THORChain: A Decentralised Liquidity Network

www.thorchain.org

Abstract. A network is presented that can facilitate permissionless exchange of digital assets at fair market prices, resistant to attack. The network is serviced by a group of anonymous nodes who bond a certain amount of the network's asset and are continuously churned to ensure liveness and prevent capture. The nodes are compelled to participate in continuous key-generation ceremonies to create vaults with a virtual private key. They then monitor these vaults on external networks such as Bitcoin and Ethereum and make witness transactions into a replicated state machine when inbound transactions are observed. The state machine comes to consensus on these observations and applies exchange logic to them. Outgoing transactions are then delegated to smaller signing committees for expedited processing, which participate in multi-party computation to sign the transactions and broadcast back to the correct network. The network is most optimal when 2/3rds of the native network asset is bonded and 1/3rd is staked, at that point all assets can be underwritten. Protocol incentives aim to drive this behaviour. The network is governance-minimal, letting staked capital drive network direction.

1. Introduction

In the eleven years since Satoshi Nakamoto launched Bitcoin and gave the world its first and most important digital asset built on economics and computer science [1], a pareto distribution of other digital assets have emerged from the free market with various characteristics. However, users have never had the ability to exchange these assets using similar principles of trustlessness and decentralisation. Current digital asset exchanges carry the same design as traditional financial exchanges, being custodial, competitive and centralised. This results in numerous problems to do with security and liquidity.

What is missing is the ability to exchange digital assets in a manner that is non-custodial, connected and distributed. Such a network should have incentives for anyone to contribute to either security or liquidity, which should be tightly coupled so that the network is never unsafe. This network should be agnostic to other chains such that it can be connected to any, and must be able to scale sufficiently to handle connections to as many external networks as required. This network should be permissionless; anyone should be able to transact on it or contribute to it. This network should be resilient, being able to evade oppression and continually stay live despite adversity. This network should regularly churn its consensus set to prevent stagnation and capture.

This paper describes a network that has these desired characteristics and is able to solve the problems of security, liquidity and more. The technology to build such a network has only recently been made available, yet they are based on well-known fundamentals. The network is secured with very few assumptions, the most important being that a super-majority of network participants never collude. Implementations of this network should take precautions to prevent collusion prior to launching.

2. Overview

Current digital asset exchanges are comprised of infrastructure that as a minimum has the following:

1) A service that monitors incoming user deposits on different chains, crediting their accounts.
2) A trade engine that allows users to trade assets for each other typically using order books.
3) A service that allows users to withdraw assets on different chains.

While the system works well enough, it still suffers from the inherent weaknesses of centralisation, trust and inefficiency. Exchanges must be trusted not to debase user-deposits, else insolvency can easily occur. Operators must be trusted to handle hot and cold wallets carefully else irreversible theft of assets can happen. Since order books are an outdated liquidity model with numerous compromises, (such as having fixed sizes, requiring counter-orders to match with and having poor incentives for market-makers to place them), there is poor incentivisation for liquidity and order books are usually thin and easily manipulated.

A solution to the problem is to use a replicated state machine [2] [3] serviced by a significant number of nodes to observe user deposits on connected chains. The nodes come to consensus on the observations of these inbound transactions and then can process asset exchange logic using the state machine. Instead of order books, liquidity pools that function as automated market-makers allow precise, continuous and always-on liquidity [4]. Fees ensure market-makers are always paid. Assets inherit deterministic prices that can be utilised to signal purchasing power to the network. Outbound transactions are handled by the same nodes, immediately remitting the assets as soon as the transactions are signed via a threshold signature scheme. Security of pooled assets can be solved by coupling the system's security with its liquidity. Nodes bond capital into the system which prevents sybil attacks and ensures bonded capital is always a multiple of pooled capital. Assuming rational actors, the only security assumption is that the super-majority of nodes never collude with each other.

3. Nodes

Nodes are second-class citizens that service the network in return for incentives. They are not able to apply governance or have an opinion on the network's transactions or assets. Nodes should stay anonymous, never trying to coordinate, communicate or socially-signal. Nodes are penalised for not making witness transactions, committing blocks or interrupting key signing ceremonies. Assuming expected behaviour, nodes earn continuous incentives that become available to them when they leave the network.

Anyone can bond the required capital and apply to become a node by sending the capital into the system's primary vault. At this point they become whitelisted and are able to make transactions on the network. On a regular cycle, no more than a few days, multiple nodes are removed from the network and new ones that are bonding the highest amount of capital are added. This ensures the network stays live, is regularly recycled and prevents capture.

All nodes are eventually churned out, however any node can leave at any time by request which will be processed within a small number of hours. When a node leaves, or is naturally churned out, their bond and earned incentives will be paid back to them. At this point they can reallocate their capital and if they want, re-enter the system. Before they re-enter, they should

purge their old infrastructure and rebuild it from scratch. This ensures they are always running the latest software when they re-enter the system and this behaviour continually updates the network.

4. Chain Connections

Connections to chains are maintained using one-way state pegs, where instead of a token asset being pegged, only transactional state is synced to the network. Transactional state includes the asset payload, associated data (such as memo or payment ID), transaction ID and from/to addresses. Block-scanning logic is maintained that is unique to each connected chain. Only transactional state concerning the network's vaults is synced.

When a node observes a transaction concerning a vault they are monitoring, they compose a witness transaction and broadcast this into the network using their whitelisted address. This witness transaction contains the transactional state and is the same for every connected chain, no matter how that chain stores the original data. The network collects witness transactions from all active nodes and counts them as votes against unique transaction IDs. Once super-majority consensus is achieved, logic is applied against the transactional state. Nodes that don't make witness transactions, or fail to make correct transactions (as defined as the super-majority consensus), are penalised.

Nodes deal with network variability by maintaining a local cache of relevant transactions for each external chain. Nodes are able to identify activity such as double-spending and chain re-organisations and update the network if any part of a transaction changes. In this way the network is able to stay in sync and apply logic to counter any change in state that has been undone. If a connected chain suffers a contentious fork and not all nodes are on the same block height on the same chain, then it is safest for all nodes to simply stop observing that chain. This will invoke a chain-specific shut-down that returns all assets on that chain to users. Further edge cases generated from unreliable chains are generally not handled, so the network should avoid connecting to low-value, low-security chains.

5. Liquidity Model

Liquidity is achieved by bonding each supported asset to the network's native asset in pools, where the purchasing power of the asset is measured simply by the ratio of the depths of both assets. This ensures that all assets can be linked via the single native asset and allows the network to become aware of the prices of each asset at any time. Assets can be exchanged for each other by placing an asset into its pool and specifying the desired asset to exchange to, even if it is located in another pool.

The pools are virtual and a pool's assets don't reside on a single address, since the assets in a pool can be located in multiple vaults, across multiple chains. The network maintains awareness of the balances in each pool and where the assets are located. Since the network is aware of the instantaneous purchasing power of all supported assets, it is able to collect fees against any asset, such as subsidising gas to process outgoing transactions. Additionally, since the network's native asset is pooled against all supported assets, it is able to tightly couple liquidity with security. As the value of pooled assets grow, the value of the network's asset grows linearly, assuming prompt market arbitrage [4].

Liquidity providers add liquidity into pools by making special transactions into the network's vaults. The network observes the quantity of assets sent, and tracks pool ownership in its state. Pool ownership at the time of the transaction is given by the following equation:

$$Pool\ Ownership = \frac{((N+P)*(n*P+N*p))}{(4*N*P)}$$

$n = Native\ Asset\ Staked,\ N = Native\ Asset\ Balance$
$p = Pooled\ Asset\ Staked,\ P = Pooled\ Asset\ Balance$

Liquidity can be reclaimed by the liquidity provider making a special withdraw transaction into the network. The share of assets owed to them are remitted back immediately, including any liquidity fees or incentives earned during the period of time the provider had a share in the pool.

Users can exchange one asset for any other, by making a transaction into the network's vaults. They specify their desired asset, destination address and any price they wish to achieve. The network processes the request by calculating the final quantity of assets and sending to their desired address. The exchange of assets follows the following formula, known as the CLP formula. The CLP formula is derived from the constant-product market-making formula, but includes a slip-based fee, derived as follows:

$x = input,\ X = Input\ Asset\ Balance$
$y = output,\ Y = Output\ Asset\ Balance$
$P_0 = Start\ Price,\ P_1 = End\ Price$

The constant-product formula:

$$X * Y = K$$

Rearranging for y:

$$\frac{y}{Y} = \frac{x}{x+X} \rightarrow y = \frac{xY}{x+X}$$

Determining slip:

$$P_0 = \frac{X}{Y}, P_1 = \frac{X+x}{Y-y}$$

$$outputSlip = \frac{x/P_0 - t}{x/P_0} = 1 - \frac{Xy}{xY} = \frac{x}{x+X}$$

Determining the slip-based fee (slip multiplied to output):

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

Accounting for a slip-based fee in the final output (emission):

$$tokensEmitted = tokensOutputted - liqFee$$

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

Exchange of assets are charged a slip-based fee, whilst all outgoing transactions are charged an additional network fee. A slip-based fee ensures the final fee paid for is proportional to the demand of liquidity and that market participants consider the time imperative. Impatient trades pay higher fees, whilst patient trades give the market time to absorb any new information that could change market prices. A network fee ensures that the network can recoup processing costs for each outgoing transaction, as well as reimbursing for dynamic gas costs. Gas is paid for by the base

asset of each chain, taken from its pool. The network fee is a multiple of a trailing average gas fee for each network, and allows the fee to stabilise even though external network fees may have variable gas costs.

6. Vault Manager

The networks manages a number of vaults that hold assets associated with the pools, bonds and reserves. The vaults are considered non-custodial since all outgoing transactions are facilitated in accordance with the network's rule-set, can only be authorised by a valid transaction from the originator of the funds, and no node ever holds a key that can spend funds in isolation. Each vault is managed by a threshold signature scheme that requires a super-majority to participate in. It is not possible to track which nodes participated in any of the outgoing transactions from the point of view of an external observer.

Every time node membership changes vaults are re-generated and assets moved. For every chain there is a primary vault to receive incoming assets, and multiple secondary vaults that are used to remit outgoing assets. Each node is party to the primary vault and one of the secondary vaults.

Network participants query a node for the latest primary vault address, as well as its expiry time. Network participants should query multiple nodes to ensure they are not being eclipse attacked. For the case where there is multiple primary vaults, the system will return after query the vault with the latest expiry time.

If an outgoing transaction is to be processed, the network delegates one of the secondary vaults to process it. The secondary vault has a much smaller committee size and can process the transaction much faster. All nodes witness the finalised outgoing transaction in order for the accounting to be completed, including the final gas fee that was used. If the participants to the delegated vault do not process a finalised transaction, they are all penalised to the extent of the transaction value and the primary vault finalises the transaction instead.

The network monitors the balances of all secondary vaults and delegates outgoing transactions to vaults with sufficient balances. The network tops up secondary vaults regularly by sending assets from the primary vaults to the secondary vaults. At any stage half of all pool capital is kept in secondary vaults with the rest in the primary vault.

The primary vault is cycled when the network changes its active node membership, which is once every few days, or earlier to facilitate a voluntarily leaving node. Secondary vaults are only cycled when a node that is a participant to a secondary vault leaves. Cycling vaults regularly ensures the vaults have active signing participants.

7. Threshold Signature Scheme

The Threshold Signature Scheme (TSS) is the Genarro-Goldfeder 2018 [5] process, which allows efficient signing with no trusted dealer. There are two multi-party computation routines; key-generation and key-signing. The key-generation ceremony allows the nominated committee to construct the parameters for a new vault and the output is a public key from which the vault addresses for each chain is derived. Both secp256k1 and ed25519 chains are supported, and the vault address derivation is dependent on the chain.

When the network delegates an outgoing transaction to be signed from a certain vault public key, relevant signers recognise their participation, prepare a copy of the message to be signed from their local key-value storage, and enter a signing session. The key-signing ceremony begins when the required number of participants are present and allows a signature for an outgoing transaction to be generated. Once a valid signature is generated, all signers attempt to broadcast it to the relevant external network and one will be accepted. Since external networks handle sequence numbers, nonces or UTXOs differently, a chain-specific module is used to translate generic outgoing

transactions from the network into compatible transaction messages for each chain. Batch-signing is supported for networks that can handle it in order to increase signing efficiency.

The key-generation ceremony involves a communication round where each participant checks the validity of all other expected members, as prescribed by the system. Additionally to prevent spoofing, a commit-reveal scheme is used to ensure secret shares for each participant cannot be changed after the fact. During key-signing, all signers prepare and sign a locally-generated message only, so they also cannot be spoofed. If any of signers abort a key-generation or key-signing process which results in an attributable failure, all participants can make a blame transaction. If consensus is reached on who to blame, the blamed node is penalised and cycled to be churned out.

8. Network Security

The network has three important consensus bottlenecks. The first is collecting consensus on witness transactions, which ensure the state being pegged into the network is correct as per the super-majority's perspective. The second is arriving at consensus of the construction of block, which comprises of the state changes created from witness transactions. The final is arriving at consensus around the generation of valid threshold signatures for outgoing transactions. In all three cases a super-majority (67%) is required, but the network has no opinion on who should participate, as long as participants are bonded nodes.
By ensuring each node bonds a large quantity of the native asset, the system can prevent sybil-attacks. Since the native network asset can be acquired off public markets and has a fixed-supply, acquiring a majority of the asset becomes impossible due to run-away costs. The only attack vector thus becomes collusion between a super-majority of the nodes. If such a situation occurs, the entire network can be overcome, no matter the economic penalties.

Since the network holds security of assets external to it and allows nodes to leave voluntarily, there could be a case where the network node count drops below that required to maintain consensus. At this point the network invokes a global shutdown, where all assets are returned to original owners. This ensures that asset security is preserved during the entire lifecycle of the network. Since the native asset is accounted for on a separate chain, it means the network can be restarted post a global shutdown.

9. Scaling

The primary limit to how large committee sizes can be is that of TSS key-generation speeds. The network grows its TSS committee sizes for its primary vault as new nodes join. At some point, a new TSS key-generation ceremony will hit a pre-designated time-limit. The committee size at this point is the limit for the primary vault, and the network then chooses to shard the primary vault into two partitions, and continue growing them until they time-out again. This is continued until the network hits either the block consensus limit or the economic limit.
Since the network uses tendermint consensus, there is a block consensus limit. Large committee sizes with inefficient connections with each other result in block timeouts, increased block times and poor network performance. Increased block-times reduce the block-based revenue for nodes and should reduce node count until back below an equilibrium. The network targets block-production speeds of around five seconds. With a minimum bond limit, as well as a fixed-supply asset, there is an absolute maximum of the number of nodes that can participate, known as the economic limit.

The network can scale to handle a significant number of connected chains, and a practically unlimited number of assets. Connected chains have a resource burden on the network, but this can

be alleviated since light nodes that have been fast-synced are more than sufficient to function as chain connections. There is an asset listing cycle of a few days which ensures that only in-demand assets are listed and throttles how quickly new assets are added. Assets are listed in order of how much capital is staked towards them.

10. Network Economics

The system must safely secure external assets, which all have free-floating market prices. Since the network's asset is used as security, there must be a mechanism to couple the value of the native network asset with that of the assets it is securing to ensure the network is always safe. If there ever is a case that the capital required to bond is less than what can be gained by a majority-collusion theft from the network, then the network is insecure.
Since staked capital is held in pools where the purchasing power of the native network asset is always paired to that of the pooled asset, a leveraged coupling in value can be achieved. As long as bonded capital exceeds double that of staked capital, then all staked capital can be underwritten. Additionally, since the native asset is worthless outside of a functioning network the value of bonded capital must simply exceed the value of non-native staked capital. Therefore the system is designed to target a bonding rate of 67%, with a staking rate of 33%, by emitting incentives to nodes and liquidity providers in this proportion. At this point, the system is both efficient and safe. An incentive pendulum is added to shift incentives to the side that is not optimal, such that when bonded capital equals staked capital, staking incentives are zero, and when staked capital is zero, staking incentives are at their maximum. The incentive pendulum is driven by the following formula, which seeks to determine the share of rewards to pay stakers at any given capital allocation:

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

Since nodes and their bonded capital are regularly churned, it is desirable to have a consistent bonding rate such that the security characteristics of the network do not change when nodes are removed. Additionally, a consistent bonding rate creates a flatter distribution of assets, which raises the median bonding rates. As such, all nodes are paid an equal share, no matter the amount of bonded capital. To ensure continual consideration to the allocation of capital, nodes have their bond fully returned to them once churned out. Whilst they are bonding, they earn bond rewards emitted each block to all nodes. Nodes incur penalties from their accumulated bond rewards in order to drive them to highly-reliable and consistent network service. If they fail to make witness transactions on incoming transactions they are penalised, which ensures reliable witnessing. If nodes abort a key-generation or key-signing ceremony, they are also penalised since the network loses availability. If a secondary vault fails to process an outgoing transaction then the system could suffer an unintentional duplicate spend, so the at-fault nodes are fined the full amount of the transaction from their bond instead.

Liquidity providers are paid incentives for providing pool capital. Incentives are derived from either the value of the slip-based fee (if a block contains trades) or the depth of assets in each pool if no trades are processed in a block. Liquidity incentives are always 1/3rd of System Income, which is the value of all collected fees in a block, in addition to the block emission. If the slip-based fees exceed the 1/3rd threshold already, the excess is deducted from the pools in the native asset, and awarded to nodes as a bond reward. This allows the system to transition to a fee-only incentive structure when emission is negligible.

Since the network fee is always collected against every outgoing transaction and placed back into the protocol reserve, the system can build up long-term revenue during periods of high

economic activity. The emission curve of the native asset seeks to begin with 25% annual emission and reduce to target 2% after 10 years. The native asset is secured by the protocol in a reserve held in the primary vault. The network emissions may never run out if there is economic activity on the network.

10. Governance

The network follows a governance-minimal approach to prevent social-signalling amongst nodes and encourages psuedo-anonymity where possible. There is no supported ability for node operators to coordinate or communicate with each other. Users signal via capital allocation in the network and developers respond to network needs by observing capital allocation and delivering. Nodes select and run backwards-compatible software, but cannot run software that is lower in version than what the nodes in the network are already running. Due to high churn rates, software versions will continue to ratchet up.

Users signal to list new assets via making staking transactions with the asset in the transaction memo. If the asset is on a supported chain but the symbol is not recognised, the network places it in a bootstrap mode with swapping disabled. Every few days the network activates the deepest bootstrapping pool for swaps and receiving liquidity incentives. In this way the staked capital drives which asset is listed and in which order. Any asset can be delisted by all liquidity providers withdrawing their assets from the pool. If it is delisted, it must follow the listing process again to be added.

Developers can respond to the needs of the community by submitting new software for testing and validation against the primary binary. Due to the modular structure of the binary, building connections against new chains should be trivial. Node operators will update to the latest software and due to high churn rates, the network will be able to support new chains relatively quickly. Once a super-majority of the nodes are witnessing a new chain, the chain is added.

In the same way, chains are delisted by nodes ceasing to witness an existing chain. Once a super-majority are no longer witnessing a chain, a chain-specific shutdown process is initiated with all assets associated with the chain being returned to members.

11. Protocol Upgrades

Node operators can upgrade the application logic, schema or even the network itself via the protocol's upgrade system. Both the application logic and its schema is versioned and each node declares the specific version they operate on when they enter the system. Due to high churn rates, the network continually upgrades by selecting for the lowest common denominator of the super-majority. If a schema or logic rollover is detected upon a churn event, the very next block is produced against new logic and data is saved against the new schema.

When the network needs to be upgraded, a block height on the old chain is earmarked to result in a chain-halt, followed by a genesis import and chain-start of a new chain with upgraded network. The design goal of all protocol upgrades is to complete it with minimal coordination, but with fall-back safety and backwards compatibility.

12. Conclusion

A network that facilitates cross-chain liquidity is presented. The network is chain-agnostic and can be connected with most other chains. The network uses economic incentives to ensure safety and the validator-set is churned regularly to ensure liveness at all times. Assets are pooled such that any one asset can be swapped for another with deterministic liquidity and fees. Pools of liquidity

function as large bounties for public market participants to engage with to correct to fair market pricing at all times.

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