THORChain: Securing Assets with Incentives

thorchain.org

Abstract. THORChain is a decentralised liquidity network to enable cross-chain exchange of assets in a permissionless way. The system seeks to do this using nothing other than the assets in its system, the information provided by connections with other blockchain networks and incentives by releasing its own network native asset (RUNE) to participants. The heart of THORChain are the Continuous Liquidity Pools (CLPs) that bind together the assets to be exchanged in a deterministic way and transmit the asset's purchasing power to the system. Liquidity pools not only solve the problem of asset exchange by allowing always-on incentivised liquidity, but also allow manipulation-resistant price feeds. By having price feeds internal to its system, THORChain is never concerned with dealing with external pricing sources prone to manipulation. Since the network's asset is both the settlement asset of all pools as well as being bonded by network nodes, THORChain is able to secure only the assets in its system and these assets are always fully-underwritten. The system is tightly-coupled; as assets enter the system via either being contributed as pool liquidity, or through the act of exchange, they necessarily transfer purchasing power to the network's native asset. As the collective value of all external assets secured by the network increases, the value of the network's native asset increases in proportion, ensuring constant security.

1. Introduction

For a system to secure assets that are external to it, it must ensure that the economic cost to attack the system must always exceed that value of assets it is securing. If there ever is a case where more can be gained from an attack than it costs; the assets are insecure. Proof-of-Authority networks (such as Blockstream-Liquid) keep assets safe by selecting for entities to put their corporate reputations at risk. The security of these networks are more difficult to assess, since actors are also assumed to be partially altruistic (eg, Blockstream), and reputation costs are subjective. Proof-of-Work (PoW) is not a viable option, because it would be very difficult to couple the value of assets secured with the cost of setting up hash-power to secure the network. Additionally, PoW compute cost lends itself to be <1% validation, with the other 99+% being devoted to work. This makes the system inefficient as a computing platform.

Thus THORChain is a Proof-of-Stake network, where the costs to attack are purely capital-based and a system of incentives can be designed to target the desired behaviour. Assuming that all participants are rational profit-seeking actors, the assumptions can be narrowed and the system's security can be objectively measured. To ensure assets are secure, the system needs the following:

1. Knowledge of the value of all the assets it is securing
2. Knowledge of the value of the assets providing security
3. A means to drive behaviour such that assets providing security are always more valuable than assets to be secured

To achieve this, the system uses Continuous Liquidity Pools to transmit the purchasing power of all assets on its network, and uses incentives to keep the amount of capital in its system optimal. Thus it is able to ensure that the value of assets bonded by nodes always exceed the value of assets held in its pools.

2. Definitions

Network Native Asset: The network's asset used as a bond for nodes and the settlement asset in each pool.
Bonded Capital: Capital placed as a "bond" by nodes to join as a participant. It is in the form of the network's native asset.
Pooled Capital: The capital placed in liquidity pools, where each pool is comprised of an equivalent amount of the external asset and the native asset.
Pooled Native Assets: Solely the native asset in the pools.
Pooled External Assets: Solely non-native assets in the pools.

3. Assumptions

There are three main assumptions upon which the network operates with:

1) There is always less than 1/3rd participation of irrational actors.
2) There is always less than 2/3rds participation by colluding actors.
3) The network asset is worthless if the network is attacked and assets stolen.

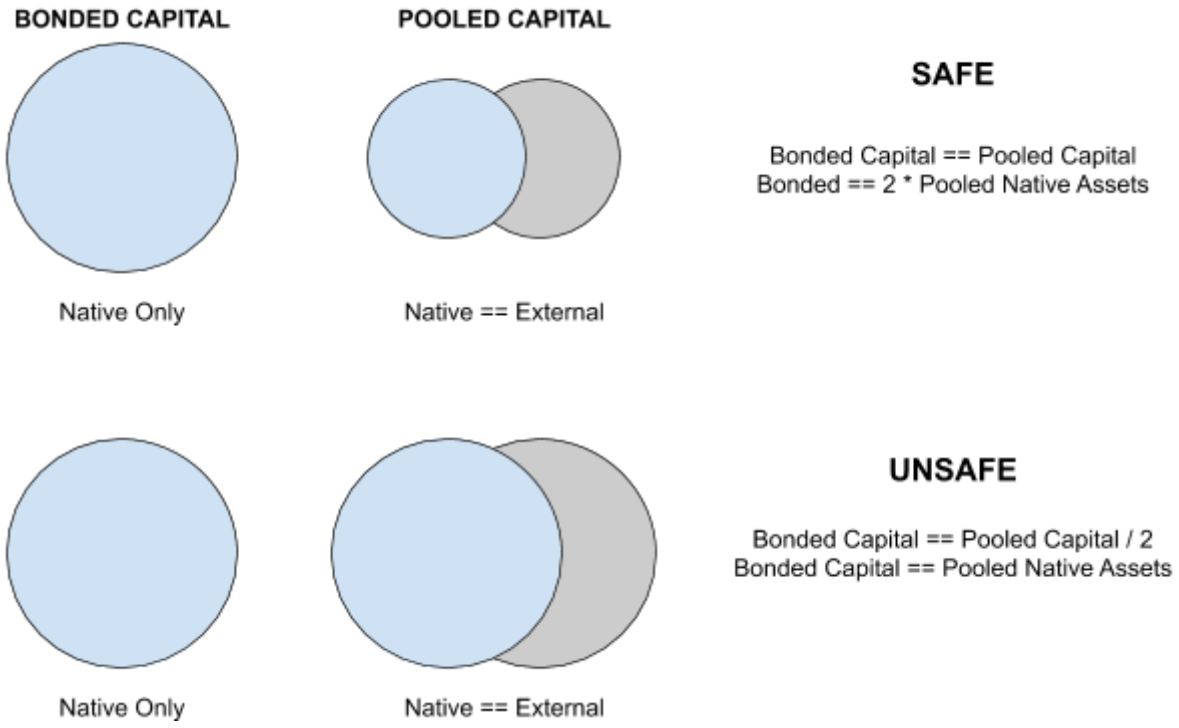
Irrational actors are not profit-seeking and simply wish to halt the network, preventing the redemption of any assets (bonded or staked). They are willing to bear economic cost with no gain. An irrational actor would be a state-organisation wishing to shut down the network. The network can be halted if blocks are not committed or TSS signing sessions are continually aborted.

Colluding actors are profit-seeking and have communication channels with each other. They contribute funds in order to become participants in the network. They are able to craft outbound transactions to spend any of the system's assets (bonded, staked or reserved) if they can coordinate such a transaction.

The network asset is liquid and has value simply because it is the settlement asset in all pools. If the pooled assets are stolen, then the network asset loses all liquidity. If the bonded or reserved assets are stolen, then they can not be redeemed anywhere and the market will be flooded with new supply, making it worthless. As an adjunct to (3), it is assumed there are insignificant secondary markets servicing the asset, primarily because the network asset is the most productive and liquid when it is in the system itself. Any secondary market trading of the asset actually erodes both the capital efficiency of the network, as well as the security assumptions of (3).

4. Optimal Capital Distribution

The network is categorically unsafe (under Assumption 3) when the bonded capital is less than the pooled native assets. At this point, the value of the bond in the native asset equals the value of the pooled external assets and all the nodes can steal more than what they bonded.



However, only 2/3rds of the nodes need to cooperate, which increases this threshold. If attacking nodes contributed 2/3rds of the bonded capital, and can steal all of the external assets, then the system is unsafe when less than 3/5ths (60%) of the network's assets is bonded:

$$AttackBond = ExternalAssets$$
$$2/3 * BondedCapital = 1.0 - BondedCapital$$
$$B * (2/3 + 3/3) = 1$$
$$B = 3/5$$

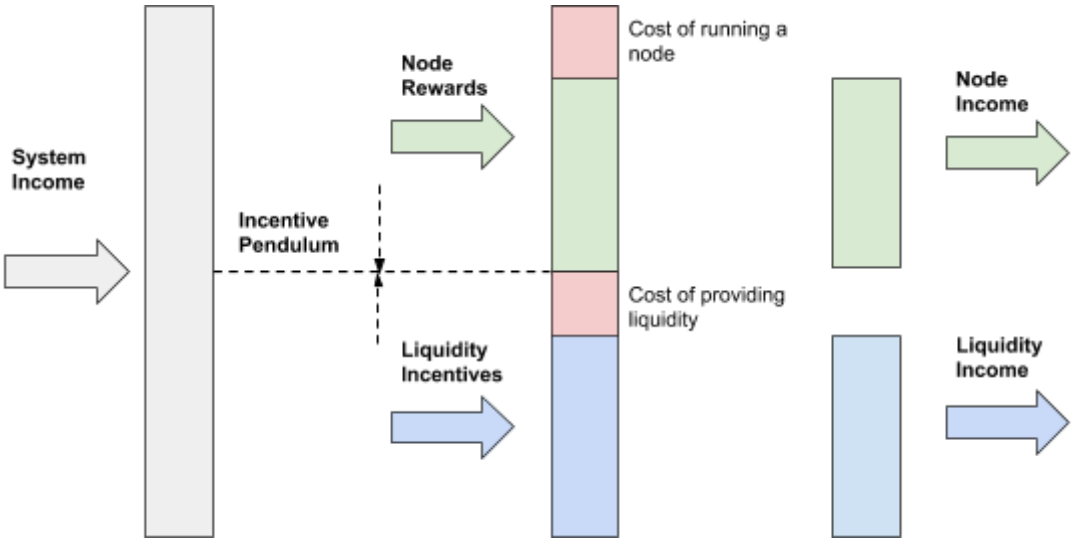
This threshold is actually slightly less, because the attackers can only access the assets in the Primary Vault, and 2/3rds of the Secondary Vaults. Since the Secondary Vaults only contain 50% of the assets at any given time, the attacking nodes can only access 5/6th of the total assets:

$$AttackBond = ReachableExternalAssets$$
$$2/3 * BondedCapital = 5/6 - BondedCapital$$
$$B * (2/3 + 3/3) = 5/6$$
$$B = 1/2$$

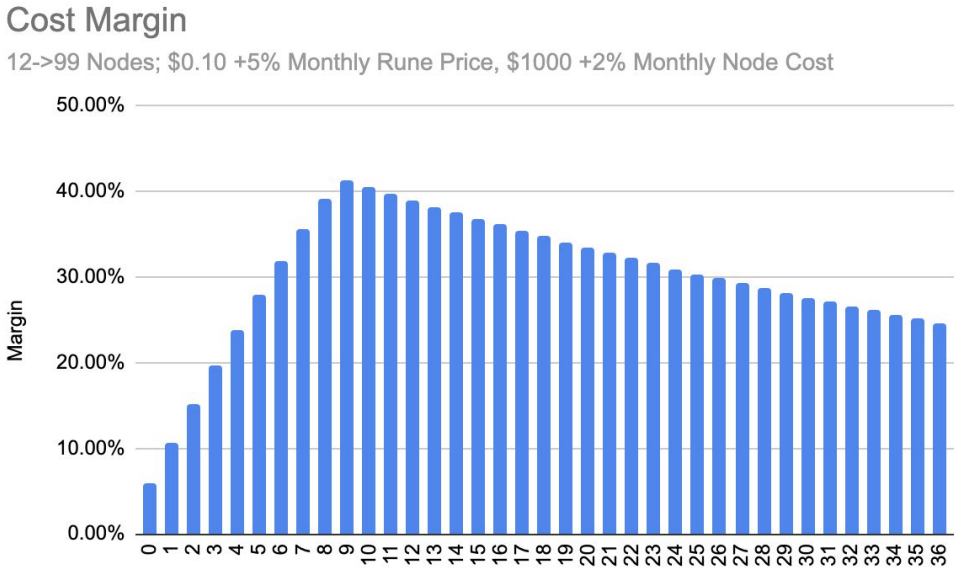
If there are any assets on secondary markets that could provide residual value to the asset and erode Assumption 3, then this in turn increases this threshold again. It is likely to be less than 10%. Thus the system should target the bonded capital to equal that of pooled capital, (where 2/3rds of the asset should be placed as a bond, and 1/3rd should be placed in the pools). This gives an adequate safety buffer to the safety stop of 50% bonded (+10%).

5. System Income

The system collects income from both liquidity fees and the block rewards, which are summed for each block. The System Income is divided between Node Rewards and Liquidity Rewards as a function of the Incentive Pendulum. Any costs (real or intangible) incurred by Node Operators or Liquidity Providers reduces their final income. The final income for both types of actors should be equal in order to drive their capital allocation such that 50% is allocated to bonding and 50% is allocated to pools (split between native and external). Assuming equal costs incurred, 1/2 of the System Income should go to Node Rewards and 1/2 should go to Liquidity Rewards in order to drive equivalent capital allocation.

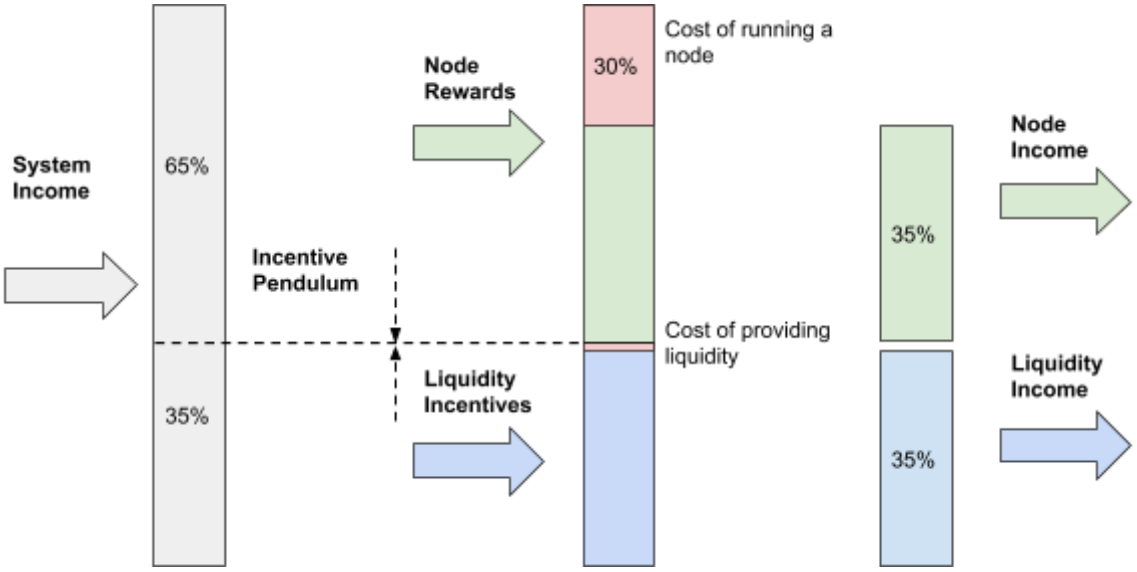


Nodes incur a cost to run infrastructure. This is likely to be around $1,000USD per month. Assuming the network's churn rate of 1 additional node every 3 days, the income for each node can be estimated. With some assumptions to the value of the RUNE asset, monthly costs, and the rate at which they increase, over 36 months the cost margin is found to not exceed 40%, reducing to below 30% long term:



Liquidity Providers incur very little real-world costs, however they do experience a phenomenon called Impermanent Loss, which is related purely to their entry and exit prices on capital. Assuming that the RUNE asset increases in the same nature as treated for nodes of a monthly increase of 10%, and that liquidity providers have a 12-month entry and exit, then their costs can be measured. Over 12 months, the cumulative gain in the RUNE asset is 150%, which translates to a 10% Impermanent Loss. This translates to a monthly intangible cost of just under 1 %.

This allows the costs for both Nodes and Liquidity Providers to be estimated and their income split to be revised:



Also to consider is that node operators have a significant barrier to having the technical skills in arranging and managing the infrastructure required to run a node. Additionally, the system is safer when it is slightly over-bonded, instead of being under-bonded.

Thus the final split given the assumptions above should target 2/3rds (67%) to Nodes and 1/3rd (33%) to Liquidity Providers when Bonded Capital is equal to Staked Capital. At this point, the system should be stable, since the return on capital invested would be roughly equal to both actor-types.

6. Incentive Pendulum

The system has a mechanism to continually re-target Optimal Capital Distribution. If more than optimal bond, the Bond Rewards reduce, and Liquidity Rewards increase. At the point where bonded assets are equal to staked native assets, all the rewards go to Bond Rewards and Liquidity Rewards are zero. This creates an opportunity cost for stakers and encourages them to leave the system, making it safe again. At the same time, huge incentives exist for liquidity providers to acquire additional native assets and bond as a node, restoring equilibrium.

If less than optimal bond, the Bond Rewards increase and Liquidity Rewards reduce. At the point where all the assets are bonded, then all rewards are paid to liquidity providers. This encourages some nodes to leave the system and become liquidity providers instead. This behaviour is driven by the following algorithm:

$$poolShareFactor = \frac{(b+s)}{(b-s)}$$

$$s = totalStaked,\ b = totalBonded$$

Scenario	Bonded Assets	Staked Native Assets	Share Factor	Bond Rewards	Liquidity Incentives
Inefficient	100%	0%	1	0%	100%
Over-bonded	75%	25%	2	50%	50%
Optimal	67%	33%	3	67%	33%
Under-bonded	60%	40%	5	80%	20%
Unsafe	50%	50%	-	100%	0%

The system should never reach the Unsafe or Inefficient conditions, and in reality is likely to be in an over-bonded state, especially if the barrier to entry for Node Operators is reduced, or node operational costs reduce.

7. Bond Rewards and Penalties

The system awards Bond Rewards, paid simply into a balance that is allocated to Nodes. When nodes join the system the block height they join at is recorded. When they leave, the total number of blocks they were active for is their total “Bond Units”. Every block the number of outstanding Bond Units increase by the number of active nodes. Thus at any point, the ownership of total allocated Bond Rewards for each node is simply the share of Bond Units redeemed. The total outstanding Bond Units is reduced when a Node leaves.

Nodes incur various penalties for negligent behaviour. Penalties are simply tracked by deducting Bond Units (and simultaneously reducing total outstanding Bond Units tracked). When the Node goes to leave if they have no Bond Units, they do not claim any Bond Rewards.

Nodes are also slashed if they steal assets from secondary vaults. Assuming the number of parties in a secondary vault is 3, with 2 required to sign, to ensure retribution of assets each party has to be slashed 1/2 the value of the stolen assets. Thus a colluding majority would steal the assets, but lose (2 * ½ = 1) of the value of them, for no gain. In total, 1.5x (3 * ½) of the assets are recovered. The additional 0.5x simply becomes a pool dividend.

8. Liquidity Rewards

The system awards Liquidity Rewards by simply paying out into pools. Importantly, the system defaults to amplifying existing fees collected. To do this, it sums up the total fees in each block, then calculates the existing share of fees per pool, then divides the total Liquidity Reward into each pool.

If there are no fees collected, the system simply pays out based on pool depth instead. This behaviour seeks to first amplify liquidity fees, before simply paying out dividends based on liquidity depth.

9. Gas and Mining Fees

The system subsidises all gas from the base asset pool for each chain. When the outgoing transaction is observed the final gas amount is reported on and deducted from the base asset pool. Normally this will be negligible. To offset the base asset being siphoned out from its pool over time, the system attempts to pay back the gas fee in the native asset to the base asset pool from System Income.

10. Network Fee

The system charges a fixed Network Fee on all transactions. This is a multiple of the trailing average of reported gas prices. This fulfils three objectives:

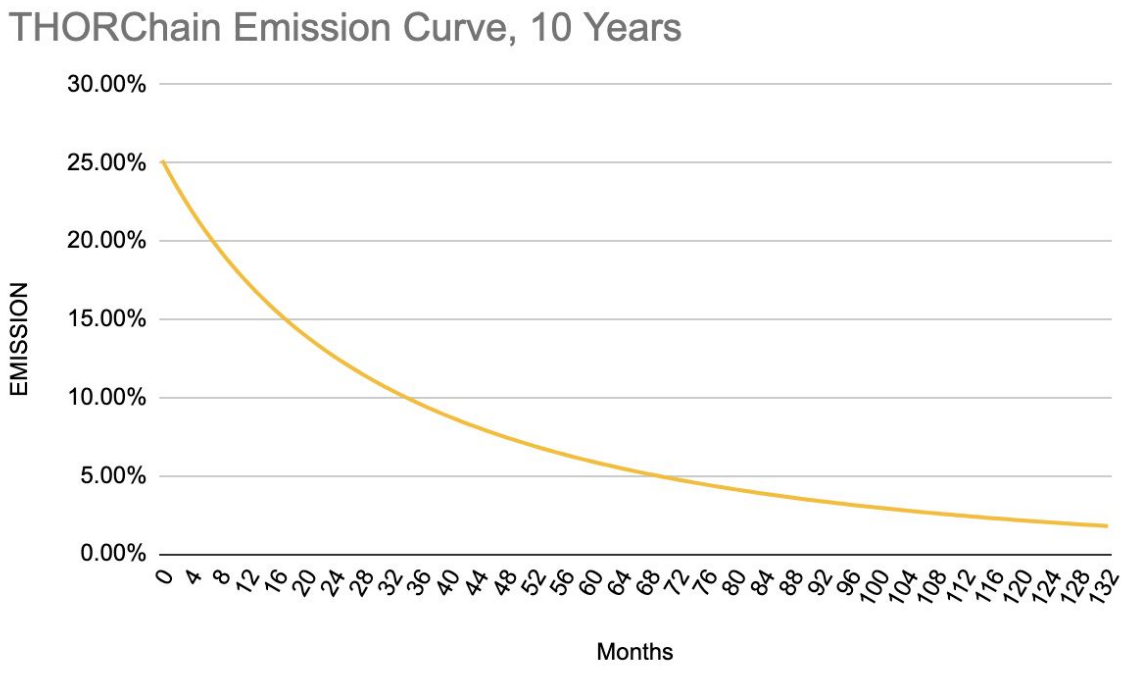
- To charge a fixed cost on all transactions to prevent dust attacks
- To accrue long-term income to the system during times of high economic activity
- To cover all dynamic gas fees charged and allow a fixed fee instead.

The network fee is always collected in the network asset. If the outgoing asset is not the network asset, the instantaneous equivalent is retained in the system. The network fee is always moved to the Protocol Reserve.

11. Emission Schedule

The system seeks to emit the Protocol Reserve to participants, targeting a 2% annual inflation after 10 years, emitting 1/6th the remaining Protocol Reserve each year.. To do this, it expects an average block time of 5 seconds, giving a total of 6,311,390 blocks per year. Thus at any given block, the equation for the block reward is:

$$blockReward = \frac{protocolReserve}{6 * blocksPerYear}$$



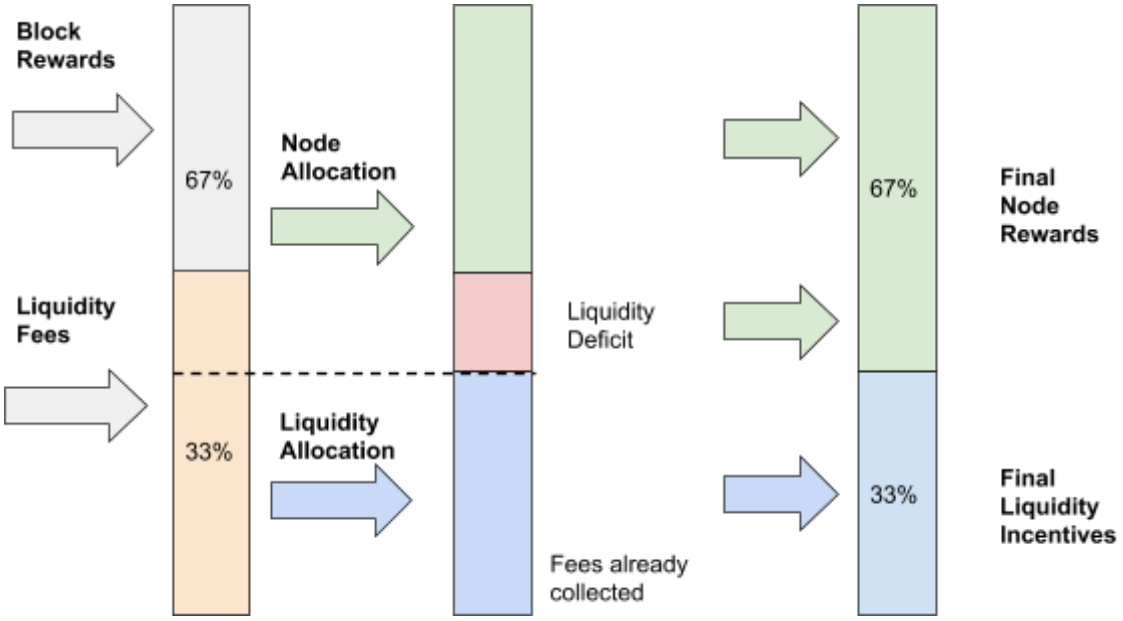
12. Liquidity Pools

The system uses Continuous Liquidity Pools to manage the exchange of assets. This allows a deterministic emission of tokens for each swap, as well as retaining liquidity fees. The following equations govern this process:

$$liqFee = \frac{x^2 Y}{(x+X)^2}$$

$$tokensEmitted = \frac{xYX}{(x+X)^2}$$

Liquidity fees can be also collected both ways, so liquidity fees collected in non-native assets (when buying an asset) are tracked by the instantaneous equivalent in the native asset. All liquidity fees are immediately collected as System Income and split in accordance with the Incentive Pendulum. Thus, a liquidity fee collected as the asset will remain in the pool, with an equivalent amount in the native asset being tracked each block. If the total liquidity fees for a block exceed the liquidity incentives owed, then the correct proportion is deducted as a liquidity deficit in the native asset, and paid to Nodes. In time, this deficit will grow as block rewards reduce and can be regarded as a tax on Liquidity Providers, such that they pay Nodes for security of assets.



13. Conclusion

THORChain is a system for managing the security of assets whilst allowing the predictable and incentivised exchange of them. It does this solely using the information available in its own state machine, information from the clients of connected blockchains, and incentives from emitting its own asset. It operates on very few assumptions, and has a very narrow attack surface. Assets can be regarded as always safe as long as the system has more Bonded Assets than Pooled Native Assets. This system uses an Incentive Pendulum to continually re-target Optimal Capital Distribution, which is an equivalent amount in Bonded as Pooled Capital.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, https://bitcoin.org/bitcoin.pdf, 2008

[2] E. Buchman, J. Kwon, Z. Milosevic, “The latest gossip on BFT consensus”, https://github.com/tendermint/spec/releases/download/v0.6/paper.pdf, 2018

[3] E. Buchman, J. Kwon, “Cosmos: A Network of Distributed Ledgers”, https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md, 2018

[4] G. Angeris, H. Kao, R. Chiang, C. Noyes, T. Chitra, “An Analysis of Uniswap Markets”, https://web.stanford.edu/~guillean/papers/uniswap_analysis.pdf, 2019

[5] R. Gennaro, S. Goldfeder, “Fast Multiparty Threshold ECDSA with Fast Trustless Setup”, https://eprint.iacr.org/2019/114.pdf, 2019