HyperNova: Efficient, Trustless Cross-chain Solution

Version 1.0

Supra Research

Abstract

Blockchain bridges enable the transfer of assets and data from one ledger to another. Traditional decentralized bridges typically generate a multi-sig or threshold signature on events of one chain to transfer assets and information to another. These bridge networks introduce new security assumptions typically in the form of an honest majority requirement among the bridge nodes. However, since these networks attract and settle such large volumes of cross-chain transfers, they are quickly seen as honeypots for targeted attacks. It is ideal to realize solutions that do not introduce new security requirements beyond the chains' native security assumptions.

Trustless bridges or Relay bridges realize this ideal and preserve the base-layer native security assumptions of the blockchains themselves, avoiding the introduction of any new security requirements. HyperNova is Supra's trustless bridge solution for the scenarios in which the destination chain can verify the correctness of events of the source chain. To verify the correctness of events of the source chain, the destination chain maintains updated auxiliary information, for example, the validator set in PoS blockchains. The safety of HyperNova and such solutions is guaranteed even when all Bridge Relay nodes are malicious, meaning bridge nodes cannot tamper with Bridge requests, nor can they spoof Bridge-requests to mint illegitimate tokens. For liveness, it requires at least one honest node. Hence the realized Trustless bridge HyperNova is secure, has low latency, provides *revertable* Bridge requests as a feature, and does not impose any rate-limiting on the amount of assets that can be transferred between chains.

1 Introduction

Blockchains have introduced "decentralized trust" across many services. As the rise of the adoption of Decentralized Finance (DeFi) protocols demonstrates, finance services have been the "killer application" of blockchains. DeFi provides alternative financial rails through automated and distributed-protocol driven, cryptography-enabled trustless services. This starkly contrasts the status quo of centralized, human-driven services, requiring too much trust. As per DefiLlama as of 7th September 2023, around USD 37 billion in total value locked (TVL) has been secured on all DeFi protocols across all blockchains. Though DeFi has been the main driver, the concept of decentralized trust offered by blockchains is being adopted for other applications and domains.

By design, most blockchain capabilities are decided by the capabilities of their validators. A blockchain's validators are the decision-makers in ordering blockchain transactions and forming the source of ground truth for the blockchain's state. They validate which users control what assets on their blockchain.

Each blockchain offers unique qualities, and many offer varied and distinctive services. As the circulation of assets and value is the economy's basis, financial assets naturally need to be bridged across blockchains. It's becoming ever clearer that DeFi is headed for a multi-chain future. Hence, we see many users using multiple services – sometimes the same, sometimes different – across multiple ecosystems. Blockchain interoperability, mainly for transferring assets and information from one chain to another, has thus become a necessity – and has naturally garnered the focus of researchers and builders alike.

The challenge of facilitating blockchain interoperability has led to the development

of blockchain bridges. Most bridges are currently designed as what are called multi-sig bridges [18, 21–23]. These bridges generally consist of a set of staked bridge nodes, each of which individually signs the events happening on a source chain (e.g., locked funds in the source chain's currency), aggregates the signed events and relays them with the signatures/seals of their agreement – a "multi-sig" – to a destination chain. This facilitates a corresponding action (e.g., the release of funds in the destination chain's native asset or currency) on the destination chain. As per DefiLlama on 7th September 2023, around USD 173 million of value is bridged in the preceding 24 hours, demonstrating the demand and utility of cross-chain bridges.

Generally, the coefficient of "decentralized-trust" is a function of the number of independent validators in the network and/or the value of the stake involved. The number of validators and the staked value of multi-sig bridges are generally much lower than the stake value or the number of validators of Layer 1 blockchains. Typically, "a chain is only as strong as its weakest link". In the long path of the circulation of assets, multi-sig bridges have shown themselves to be the weakest link as their coefficient of "decentralized trust" is generally lower. Consequently, bridges have become the target of both large and small-scale attacks. In fact, an August 2022 report from Chainanalysis estimated that 2 billion has been stolen from bridge hacks alone [5]. Therefore, to be effective and safe, bridges must NOT dilute the "decentralized trust" of the Layer 1 blockchains they transfer assets between. Hence, the security of bridge protocols is fundamental and paramount in establishing confidence among dApp developers to settle high volume, cross-chain transfers. Regular hacks also hinder blockchain adoption as end users lose confidence despite seeing value in these services.

Supra also offers a multi-sig bridge platform called HyperLoop. As far as we know, to date, we are the first to offer a multi-sig bridge that is analyzed using game-theory and is proven to be secure. The highlights of our design are the requirement of an honest simple majority instead of an honest supermajority amongst bridge nodes, a sliding window mechanism of request-rate-limiting, an incentivized set of whistle-blowers for detecting collusion, highly scalable and gas-efficient batching mechanism, and insurance coverage for any losses in the rare case of a breach.

In this paper, we explore a bridge solution that can preserve the security guarantees of the chains being bridged without introducing any new security requirements. We found that the optimal direction is to relay the events of a source chain and explicitly verify the validity of these events on the destination chain. This is in stark contrast to using a 'multi-sig' majority protocol for the conventional bridges. This entails verifying the consensus of the source chain on the destination chain. The nodes that relay events form the "bridge." We call our Relay bridge **HyperNova**. The Relay bridge design is depicted in Figure 1.

This change of bridge design from Majority Agreement to Relay removes the impediment of the aforementioned degradation of decentralized trust. As the validity of events of the source chain is verified independently on the destination chain, no relay bridge node can subsequently tamper with the information related to events on one chain while relaying, as otherwise, it will be caught in the verification phase of the L1 consensus and disregarded. So, attacks that give false information to release or mint assets on the destination chain without locking the corresponding assets on the source chain are not possible.

In the Agreement bridge design, we trust that a simple majority of bridge nodes always remain honest, whereas, in the Relay model, we need NOT trust any intermediary bridge node for the correctness of the relayed information. In the Relay model, we only require that relevant events on the source chains are not missed, censored or delayed in relaying them to the destination chain. Hence, the requirement of a simple majority of honest nodes in the

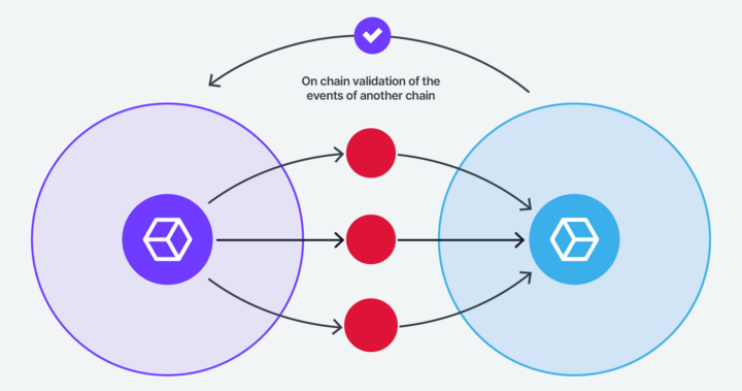

Figure 1 Relay Bridge relaying source chain information to the destination chain

Agreement model is reduced to a single honest node in HyperNova's Relay model.

The rest of the paper is organized as follows. We discuss related works in Section 2. Next, we study a concrete case of Ethererum [26] and describe the HyperNova design for bridging Supra and Ethereum trustlessly in Section 3. We then state the correctness properties in Section 4, present the general HyperNova protocol in Section 5 and show that HyperNova satisfy the correctness properties. We provide a short discussion about relevant nuances of this bridge design in Section 6. We finally conclude the article in Section 7.

2 Related Work

We omit comparing with multi-sig Bridges such as Axelar [21], Layer0 [22], Zetachain [18] etc, as they are already discussed in Section 1.

We understand that zero-knowledge-based bridging approaches [16, 19, 25] also yield a Relay bridge design, but the proof generation times of the current state-of-art approaches are still high compared to the other approaches. We understand that the zero-knowledge-proof space is constantly evolving and improving, and we will continue to watch and investigate solutions in this space.

Rollup bridges. Rollup bridges like Arbitrum bridge [4] connect specific chains like their base Layer 1 chain (e.g., Etheruem) to the rolled-up chain (e.g., Arbitrum). The execution security of the Layer 2 chain depends on the fraud-proof challenge period in case of optimistic rollups and the validation of Zero Knowledge proofs in case of Zero Knowledge based rollups. The data availability is dependent on the security of the Layer 1 chain. These bridges are also Relay bridges similar to HyperNova but are instantiated for specific chains.

As HyperNova is a pairwise bridge solution, it is appropriate to compare it against other pairwise bridges and not against interoperability-focused systems like Cosmos [6] and Polkadot [11] directly. So, we compare only the pairwise subchains of these systems or communication between a subchain and an external chain. A comparison of these interoperability-focused systems is scoped through our IntraLayer [15]. Very briefly, Supra's IntraLayer approach positions Supra at the center and connects to external chains using HyperNova and HyperLoop bridge solutions.

Cosmos

Cosmos IBC [6] is of hub-and-spoke architecture, with Cosmos Hub in the center and zones as the spokes. All the zones are independent and run Tendermint [17] consensus protocol and communicate with other zones via Cosmos Hub using Inter Blockchain Protocol (IBC). The nodes of the zones also run the Tendermint light clients of the Cosmos Hub so that all the communication is bridged via the Cosmos Hub. The Hub does not host user transactions. The communication between zones works on the same principles of adding no extra trust assumptions on top of the consensus mechanisms of the zones being bridged. With all zones (including the Hub) running the IBC client, they all know the Merkle root of the state of all the other zones and can verify the events of the different zones. The communication between Cosmos zones is the same as HyperNova in principle, but they are slow owing to an additional hop via the consensus on the Consensus Hub.

Since Tendermint light clients cannot be run on the validators of external (to Cosmos) chains like Ethereum, it requires another network, a specialized Cosmos zone called *peg zone*, to bridge. HyperNova is similar in principle to IBC when IBC operates inside Cosmos system, but it differs completely when bridging to external chains.

Polkadot

Polkdot [11] is also of the hub-and-spoke model with a Relay chain at the Hub. The Relay chain does not host any user transactions. Only the parachains (spokes) contain the user transactions. The relay chain maintains the state and the corresponding Merkle root of all the parachains. It also provides finality to the blocks of the parachains.

The communication between parachains happen only through the Relay chain. A parachain can know the Merkle root of the state of another parachain through the Relay chain. Hence the events of one parachain can be verified on another. The bridging between parachains is similar to HyperNova, that is, Relay Bridging. However, the security and the finality of the parachains are entirely dependent on the security and finality of the Relay chain, unlike HyperNova.

Consider the communication between an external chain and a parachain. Snowbridge connects BridgeHub parachain to Ethereum. Through Relay chain, any other parachain can connect to Bridgehub and thus connect to Ethereum. Like HyperNova, they also use Ethereum's sync committee to validate Ethereum's state. And on Ethereum, they use Polkdot's BEEFY client to verify Polkadot's state. HyperNova and Snowbridge work on the same principles of trustless bridging.

3 HyperNova: Supra $\longleftrightarrow$ Ethereum

We illustrate the HyperNova design by describing this Trustless Bridging from Supra to Ethereum and vice-versa.

Ethereum Consensus

Ethereum consensus [9] is a combination of Casper the Friendly Finality Gadget (Casper-FFG [24] and the LMD-GHOST fork choice algorithm. Ethereum has a large set of active validators (more than 770K as of 7th September 2023). The set of consensus validators are fixed during an *epoch*, and can only be changed at epoch boundaries. An epoch has 32 *slots*, spaced 12 seconds apart. All the validators are equally distributed and randomly assigned (and made known 2 epochs in advance) so that every validator is expected to attest (vote by

signing) precisely one block every epoch. As a consensus proof, every block carries a *multi-sig* – an aggregation of these attestations and a *bitmap* representing the participating validators.

Verifying a block's consensus entails aggregating the public keys of those validators which participated in the attestation of that block and then validating the aggregate signature using an aggregate public key. Since a large number of public keys must be aggregated, this mechanism requires long verification times and is inefficient in terms of gas consumption on the destination chain. This would not be a feasible approach for a bridge mechanism.

This motivated the introduction of a secondary consensus step using a *Sync-committee* in the Altair Fork [7] of Ethereum. A Sync-committee is a sub-committee of a fixed size of 512 nodes drawn randomly from the full validator set of Ethereum and is updated only once every 27 hours. The blocks of Ethereum are attested by Sync-committee validators on top of the regular attestation by the full set of validators. Accordingly, a Sync-committee attestation offers a cheaper way of verifying Ethereum consensus on other chains, thus facilitating a Relay bridging model (like HyperNova).

Is the Sync-Committee based Consensus Secure?

Collusion attack. The Sync committee consensus has been criticized [2,20] mainly for NOT having slashing mechanism as part of its protocol. The main argument is that a dishonest Sync-committee can get away with a collusion attack targeting a specific trustless bridge. Note that the Sync committee based consensus is an additional consensus. So even if all of the Sync committee is dishonest, they cannot produce a fraud Ethereum blockchain as long as the primary consensus of Ethereum is correct. However, a dishonest and colluding Sync-committee can trick a blockchain that verifies only the Sync committee based consensus but not the primary consensus by providing an incorrect Ethereum block. This attack requires the colluding Sync committee to also collude with a Relay node to forward the incorrect block to the chain.

This attack is specific to Ethereum as it has an additional Sync committee based consensus mechanism. For the PoS chains with only one consensus mechanism like Aptos [3] and Sui [14] unless the chain itself has been attacked, the above scenario is not possible. In that case no guarantees can be given by any bridge. We assume that such scenarios leading to compromised blockchains to be not possible in this paper.

Probability of having a dishonest Sync-committee. Studies from Succint [13], Snowfork [12], and T3rn [8] show that the probabilities of such dishonest Sync-committee formation to be extremely low. For instance, a trustless bridge could require that more than 90% of the Sync-committee validators have signed off on a block to be considered valid. Then, even assuming that $\frac{1}{3}$ of the full validators of Ethereum are dishonest, to begin with, the chance of having a dishonest Sync-committee (not having at least 10% honest Sync-committee validators) is once in 10^{31} years. These studies also remark that, apart from protocol-based slashing, there are other practical security mechanisms, owing to practical pseudonymity and reputation damage, to deter such a dishonest Sync-committee act. These probabilities and practical security aspects are sufficient to leverage the trustless bridging model.

We cover a hypothetical case of dishonest collusion between Sync-committee and Relay nodes in Section 6.

To reiterate, in general, only one honest Relay node is required so that the relevant events are not missed to be relayed. Since we do not trust the intermediary bridge nodes for the correctness of the passed information, this Relay bridging is popularly referred to as a

Trustless bridge. So in general, we rely only on relay bridge nodes for data passing and not correctness.

Protocol

So, in the design of Supra's HyperNova connecting Ethereum to Supra, we have a set of Relay nodes running Ethereum Beacon and Execution full clients. As mentioned previously, HyperNova only requires a guarantee that not all of the nodes in this set are Byzantine. They watch for the relevant events on Ethereum, mainly of two kinds:

Sync-committee handover that happens every 256 epochs (approx 27 hours). The Beacon state maintains the lists of the current Sync-committee public keys and the next one. So, at the end of a Sync-committee period, say s, the aggregate signature of the existing Sync-committee on the Beacon block validates the selection of the Sync-committee for $s+2$. As long as we are tracking the current Sync-committee, we know the next Sync-committee and the next-to-next Sync-committee correctly (because of this handover).

Cross-chain requests typically of the sort of locking funds or cross-chain smart contract calls on Supra's smart contracts on Ethereum.

After detecting such events, the Relay nodes package them with their inclusion proofs and Beacon block header into a transaction and submit it to the Supra chain. Since the Supra chain maintains the public keys of Ethereum Sync-committee validators, it can fully validate the submitted Beacon block header and the submitted event and determine that they are indeed from Ethereum. Then, the Supra chain can take appropriate actions to update the Sync committee or fulfill cross-chain requests, such as releasing funds.

Other Bridges, such as Near's Rainbow Bridge [10] connecting Near and Ethereum, T3rn Bridge [8] and Snowfork [12] Bridge connecting Polkadot and Ethereum, are also built on the sample principles as HyperNova leveraging Ethereum's Sync-Committee.

Interestingly, this idea naturally extends to many other chains. We are building HyperNova instances for other chains, including Aptos and Sui. Similar to the aforementioned process for Ethereum, we also gather the validator set information. Then we have a bridge-relay node relaying the relevant events on these chains to Supra SMR where these events are validated on Supra SMR.

Supra L1's Compatibility to Relay Bridging

An important point to note is that Supra's Layer 1 consensus protocol uses threshold signatures for attesting blocks. Verifying Supra's blocks on any other chain is as easy and efficient as verifying one BLS signature. That is to say that from Supra's side, trustless Bridging naturally derives from the first principles of its infrastructure's design.

4 Correctness Properties

HyperNova is a pairwise Relay Bridge protocol connecting two blockchains, enabling information and asset transfer between them. A set of Relay nodes realize it; they run the clients of both the source and destination chains to be aware of the events on both chains. We now define the properties we expect the bridge to satisfy.

We use *Bridge requests* for the transactions on the source chain requesting a message transfer, an asset transfer, or some service from the Bridge network. The corresponding transaction posted on the destination chain by the Bridge nodes is termed the *Bridge response*.

The Bridge requests could come with an optional *revert* period τ_{rev}, indicating that the submitter of the request expects his/her transaction on the source chain to be reverted in case an appropriate Bridge response is not posted on the destination chain within τ_{rev}. τ_{rev} is specified as a wall-clock time approximated by block time stamps or the number of blocks on the destination chain. Such requests are termed *revertable*, and the corresponding reverting transactions on the source chain are termed *reverted Bridge requests*.

We use the following notation and definitions in describing the properties expected from the bridge:

$\langle \texttt{req}_1, \texttt{req}_2, \ldots, \texttt{req}_k \rangle$ indicates the transactions on a chain respecting the total order from $\texttt{req}_1$ to $\texttt{req}_k$.

`valid(req,res)` is a relation that holds on a tuple – a Bridge request and a Bridge response, only if they are valid and successful.

`validRevert(req, rev)` is a relation that holds on a tuple – `req` a Bridge request and `rev` a transaction on the source chain that reverts the request.

The Relay nodes can be classified into the following:

Honest node that always follows the Bridge protocol steps and does not deviate from expected actions or responses according to the defined protocol.

Malicious nodes can arbitrarily deviate from the protocol. The actions taken by the node need not maximize the utility of the node nor result in an economic incentive. Nodes compromised by the adversary can act malicious and are called Byzantine.

We now present the properties we expect from the Bridge and group them as the classical safety and liveness properties.

Safety

Validity.
- Every pair of non-revertable Bridge request `req` and its Bridge response `res` must satisfy `valid(req, res)`.
- Every pair of revertable Bridge request `req` and its Bridge response `res` satisfies `valid(req, res)` if and only if `res` occurrs within τ_{rev}.
- Every pair of revertable Bridge request `req` and its reverted Bridge request `rev` satisfies `validRevert(req, rev)` if and only if there is no Bridge response `res` within τ_{rev} such that `valid(req, res)` holds.

Ordering.
- Let the ordered sequence of Bridge requests be $\langle \texttt{req}_1, \texttt{req}_2, \cdots \rangle$, and let the ordered sequence of Bridge responses be $\texttt{s} = \langle \texttt{res}_1, \texttt{res}_2, \cdots \rangle$. Then, for every $1 \leq i$, if $\texttt{res}'_i \in \texttt{s}$ then $\texttt{valid}(\texttt{req}_i, \texttt{res}'_i)$ holds, otherwise $\texttt{req}_i$ is a revertable Bridge request.
- Let the ordered sequence of reverted Bridge requests be $\langle \texttt{rev}_1, \texttt{rev}_2, \cdots \rangle$. Then there must be an ordered subsequence of revertable Bridge requests $\langle \texttt{req}_1, \texttt{req}_2, \ldots, \rangle$ such that $\texttt{validRevert}(\texttt{req}_i, \texttt{rev}_i)$ with $1 \leq i$ holds.

Liveness

- For every non-revertable Bridge request, there must exist a corresponding Bridge response.
- For every revertable Bridge request, there is either a corresponding
  - Bridge response `res` within τ_{rev}, (XOR)
  - reverted Bridge request.

5 HyperNova

In this section, we describe the HyperNova cross-chain protocol, discuss its features, and expand on its security.

As mentioned, HyperNova is a pairwise Relay Bridge protocol connecting two blockchains, enabling information and asset transfer between them. It has a set of Relay nodes that relay the Bridge requests from the source chain to the destination chain. The smart contracts deployed on both chains implement the semantics of the Bridge, meaning codifying the relevant responses for different requests.

The essential protocol is explained in Section 1 and illustrated for the case of Ethereum in Section 3. We now discuss some nuances beyond the basic protocol and premises under which the protocol is secure.

Reverts

The feature of *reverts* can be offered, given the notion of common time (see Section 4). We assume that the smart contract language on the destination chain exposes a `Block.timestamp` service that gives the value of the timestamp of the Block that includes the transaction. Note that the revertable Bridge requests come with τ_{rev} parameter. The smart contract on the destination chain can then be designed so that a successful Bridge response occurs only when the `Block.timestamp` $< \tau_{rev}$.

As long as there is one honest node amongst the relayers, the destination chain is guaranteed to receive the relay of every Bridge request on the source chain. It is possible that this request is received later than τ_{rev} for a revertable Bridge request. In that case, the Bridge response is marked as failed by emitting an appropriate event on the destination chain, which the Bridge Relay nodes are expected to relay to the source chain. Upon receiving this reverting transaction, the source chain reverts the Bridge request accordingly.

In some chains, the ordering and execution of transactions are decoupled, and typically, the execution trails the ordering. In these cases, the timestamp on the block generally conveys the ordering time and not the executed time. We ignore this time difference, and for reverts, we take the block's timestamp.

Security

Premise 1. The *premise* of Relay Bridging is that a smart contract on the destination chain can validate the event of a source chain. Because of this, all the safety properties of Section 4 are satisfied readily.

Premise 2. The safety properties are also satisfied for the case of reverts, again for the same reason that the failure of the Bridge response on the destination chain is validated by the source chain smart contract. For the reverts, then, we need an additional premise that a smart contract on the source chain validates the events of the destination chain.

Trusted Root. As far as we know, there are two ways through which the above premises are met: using a Zero Knowledge Proof (ZKP) and via information about the validators. The ZKP method is straightforward: the Relay nodes construct the ZK proof and forward it to the destination chain smart contract for verification. As for the latter, the destination chain smart contract needs to maintain the list of active validators of the source chain. We require the active validators to produce blocks multi-signed (or threshold-signed) by the source chain's consensus protocol. Then, these blocks can be verified on the destination

chain. However, note that the active validators of a blockchain keep changing. Normally, the notion of an epoch is introduced where the validators are fixed for the duration of an epoch, and the set of validators is modified only at the epoch boundaries. Typically, there is a *trusted handover* wherein the validators of one epoch multi-sign the set of validators for the next epoch. The destination chain smart contract is then updated with the new set of validators. However, the protocol is not entirely trustless, as the first set of validators registered on the destination chain's smart contract is to be trusted. This initial trusted setting is the *Trusted Root.*

So, we have the following theorems under Premises 1 and 2 and the Trusted Root assumption.

▶ **Theorem 1** (Safety). *HyperNova satisfies all the safety properties (of Section 4) even when all the Relayers are malicious.*

Owing to the above theorem, the Bridge Relay nodes need not be restricted to only those enrolled with Supra. In fact anybody in the world can play this role of a Relay node, making this solution a truly anonymous and permissionless solution. Since there is no requirement for any registered set of Bridge nodes, we call this a **Bridgeless Cross-Chain Solution**.

▶ **Theorem 2** (Liveness). *If at least one honest relay node exists in the bridge network, HyperNova satisfies the Liveness property (of Section 4).*

6 Discussion

Resolution to the Sync committee collusion attack

Suppose that a dishonest and colluding Sync-committee is formed, and they also collude with a Relay node and successfully post a fraudulent block to the Supra chain. Note that the finality of a block on Ethereum chain requires 12 minutes. By this time, an honest Relay node submits the correct block. Then upon seeing conflicting blocks for the same height with valid Sync-committee signatures being relayed, the smart contract on Supra chain can be programmed to ignore any blocks for this height and the bridge requests therein. So all it needs is one honest node to instruct Supra chain to not act upon the Bridge requests in a block generated by such a collusion attack. The node submits the correct block within the block finalization time.

A governance entity, typically a DAO network, can then look into these conflicting blocks, validate the correct block, and send a special transaction to Supra's bridge smart contract to process the bridge requests of that block.

Thus in the specific case of bridging Ethereum, we require an honest bridge node to submit the correct block within the block finalization time, to preserve the safety of the bridge. Note that this requirement of at least one honest node for safety is an exception for Theorem 1.

A detailed analysis of this collusion goes into the complexities of Game theory under the modeling of *rational* nodes. Modeling the validators of a blockchain as rational is known to be hard, and we reserve this to be a work outside the scope of this paper.

Staking

Typically, the concept of staking is considered only when there is a possiblity of safety violation due to malicious nodes. Because of Theorem 1, we need not consider any stakes for the designated Relay nodes.

But from the practical viewpoint of making this cross-chain solution efficient and highly responsive, a basic incentive-penalty mechanism on a set of designated Relay nodes is still useful. Hence, as part of future work, we will be exploring the utility of small deposits and giving rewards for these relay nodes.

Implementation

Using Helios (A16Z Ethereum Consensus Light Client [1]) we have prototyped a HyperNova instance between Supra and Ethereum. Similarly, we have also built a HyperNova instance between Supra and Aptos. Work on other chains is underway.

7 Conclusion

HyperNova is Supra's trustless cross-chain solution. It realizes a Bridge without introducing a new security requirement thereby removing the problem of becoming the weakest link in the circulation of assets across chains. The central idea is to validate the events of the source chain on the destination chain directly so that bridge safety is readily satisfied, preserving the security of the chains.

References

1 A16z helios. https://github.com/a16z/helios.git.
2 Altair has no Light Client. https://prestwich.substack.com/p/altair.
3 Aptos - white paper. https://aptos.dev/aptos-white-paper/.
4 Arbitrum bridge. https://bridge.arbitrum.io/.
5 Chain analysis report: Cross-chain-bridge-hacks-2022. https://www.chainalysis.com/blog/cross-chain-bridge-hacks-2022/.
6 Cosmos - inter-blockchain communication protocol. https://cosmos.network/ibc/.
7 Ethereum altair fork. https://www.chainalysis.com/blog/cross-chain-bridge-hacks-2022/.
8 Exploring Eth's Altair Light Client Protocol: t3rn's vision. https://www.t3rn.io/blog/exploring-eths-altair-light-client-protocol-t3rns-vision.
9 Gasper. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/gasper/.
10 How the rainbow bridge works. https://aurora.dev/blog/2021-how-the-rainbow-bridge-works.
11 Polkadot network - whitepaper. https://assets.polkadot.network/Polkadot-whitepaper.pdf.
12 Snow fork. https://snowfork.com/.
13 Succint. https://www.succinct.xyz/.
14 Sui labs research. https://sui.io/research.
15 Supra intralayer. https://supraoracles.com/intralayer-product/.
16 Telepathy. https://docs.telepathy.xyz/.
17 Tendermint. https://docs.tendermint.com/.
18 Zeta chain. https://www.zetachain.com/.
19 Zk bridges: Empowering the cross chain world with zero knowledge proofs. https://tinyurl.com/zkbridge-empower-bridge.
20 zkCasper. https://research.polytope.technology/zkcasper.
21 Axelar Network. https://axelar.network/axelar_whitepaper.pdf, 2022.
22 Layer Zero. https://layerzero.network/, 2022.
23 Wormhole. https://wormhole.com/, 2022.
24 Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget, 2019.

25 Oana Ciobotaru, Fatemeh Shirazi, Alistair Stewart, and Sergey Vasilyev. Accountable light client systems for pos blockchains. Cryptology ePrint Archive, Paper 2022/1205, 2022. https://eprint.iacr.org/2022/1205.

26 Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.