Optimistic Bridges

A New Paradigm for Crosschain Communication

Arjun Bhuptani

Follow

Connext

--

3

Listen

Share

A couple of months ago, we announcedour close partnership with Nomad, a crosschain communication protocol that uses fraud proofs (similar to Optimistic Rollups) to relay data cross chains.

In this post, we'll deep dive into how optimistic bridges work, what their tradeoffs are, and why we love them.

Recap: The Interoperability Trilemma

The Interoperability Trilemma is a model to explain the tradeoff space around bridging, with a taxonomy of the types of crosschain communication protocols that exist today.

When we wrote about the trilemma last year, we classified bridges into three types based on how they are verified:

- Locally verified (atomic swaps & fast liquidity systems)
- Externally verified (multisig, mpc, threshold, PoS, and validator bridges)
- Natively verified (light client header relays, rollup bridges)

In each case, the verification mechanism led to a tradeoff of at least ONE of three highly desireable properties:

- Trust-minimization: adding no economic security assumptions beyond those of the underlying chain.
- Generalizeability: supporting passing arbitrary data across chains.
- Extensibility: deployable to many heterogenous chains with minimal custom work.

Optimistic Verification

Unlike locally, externally, or natively verified bridges, optimistic bridges explore a new tradeoff: latency

.

Here's how they work at a high level:

1. Similar to other bridges, data is posted on an origin chain to a contract function by a user or dApp.
2. An agent called an updater

signs a merkle root containing the data from (1), and posts it to the origin chain. Similar to a rollup sequencer, an updater bonds funds that are slashed in the event of fraud.

1. At this point, any relayer system (e.g. Gelato, Keep3r, Biconomy, etc.) can read this root on the origin chain and post it to one or many destination chains.
2. Posting the data to the destination chain starts a 30 minute fraud proof window (similar to an optimistic rollup exit window). During this time, anyone watching the chain (a watcher

) can prove fraud on the origin chain and disconnect the communication channel to the destination. If this happens, the updater's bond is slashed i.e. their funds are taken and given to the disputing watcher.

1. If no proof of fraud occurs within the 30 minute window, the data passed to the destination chain can be considered finalized and consumed by an application. Typically, this happens by having a service provider (a processor

) submit a merkle proof of the data for the bridge and then utilize that data to call a contract function on the destination chain.

Because the data passed around is completely arbitrary, optimistic bridges let us build any type of crosschain applications/usecases with minimal trust

. Some examples:

- Lock-and-mint or burn-and-mint token bridging.
- Connecting DEX liquidity across chains in a single seamless tx.
- Crosschain vault zaps and vault strategy management.
- Critical protocol operations such as replicating/syncing global constants (e.g. PCV) across chains.
- Bringing UniV3 TWAPs to every chain without introducing oracles.
- Chain-agnostic veToken governance.
- Metaverse-to-metaverse interoperability.

Economic Security Model

Similar to optimistic rollups and state channel networks, optimistic bridge designs rely on a set of watchers to watch the chain and report fraud. This is a fundamentally different security model than externally verified bridges in the same way that rollups have a fundamentally different security model than sidechains.

The Cryptoeconomics of Externally Verified Bridges

Externally verified (i.e. multisig, validator, PoS, mpc, or threshold) bridges (and sidechains/L1s themselves!) utilize an honest majority assumption

— in other words, m of n

participants in the system need to correctly verify updates. In cryptoeconomic terms, this means:

The cost to attack an externally verified bridge with n

verifiers is equal to the cost to corrupt or hack m

verifiers.

It’s important to note that this is a new

potential vector for attack —unless the bridge’s economic security is greater than the cost of 51% attacking the chain, this necessarily means that the externally verified bridge adds a (often significant) trust assumption.

Full economic security for an externally verified bridge could alternatively be achieved if the system can (a) reliably prove fraud, and (b) pay users back the full amount of all of the value that could be lost in a hack. In other words, users and/or LPs can only be insured if the value of slashable stake (e.g. in RUNE on Thorchain) is greater than or equal to the TVL of the whole system. Note that (a) is a strong assumption — conclusively proving fraud here itself requires a trustless crosschain communication mechanism, making the problem somewhat recursive.

The Cryptoeconomics of Optimistic Bridges

The watcher + fraud proof pattern, on the other hand, uses a single honest verifier

assumption. In other words, optimistic bridges only require 1 of n

parties in the system to correctly verify updates.

The cost to attack an optimistic bridge with n

verifiers is equal to the cost to corrupt or hack n

verifiers.

If the watchers for an optimistic system (beit rollups, channels, or bridges) are permissionless (and we assume the underlying chain is live), then the economic cost of attacking the system is unbounded

. This is because there is no way to be sure that there is not at least one single watcher operating anonymously in the world that can prove fraud.

This has a really interesting consequence:

In an optimistic bridge, the EV of attempting fraud is always

negative because, so long as the underlying chains are secure, no amount of money can guarantee that your attack will be successful. As such, the amount of slashable stake that needs to be bonded by the updater only needs to be high enough to prevent attempts

at fraud (i.e. stop griefing).

This is why ORU sequencers, for instance, only need to bond a small subset of the total TVL of a rollup.

Failure Modes

Perhaps the most important improvement that optimistic bridges make over externally verified bridges is to trade off liveness

for safety

. In other words, so long as the underlying chain itself is secure, the theoretical worst-case-scenario is no longer a loss of funds,

it's a system halt.

Updater DoS

Similar to a rollup sequencer, it's possible for a centralized updater to maliciously or accidentally halt the system if it ceases to sign updates.

Decentralizing Nomad's updater is a fairly straightforward task, however. A simple example construction would be to have many bonded updaters (rather than a single one) and use a round-robin approach to signing updates, with failover and slashing if a given updater misses their "turn".

Updater Fraud

Any data that is relayed across chains in an optimistic bridge must be signed by the updater, meaning that any fraud in the system must originate from the updater as well.

In optimistic bridges, fraud is always deterministically provable on the origin chain (similar to how ORU fraud is always provable on L1). To do this, a watcher simply has to submit a proof of an invalid update to the origin chain contract, which then results in the updater being slashed. Then, the watcher submits a signed message to the destination chain to "disconnect" the communication channel within 30 minutes (prior to when that fraudulent data can be considered finalized).

It's actually not necessary to prove fraud on the destination chain at all. Doing it on the home chain correctly penalizes the updater which disincentivizes fraud in the first place and subsequently disconnecting the communication channel mitigates any potential damage done. That said, the ability for a watcher to arbitrarily disconnect a communication channel does open up a DoS vector, which we discuss below.

Watcher DoS

Because any watcher can initiate disconnect a communication channel in an optimistic bridge, it is possible for watchers to grief/permanently halt a given communication channel by spamming disconnections. Note that watchers do not gain anything from the system by doing this (any funds/data remain secure), and the risk of this is compartmentalized per communication channel

(i.e. disconnecting one channel would not take down the system as a whole).

This type of attack vector can be mitigated longer term by introducing the right kinds of incentives for watchers. Since watchers earn the updater's slashed bond when correctly disputing, we can mitigate watcher griefing by introducing a baseline tax for watchers to initiate fraud proofs. This tax would need need to be (a) high enough to disincentivize spam, but also (b) low enough compared to the updater's bond that watchers still have a strong incentive to initiate valid

fraud proofs. Another simple solution would be to simply post the disconnection signature generated by the watcher to the

home chain and slash the watcher if fraud could not be proven.

For now, Nomad handles this problem by permissioning the watcher set. This changes the economic security of the system because now there are a fixed/known set of watchers who could be corrupted (thus bounding the cost of attack). However, we consider this to be an acceptable stopgap solution because there is a straightforward and highly credible path to trust minimization

. This approach also mirrors to how other fraud proof systems have been rolled out:

- State channel networks have historically started with a permissioned watcher set to mitigate the exact same style of griefing vectors until the right style of incentives can be built.
- Optimistic rollups are currently in the same type of bootstrap phase where fraud proofs and disputing have not yet been activated. While this means that the rollups are more trusted currently, the broader community understands that this is only a temporary “training wheels” phase until implementations become more mature.

Chain Liveness Failures

The core assumption that we’ve discussed above is that the underlying chains themselves are able to accept transactions from watchers. This assumption is the same for any fraud proof based system, where the typical construction has some proof window

within which watchers must

complete a transaction to chain.

Nomad has parametrized their 30 minute latency based on existing research into the cost of attacking probabilistic finality chains. We’ll try to break down the research/logic behind this parametrization in a future blog post.

Comparison to Other Approaches

Every distributed system has its tradeoffs —there is no free lunch in bridging

. By far, the most glaring tradeoff of optimistic systems is the addition of the 30 minute latency for transfers, though we believe this can be mitigated by using a modular design that layers Connext on top (more on this soon!).

M of N Bridges

As we’ve shown above, optimistic bridges offer a massive step up in security & trust-minimization compared to externally verified (multisig, threshold, mpc, or validator set based) bridges. The 1 of N

security model of optimistic bridges can mitigate devastating attack vectors related to collusion or compromised keys.

For example, the $625m Ronin Bridge hack would not have been possible if Ronin had used an optimistic bridge even if all of the keys were compromised.

A similar comparison can be made to LayerZero, which utilizes two overlapping m of n

sets that functionally act simply as a larger m of n

set (where the participant set size and collusion vectors become harder to reason about unless the identity of all participants of both sets is known).

Atomic Swaps & Fast Liquidity

Locally verified systems (like Connext’s current implementation of nxtp), while trustless and easy-to-deploy like optimistic bridges, fail to support arbitrary data passing across chains.

In this sense, they underperform compared to optimistic bridges for anything beyond transfers of funds & simple contract execution. That said, it’s likely that they will still remain highly useful as mechanism to mitigate the other tradeoffs of optimistic bridges, namely latency.

Header Relays

Light client header relay systems like IBC work by validating the consensus of chain B inside of the VM of chain A. Header relays give us the theoretical best-in-class trust assumptions because the validator sets of each underlying chain verify each other — no additional parties (unlike externally verified bridges) or assumptions about liveness (like optimistic bridges) are

introduced. They are also not beholden to the latency tradeoff that optimistic bridges are.

That said, header relays are not without their own challenges:

- A custom light client must be built for each new chain/consensus mechanism.
- Specifically for Ethereum PoW, the cost of verifying consensus is prohibitive because of high memory requirements.
- Header relays may not work with optimistic rollups because of rollback risk — this is definitely an open area of research, however!

ZK Bridges

While no trustless ZK bridges exist in production yet, it is possible to build bridges based on zero knowledge proofs that utilize the same strategy as header relays to validate data across chains.

Similar to header relays, zk bridges have great trust considerations and low latency. They are also likely much cheaper than regular header relay systems because proving consensus no longer needs to happen onchain. In doing so, however, they do introduce some new tradeoffs:

- Similar to light client header relays, a custom strategy must be deployed for validating consensus for each chain, and it's possible they would not work for ORUs at all. For zk bridges, this is even more challenging given that not all chains implement the same cryptographic primitives.
- It's not actually possible to prove all consensus models in zero knowledge. In these cases, a finality gadget of some sort is needed, which adds new trust assumptions.

It's likely that there are also other drawbacks around prover cost and data availability, though these have not yet been thoroughly researched.

The Future of Bridging is Optimistic

Until now, we have not had computationally cheap mechanisms for relaying arbitrary data across chains that do not involve trusted third party verifiers. Optimistic bridges give a very high level of security/trust-minimization while retaining the simplicity and ease-of-deployment of existing multisig bridges.

For this reason, we're incredibly excited about Nomad and think it represents a huge leap forward for crosschain and crossrollup communication.

About Nomad

Nomad is a new design for radically cheaper cross-chain communication without header verification. It will form the base layer of a cross-chain communication network that provides fast, cheap communication for all smart contract chains and rollups.

The Nomad interoperability protocol is live on Ethereum Mainnet, Moonbeam, and Milkomeda, and plans to deploy to every major chain soon.

Website | Documentation | Twitter | Discord | Github | Blog

About Connext

Connext is a network for fast, trustless communication between chains and rollups. It is the only interoperability system of its type that does this cheaply and quickly without introducing any new trust assumptions. Connext is aimed at developers who are looking to build bridges and other natively cross-chain applications. To date, over $1.3b in transactions have crossed the network.

Website | Documentation | Twitter | Discord | Github | Blog

Big thanks to

[Anna Carroll

](https://medium.com/u/30aeebf6c32b?source=post_page-----fb800dc7b0e0----------------------------),

[James Prestwich

](https://medium.com/u/6f74e46a357f?source=post_page-----fb800dc7b0e0----------------------------),

[Pranay Mohan

](https://medium.com/u/d4d84a00af34?source=post_page-----fb800dc7b0e0----------------------------), and the rest of the Nomad team for the ideas and feedback that contributed to this post!