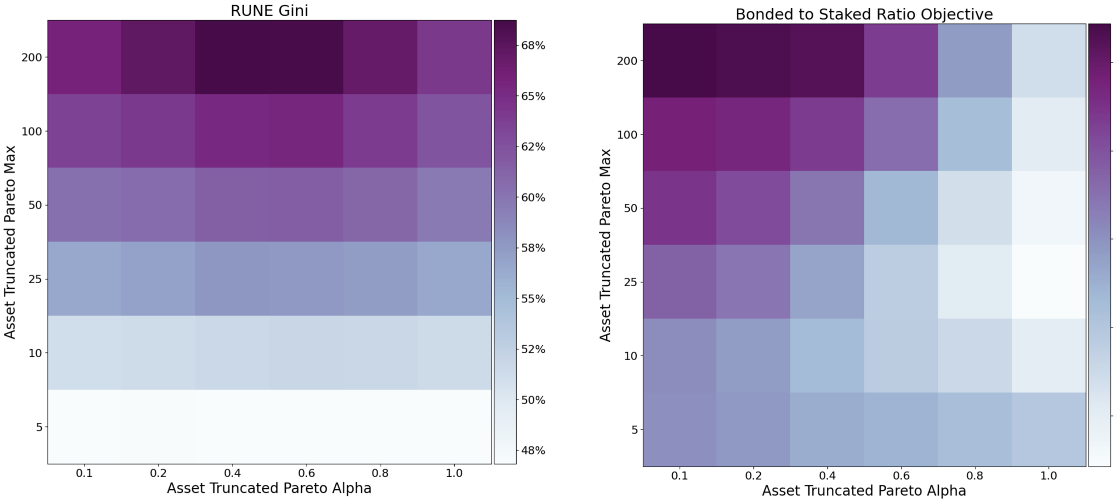


Figure 4: Initial Allocation Decentralization and Security Objective

8.2 Minimum Bonding Market Share

As discussed in Sections 3.1 and 3.2 the minimum bonding limit and total circulating RUNE quantity together determine the percentage of RUNE available for bonding, which informs an upper bound for the bonded to staked ratio. We run a total of 1350 simulations across a search space of

$$\begin{aligned}
&\text{Minimum Bonding Limit (MM)} &&\in \{0.5, 1, 1.5, 2, 2.5, 3\} \\
&\text{Bonder Staker RUNE Qty (MM)} &&\in \{80, 120, 160\} \\
&\text{LP RUNE Qty (MM)} &&\in \{30, 50, 70\}
\end{aligned}$$

In particular we are interested in the ratio q in relation to the bonded to staked ratio where

$$q = \frac{\text{Minimum Bonding Limit}}{\text{Total Circulating RUNE}} = \frac{\text{Minimum Bonding Limit}}{\text{Bonder Staker RUNE Qty} + \text{LP RUNE Qty}} \tag{19}$$

As seen in Figure 5 there is a strong inverse relationship between q and the bonded to staked ratio. This is in line with expectations since the higher the market share needed to bond, the less bonded capital available to secure the network there is. This in turn pushes the system towards an underbonded state. As a plot clarification, since the bonded to staked ratio can fluctuate throughout the simulation, the y-axis measure used for the left chart is the geometric mean of the ratios across all the time steps for a given set of a parameters.

As more RUNE enters the system the market share requirement will decline over time. If the system becomes too consistently overbonded or the maximum node capacity is reached, it may become necessary to raise the limit. Note that the system veers from overbonded to underbonded around the 0.8% mark, suggesting that the current setup of a 1MM minimum bonding limit and a total circulating RUNE supply of 160MM is close to optimal, with a trend towards inefficiency rather than insecurity,

In the right plot of Figure 5 we see the percentage of agents that are initially bonders drops quickly as q increases. Additionally the percentage of stakers who accumulate enough RUNE to become run a node also drops as q increases and is capped by 100% when q is near 0% and close to 0% when q is more than 1.5%. Note that stakers running nodes makes it easier for other stakers to follow by increasing their liquidity rewards due to

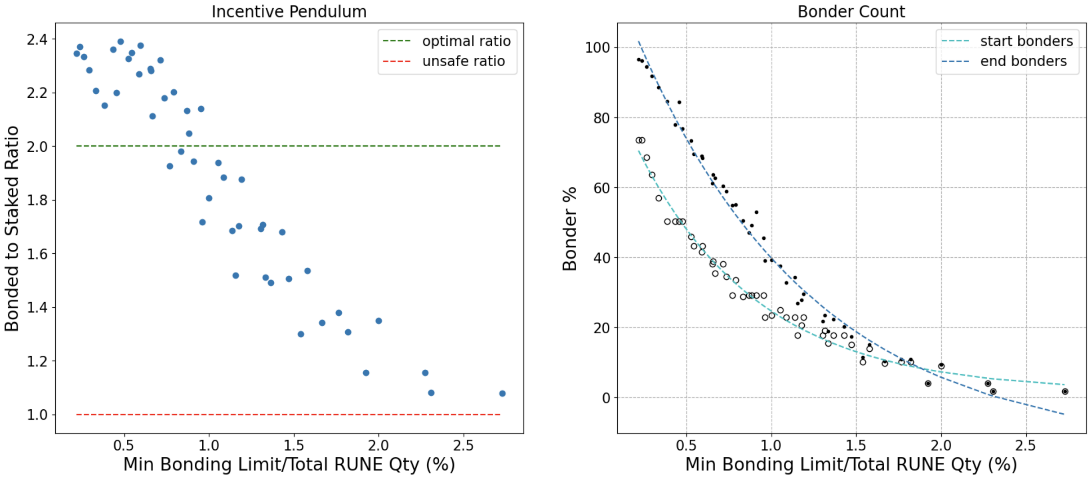


Figure 5: Minimum Bonding Market Share

the incentive pendulum response to the new bonders' staking vacancy. Because of this acceleration of bonding capital availability, it is important that q is closely monitored to ensure the network is in a stable state.

9 Incentive Pendulum

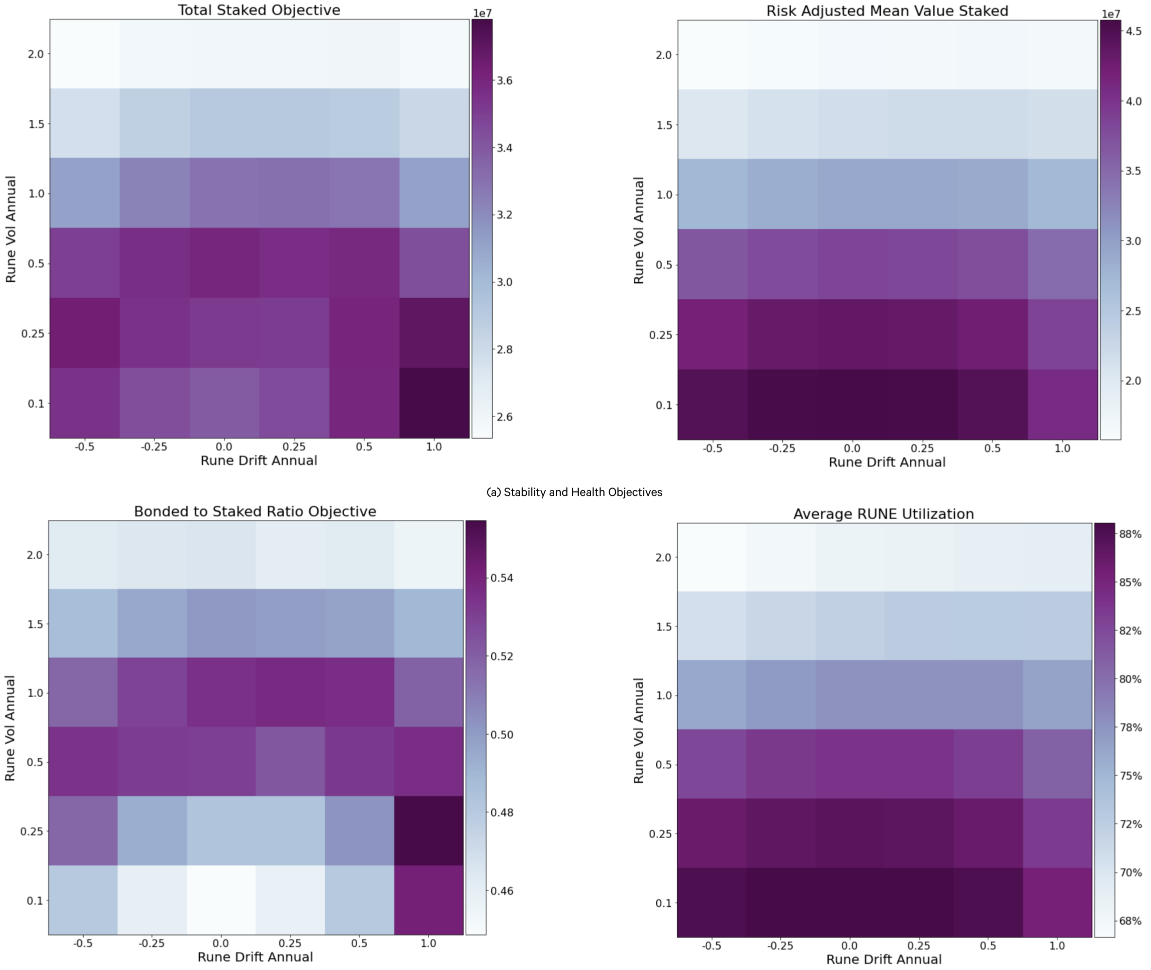
9.1 Price Trajectories

RUNE/USD annualized drift and volatility set the market context for the simulation, as alluded to in Sections 5.2 and 5.3. Since their variation affect nearly every measure of protocol success, we investigate by running a total of 1080 simulations across a search space of

$$\begin{aligned}&\text{RUNE Drift Annual (\%)} &&\in \{-50, -25, 0, 25, 50, 100\}\\ &\text{RUNE Volatility Annual (\%)} &&\in \{10, 25, 50, 100, 150, 200\}\end{aligned}$$

In the right plots of Figure 6 the risk adjusted mean total value staked (mean of total value staked at every time step minus the standard deviation) and RUNE utilization both have a strong inverse relationship with volatility and to a lesser extent unsigned drift as well. This is consistent with the specifications of liquidity providers acting more selective with staking in regimes of large price fluctuation or directional movement. In high vol and drift markets LPs are more prone to faster entry and exit to take advantage of increased fees without giving up too many holding gains as opposed to staking for long periods as a form of passive income in periods of low vol and drift. Faster liquidity turnover also affects the incentive pendulum, causing it to swing with larger magnitude which in turn may lead to more cases of bonders turning towards staking and displacing staking capital that is more content to sit unused now being less incentivized to provide to the system.

In the top left heatmap the total staked objective which is effectively a reweighting of the mean value staked according to the security scaling function from Section 5.1 is similar to the top right but more favorable towards higher vol cases and less favorable in the lowest vol scenarios. This is more readily apparent in the bottom left plot displaying the pure average objective which highlights bands of local optima around 75% annualized vol and also around 100% annualized drift. This touches upon the trade-off between security and efficiency where since the initialized total capital available for bonding comprises only 50% of all circulating RUNE, at optimal security 75% of RUNE is utilized whereas at full utilization the system is on the verge of incentivizing attacks. In low vol and drift since LPs have few risks to deter them from collecting staking returns the system tends towards less safety and in high vol and drift the risks become a significant deterrent that the system tends towards lower efficiency.



(a) Stability and Health Objectives

(b) Security and Efficiency Objectives

Figure 6: Price Trajectory Protocol Objectives

9.2 Fees and Costs

As introduced in Section 5.5 the liquidity fee scalar cap m is a big factor for LPs in deciding to stake in volatile periods. If it's too low then LPs won't stake due to impermanent loss but if it's too high bonders may decide to cannibalize some of the returns by switching to staking. From the bonder perspective the aggregate bonding cost of hardware, attack risk etc is an obvious factor in deciding to bond. To see the effects of these parameters

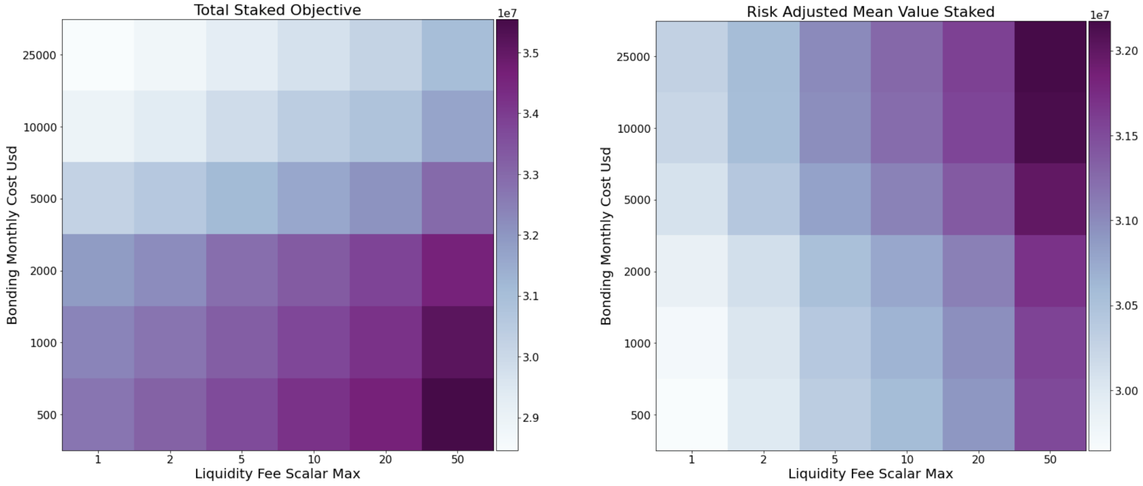


Figure 7: Fees and Costs Stability and Health Objectives

on agent decision making we run a total of 720 simulations across a search space of

$$\begin{aligned}
&\text{Liquidity Fee Scalar Max} && \in \{1, 2, 5, 10, 20, 50\} \\
&\text{Bonding Monthly Cost (USD) (\%)} && \in \{500, 1000, 2000, 5000, 10000, 25000\}
\end{aligned}$$

In the right heatmap of Figure 7 the risk adjusted mean value staked increases with both m and bonding costs. This is expected as LPs rush to stake if m is high to take advantage of periods of heightened returns, analogous to the profits of traditional market makers in volatile markets. As bonding costs rise staking becomes more appealing to bonders as well in their decision calculus. When reweighting the total value staked for the bonder objective in the left plot the total staked objective increases with m but decreases with bonding costs. The decrease can be attributed to the drop in system defenses when bonders switch to staking as capital previously deployed towards protecting the protocol now requires its own security.

9.3 Altruism

While bonder altruism φ is explicitly defined in Section 5.1, intuitively it acts as a robustness measure for the protocol. In times of systemic stress when short and long term economic incentives are oppositely aligned, φ represents the swing between taking an immediate loss to provide aid in THORChain's hour of need or dodging the hit to opportunistically return if the system survives. This is a common dilemma in many forms of investment, especially in high equity moments such as market crashes. For this reason we run a total of 1760 simulations across a search space of

$$\begin{aligned}
&\text{Bonder Altruism} && \in \{-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1\} \\
&\text{RUNE Number of Jumps} && \in \{2, 8\} \\
&\text{RUNE Jump Drift} && \in \{-0.1, 0.1\} \\
&\text{RUNE Jump Standard Deviation} && \in \{0.3, 0.6\}
\end{aligned}$$

where the number of jumps, jump drift and jump standard deviation correspond to λ, γ and δ in Equations 3 and 4.

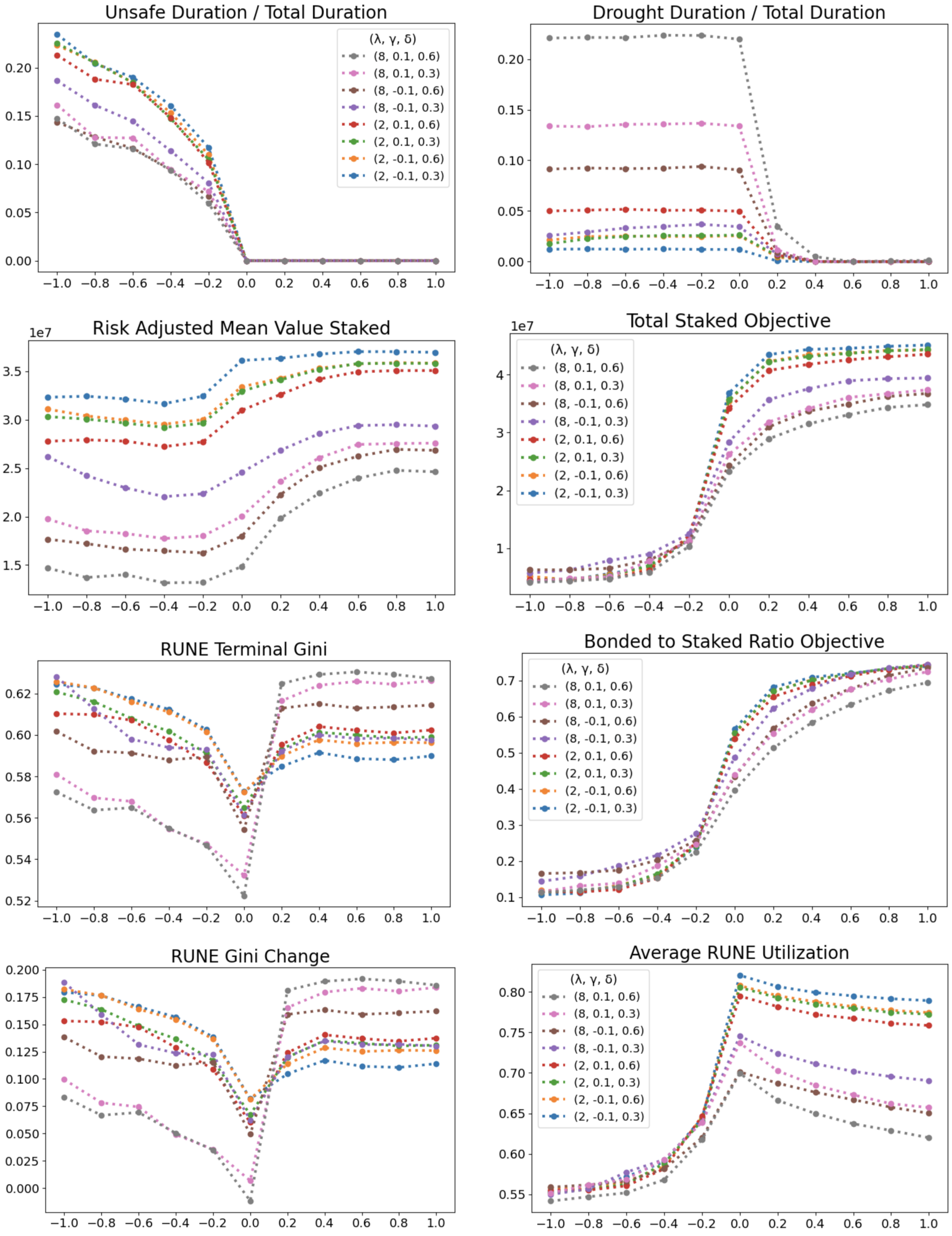


Figure 8: Altruism Protocol Objectives

In price jumps that could come from a Coinbase listing or a Black Thursday type event a common response is to withdraw capital to convert into a safe haven asset. This could lead to a shortage of either bonded or staked assets so we measure the duration of these as system failure conditions. Looking at Figure 8 the top row of plots chart how often the bonded to staked ratio is below 1 which is deemed unsafe and how often the total staked RUNE is near 0, defined as when the bonded to staked ratio is greater than 20 which is denoted as a drought.

It is important to note that changing the jump parameters have little impact on r_{bonding} from Equation 13 so the decision making of bonders is largely determined by altruism. For stakers, the jump parameters influence the estimated $r_{\text{RUNE/USD}}$ from Equation 11 over individual lookback windows where $r_{\text{RUNE/USD}} = 0$ maximizes the staking ROI while large magnitude $r_{\text{RUNE/USD}}$ disincentivizes liquidity providing. Jumps therefore correspond to less staking and increases in the number λ or the standard deviation δ lower the total staked capital. γ being positive also reduces liquidity since as a simulation default μ from Equation 3 is set to 0.2. In every plot in Figure 8 there are only one of two orderings, either in the direction of more staked capital or against, with λ being the most important factor, followed by γ and δ and we will view each graph in these contexts.

Referencing the legend all the plots share, the drought chart unsurprisingly corresponds to less staking. Droughts occur after directional moves large enough to cause stakers' ROI calculations to all turn negative, leading to complete withdrawal of their stakes. Having more jumps naturally increases the likelihood of large jumps as does doubling δ and positive drift jumps exacerbate the inherent upwards price momentum. Another observation is that even small amounts of altruism mitigate the issue with sharp reductions at $\varphi = 0.2$ and virtually nonexistent at $\varphi = 0.4$. This is encouraging for the protocol as the collective short term sacrifices on the part of the bonders needed to prevent a liquidity shortage is actually relatively small.

On the left plot the unsafe condition is breached more often in the inverse relationship of the droughts with increases in failure rate occurring with more staking. This can be attributed to negative altruism where adversarial bonders attempt to keep the protocol perpetually in either a bonding shortage or a staking shortage. Thus when the system is naturally under-staked due to the drought conditions agents with $\varphi = 0$ choose not to stake to help the system while agents with negative φ choose to bond to decrease the objective. On the other end where the system has a lot of staking liquidity, namely low λ, slightly negative γ and small δ as seen in the risk adjusted mean value staked plot in the second row of Figure 8, the disruptive bonders opt to devote their capital towards staking, plunging the protocol into a state of insecurity. It's important to note that for all cases of $\varphi \geq 0$ the system is never unsafe and that the failure rates decrease with weaker $\varphi < 0$, establishing that an agent would have to act substantially against their short term financial incentives in order to expose THORChain to attack.

In the middle rows, the risk adjusted mean value staked follows the positive staking trend with generally a modest increase in overall staking as φ changes from negative to positive. The total staked objective and the bonded to staked ratio objective also correspond to more staking but the magnitude of the effect is small. This is explained due to drought being a more prevalent issue with higher rates of occurrence while being weighted equally in the metric as insecurity. The more prominent relationship is the sigmoidal curve between φ and the objectives with inflection points around $\varphi = 0$. This is telling of the sharp changes in the success of the protocol on the direction of altruism.

In the bottom rows, the terminal RUNE Gini and the percentage change in Gini both rise with less staking. This is explained by the reduction in system income to stakers and LPs in this context which prevents them from gaining enough RUNE to run nodes and have access to bonding returns. A more interesting result is in the Gini absolute and percentage increases as φ leaves 0. This brings up a parallel between altruism and bonder cooperation where having the same φ values allows bonders to act closer to a collective and benefit the group as a whole. When $\varphi < 0$ the Gini metrics rise with the magnitude of altruism, indicating that undermining the system is a more disjointed effort before φ reaches -1. For $\varphi > 0$ even a small amount has a large impact with Gini being stable for $\varphi \geq 0.2$ highlighting the issue that even as the bonders work to help the system and improve all the other overall objectives this still leads to more income inequality and less decentralization.

As a final remark on Figure 8 the utilization understandably increases with more staking as capital is placed in the system instead of sitting in a wallet. In a similar manner to the Gini plots, utilization decreases as φ leaves 0. The drop is more steep for $\varphi < 0$ due to adversarial bonders sidelining their capital when the system is unsafe, neither bonding to help THORChain nor staking which adds marginal destabilization at the significant risk of asset loss from theft. This corresponds to the rise in unsafe duration in the top left failure plot. When $\varphi > 0$ the drop is more gradual but occurs due to increased bonder restraint in trying to maintain an optimal bonder objective instead of maximizing their short term gains by deploying all their capital. Thus the benefits of altruistic bonding behavior does come at the cost of some RUNE efficiency.

In Figure 9 we see altruism at work at an individual simulation level. The top charts display an example price trajectory for RUNE and ETH, in this particular instance RUNE has undergone significant appreciation through several upward price jumps. The charts underneath the price paths plot the total bonded and staked RUNE in the simulation run for $\varphi \in \{-1, 0, 1\}$.

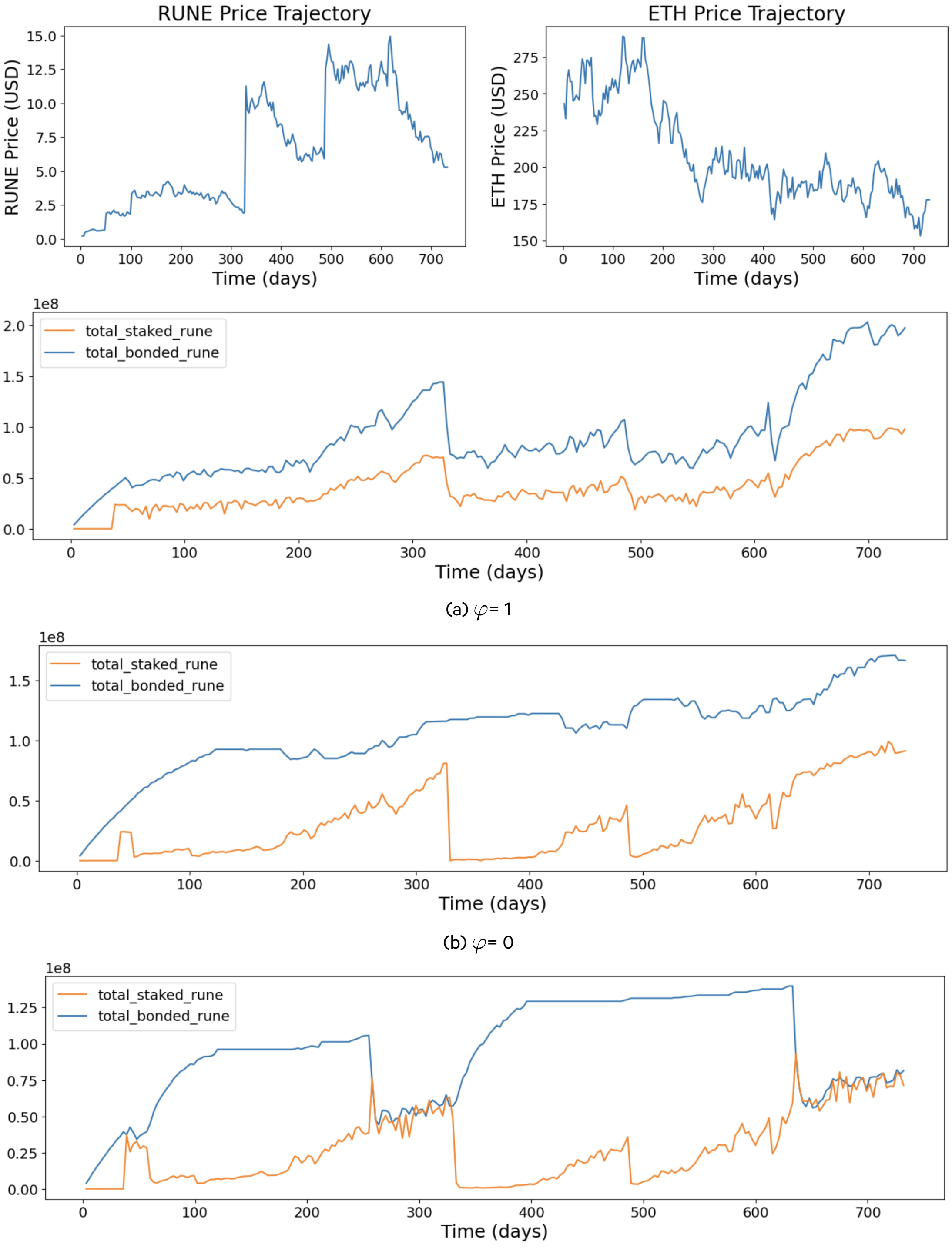
In the case of no altruism, at the first large price jump about 320 days into the simulation all the LPs have pulled their stake and the pools are dry. This happens again around the second large jump just before the 500 day mark as well. Here the system bonded capital isn't securing any assets and there are no transactions to validate either. An extended shutdown is detrimental to the protocol as traders turn elsewhere for liquidity. A few bonders are needed to take a projected staking loss to ensure the pools remain liquid but none of the myopic rational agents wants to give up their bonding rewards to take this leap of faith in this prisoner's dilemma.

In the case of positive altruism the bonders recognize the long term value of maintaining a healthy incentive ratio. The second plot shows the bonded and staked RUNE in virtual lockstep through the large price jumps where bonders opt to stake to replace the liquidity that has left the system until it returns. In the case of negative altruism bonders act in short term self interest and are Byzantine with regards to the longer term well being of THORChain. In this run they keep the system either severely underbonded or understaked. Around the 260 day mark the bonded to staked ratio hovers around 1 since enough RUNE is staked such that switching to staking lowers the objective without incurring too much immediate cost. When the price jump causes a liquidity drought the bonders actively bond more instead of staking until the pools recover around the 630 day mark at which the adversarial switch to staking occurs once more.

Part V

Conclusions

In this report we conducted an assessment of the effects of different market dynamics and bonding and staking behavior on the THORChain protocol via agent based simulation. We stress-tested the incentive pendulum under a wide range of protocol and model parameters to show that the existing choices lead to a stable and healthy system in a variety of market contexts. In particular, the results indicate robustness through a multitude of drift and volatility regimes with a sweet spot for all objectives when drift is between 50% and 100% annualized and volatility is around 50%. We also find the current distribution of the circulating RUNE supply of 160MM is suitable for initialization and the minimum bonding limit of 1MM keeps the bonded to staked ratio just above the optimal value of 2 with the caveat that they both need to be carefully monitored and may require future intervention. THORChain's incentive structure rewards slow but steady growth, which works well in maintaining stasis in the face of varying conditions up to a feasible level. Relaxing the assumptions of myopic rationality, we observe that the protocol is able to avoid potential death knells such as extended liquidity droughts or stretches of asset vulnerability with little to no amounts of bonder altruism. Avoiding potential downward spirals like bonder plutocracy could still however require a combination of governance and benevolence.



(a) φ= 1

(b) φ= 0

(c) φ= -1

Figure 9: Altruism in Price Jumps

Part VI

Appendix

10 Python Contracts

There are three contracts that, in addition to our agent based simulation platform, compose the simulation environment:

1. THORChain Contract: A modification of the Python implementation of THORChain found here fit towards our simulation backend. Handles all protocol logic (churning old nodes, managing the standby queue, allocating inflation and liquidity rewards, redistributing system income etc.) and executes agent transactions such as bonding, staking and swapping.

2. Exchange Contract: A centralized venue for agents to buy and sell tokens such as RUNE and ETH through a traditional order book. Simulates the price impact effects of trades by updating the token prices according to impact model parameters. Acts as an external price feed.

3. Portfolios Contract: A helper contract that tracks all the agent coin balances and enables transferring, minting and burning tokens.

11 References

[1] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. Preprint, 2019. Available at https://arxiv.org/abs/1911.03380.

[2] Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. Preprint, 2020. Available at https://arxiv.org/abs/2003.10001.

[3] Thorchain: Securing assets with incentives, May 2020.

[4] T. Chitra, M. Quaintance, S. Haber, and W. Martino. Agent-based simulations of blockchain protocols illustrated via kadena's chainweb. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 386–395, June 2019.

[5] Steve Yang, Mark Paddrik, Roy Hayes, Andrew Todd, Andrei Kirilenko, Peter Beling, and William Scherer. Behavior based learning in identifying high frequency trading strategies. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–8. IEEE, 2012.

[6] Grzegorz Halaj. Agent-based model of system-wide implications of funding risk. 2018.

[7] Anqi Liu, Mark Paddrik, Steve Y Yang, and Xingjia Zhang. Interbank contagion: An agent-based model approach to endogenously formed networks. *Journal of Banking & Finance*, 2017.

[8] John Geanakoplos, Robert Axtell, J Doyne Farmer, Peter Howitt, Benjamin Conlee, Jonathan Goldstein, Matthew Hendrey, Nathan M Palmer, and Chun-Yi Yang. Getting at systemic risk via an agent-based model of the housing market. *American Economic Review*, 102(3):53–58, 2012.

[9] Richard Bookstaber, Mark Paddrik, and Brian Tivnan. An agent-based model for financial vulnerability. *Journal of Economic Interaction and Coordination*, 13(2):433–466, 2018.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[11] Peter Nystrup, Stephen Boyd, Erik Lindström, and Henrik Madsen. Multi-period portfolio selection with drawdown control. *Annals of Operations Research*, 282(1-2):245–271, 2019.

[12] Chunsheng Zhou. A jump-diffusion approach to modeling credit risk and valuing defaultable securities. *Available at SSRN 39800*, 1997.

[13] Uniswap: A good deal for liquidity providers?, January 2019.

[14] Understanding uniswap returns, February 2019.

[15] Jan Hendrik Witte. Volatility harvesting: Extracting return from randomness. Preprint, 2015. Available at https://arxiv.org/abs/1508.05241.

[16] Joseph Clark. The replicating portfolio of a constant product market. *Available at SSRN 3550601*, 2020.

[17] Alex Evans. Liquidity provider returns in geometric mean markets. Preprint, 2020. Available at https://arxiv.org/abs/2006.08806.

12 Authors

Victor Xu is a Data Scientist at Gauntlet. Victor worked as a quantitative trader at Flow Traders. Prior to Flow Traders, Victor spent time at Citadel and Jane Street Capital. He studied Mathematics and Computer Science at Princeton University.

Hsien-Tang Kao is a Senior Data Scientist at Gauntlet. He has spent his career working on data science and data infrastructure roles at Sojern and Delta Electronics. He holds an M.S. in MechE with a concentration in Robotics from Carnegie Mellon University and a B.S. in Mathematics and MechE from National Cheng Kung University.

13 Acknowledgements

The authors would like to thank John-Paul Thorbjornsen, Kai Ansaari and Chad Barraford for helpful discussions. We would also like to thank the other members of Gauntlet team, in particular Rei Chiang for simulation design and best practices, Tarun Chitra for guidance and clarity, Lawrence May and Jonathan Reem for help with the simulation infrastructure, and John Morrow for his insights and editing.