Hop: Send Tokens Across Rollups

Chris Whinfrey

January 2021

Abstract

This paper describes Hop, a protocol for sending tokens across rollups and their shared layer-1 network in a quick and trustless manner. Rollups have the potential to scale the Ethereum network, but each rollup creates a siloed environment for its applications. Moving assets between rollups and the layer-1 network is slow and expensive, diminishing the savings users gain by using the rollup. The Hop protocol allows assets to be moved directly from rollup to rollup, providing cost savings and enabling cross-rollup composability of applications.

1 Introduction

1.1 Rollups

A rollup is a type of layer-2 solution that has become a cornerstone of the Ethereum scaling roadmap. Each rollup provides an execution environment that can process transactions in a similar way to Ethereum itself but at a fraction of the cost. In short, rollups increase the throughput of Ethereum by moving computation and data storage off-chain while keeping some data per transaction on-chain [1]. For the purposes of this paper, we will analyze rollups through the lens of bridging tokens across rollups and between rollups and their layer-1 base chain.

There are two types of rollups: optimistic rollups and zk-rollups [1]. While having complex tradeoffs that are not relevant to this paper, the two types differ primarily in the method in which they are verified by the layer-1 chain they are rooted in.

The way both types of rollups are validated makes moving tokens from a rollup to its layer-1 base chain slow and expensive. This is also true for any data passed from a rollup to its layer-1. The speed at which data can be passed from a rollup to its layer-1 base chain is known as the rollup's "exit time."

Optimistic rollups use fraud proofs, which involve the challenging of state changes in the rollup and the resolution of those challenges on the layer-1 base chain. These challenges take time to play out. Therefore, the exit time for Optimistic rollups is significant; typically between a day and a week.

Zk-rollups use validity proofs, which can be confirmed instantly, but each confirmation is relatively expensive. An exit from a zk-rollup to layer-1 cannot be completed until the next validity proof has been executed. Because of the relatively high expense, zk-rollups may choose to confirm validity proofs infrequently, meaning exit times will not be immediate. The time between confirmations for zk-rollups in production today typically ranges from an hour to a day.

For both varieties, moving tokens from layer-1 to layer-2 and back is expensive and slow. For users to fully realize the benefits of a diverse rollup ecosystem, they must be able to quickly and easily transfer assets between rollups without the layer-1 network becoming a bottleneck – either in time or costs.

1.2 Bridging L1 Tokens to Rollups

Each rollup has a *Native Token Bridge* that bridges tokens between the layer-1 base chain and the rollup. This bridge is typically built into the rollup itself or is at least closely associated with the rollup. A rollup's Native Token Bridge allows users to deposit tokens on the layer-1 network and receive a representation of that token on the rollup. A user can also send the layer-2 token they received back to layer-1, which will burn the layer-2 token and eventually unlock the layer-1 token after the rollup's exit time.

In some cases, applications may also provide custom bridges for their tokens called *Application-Specific Bridges*. Application-Specific Bridges may make sense for applications that already have a more flexible trust model than the rollup itself, applications that have both layer-1 and layer-2 components, or applications that use bespoke sidechains [2].

Lastly, there are *General Token Bridges* like the Hop Bridge. General Token Bridges are provided by a third party and bridge ERC-20 tokens in a generic way.

1.2.1 Canonical Tokens

A bridge may create a new layer-2 token representation of the layer-1 token being bridged, or the bridge may allow users to convert to an existing layer-2 representation. Even if multiple layer-2 token representations of a layer-1 token exist, applications will likely gravitate towards a single representation of the layer-1 token. This is because it's in each application's best interest to be composable with the other applications on the rollup. Therefore, each application will choose to use the version of each token that is most compatible with the other applications. The version most widely adopted is the *canonical* version for that rollup (e.g., "Canonical ETH", "Canonical DAI").

In most cases, the canonical version of a layer-1 token will be the token produced by the Native Token Bridge unless an Application-Specific Token Bridge exists, in which case the version produced by the application's bridge is likely to be preferred.

Because the canonical version of a layer-1 token on a rollup is the one that is most compatible with applications on that rollup, it is likely to be the version that end users want. **The goal of any General Token Bridge should be to allow users to convert between layer-1 tokens and their canonical layer-2 representations, not just create a new representation of the layer-1 token on the rollup.**

1.3 Prior Work

Bridging assets across blockchain networks is not a new problem. Solutions used to bridge various layer-1 networks and variants thereof may be useful for bridging rollups as well.

1.3.1 Proof of Authority Bridges

Proof of Authority (PoA) bridges rely on a set of authorities to attest that a token has been locked up in its native environment and then mint a representation of that token in the destination environment. This method is efficient but introduces a new trust assumption: the authorities will remain honest. A dishonest majority of the authorities can steal the locked funds and flood the destination environment with tokens that are no longer collateralized.

1.3.2 Hash Time Locked Contracts

Hash Time Locked Contracts (HTLCs), used for atomic swaps, allow users to trustlessly swap one asset for another even if those assets are on separate blockchains. The setup is simple:

1. Alice generates a hash with a secret preimage. [1]
2. Alice makes a payment to Bob that requires the secret preimage to claim the payment.
3. Bob makes a payment to Alice that also requires the secret preimage, but Bob does not yet know the secret.
4. Alice claims Bob's payment revealing the secret preimage.
5. Bob uses the newly revealed preimage to claim Alice's payments.

HTLCs have several drawbacks that have been extensively researched. Griefing attacks [5], mass-exit vulnerabilities [4], and the ”free option” problem [6] present serious hurdles for HTLC implementations.

[1] The preimage can be thought of as a secret key, that will eventually be shared with the recipient. This can be a word, a phrase, or even a random series of bytes[7].

1.3.3 Conditional Transfers

Conditional Transfers are similar to HTLC based swaps, but rather than relying on a secret preimage, the first leg of the swap is directly triggered by the second leg of the swap. This means that conditional transfers are not useful for performing atomic swaps across unconnected networks because an action on one of the networks cannot trigger an action on the other.

With rollups, it is trivial for the layer-1 network to trigger an action on the rollup, making Conditional Transfers a more straightforward way to perform atomic swaps than HTLCs when swapping between a rollup and its layer-1 network.

While Conditional Transfers are useful for reducing the time it takes to exit a rollup, there is little to no reduction in the transaction cost of moving tokens from one rollup to the next. All rollup-to-rollup transfers must still individually go through the layer-1 network, which does not relieve the layer-1 bottleneck.

2 Hop Protocol

The Hop protocol provides a scalable rollup-to-rollup General Token Bridge using a two-pronged approach:

1. Create a cross-network bridge token that can be quickly and economically moved from rollup to rollup or claimed on layer-1 for its underlying asset.

2. Use Automated Market Makers to swap between each bridge token and its corresponding Canonical Tokens on each rollup in order to dynamically price liquidity and incentivize the rebalancing of liquidity across the network.

The combined approach allows users to quickly and trustlessly swap between layer-2 Canonical Tokens using the specialized bridge token as an intermediary asset.

2.1 Hop Bridge Tokens

Hop Bridge Tokens (e.g., "Hop ETH", "Hop DAI" with symbols "hETH", "hDAI" respectively) are specialized layer-2 tokens that can be transferred rollup-to-rollup in batches and act as intermediary assets in the Hop protocol. Each Hop Bridge Token represents a deposit in the layer-1 Hop Bridge contract. For example, if 4 ETH are deposited into the layer-1 Hop Bridge contract, 4 Hop ETH can be minted from a layer-2 Hop Bridge contract.

Inversely, a Hop Bridge token can be redeemed for its underlying asset on layer-1, which burns the Hop Bridge Token being redeemed on layer-2. When a Hop Bridge Token is transferred from rollup to rollup, it is burned on the origin rollup and minted on the destination rollup. As explained below, these immediate transfers are accomplished by allowing a "Bonder" to front liquidity on the destination in exchange for a small fee. The Bonder's liquidity is returned

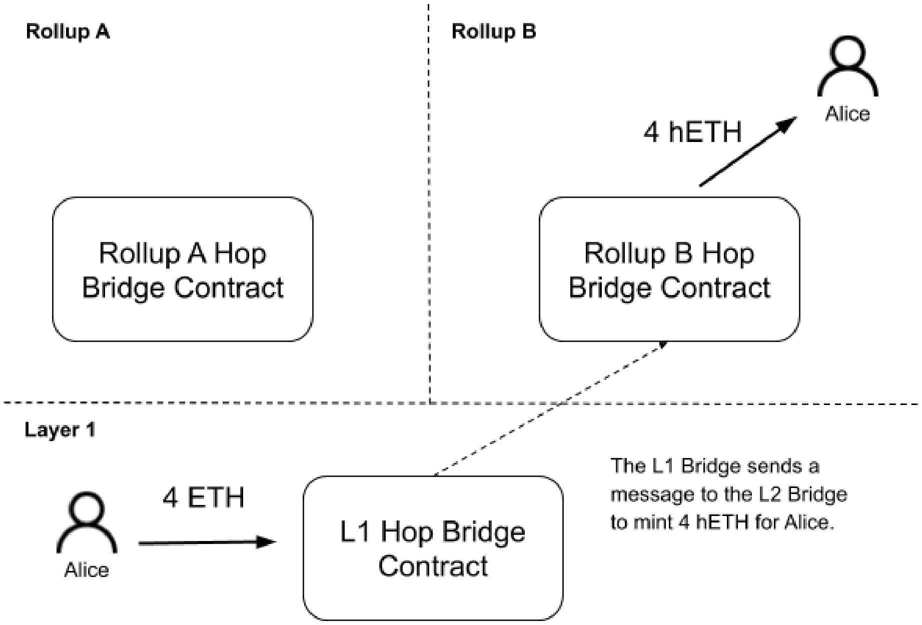

Rollup A
Rollup B
Alice
4 hETH
Rollup A Hop Bridge Contract
Rollup B Hop Bridge Contract
Layer 1
4 ETH
Alice
L1 Hop Bridge Contract
The L1 Bridge sends a message to the L2 Bridge to mint 4 hETH for Alice.

Figure 1: Alice deposits 4 ETH into the L1 Bridge contract and receives 4 Hop ETH from the L2 Bridge contract.

when the transfer eventually propagates through layer-1 as part of a larger bundle called a ”Transfer Root”.

2.1.1 Transfer

A Hop *Transfer* includes the following information:

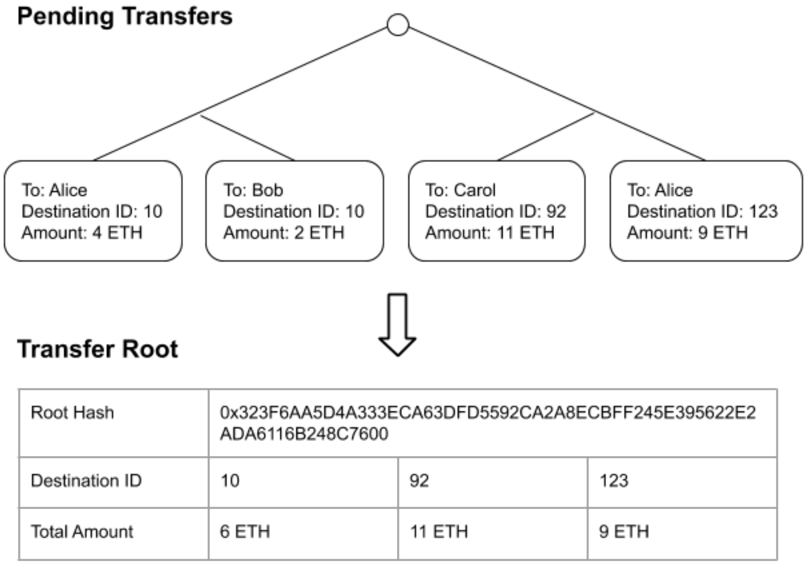
- **Destination chain ID** - The chain ID of the rollup or layer-1 destination
- **Recipient** - The address receiving the Transfer at the destination
- **Amount** - The amount of token being transferred

The Transfer may also include additional information for convenience functionality. For example, it may specify a relayer fee to allow a transaction relayer to withdraw the Transfer at its destination on behalf of the user. A future version of the Hop Bridge that handles multiple tokens would also require a token identifier to be included in the Transfer data.

2.1.2 Transfer Root

A *Transfer Root* represents a bundle of Transfers with minimal data. Each Transfer Root is composed of:

1. A Merkle root of the Transfers
2. An array of each unique destination represented by its chain ID
3. An array of total amounts being sent to each unique destination



Root Hash	0x323F6AA5D4A333ECA63DFD5592CA2A8ECBFF245E395622E2ADA6116B248C7600		
Destination ID	10	92	123
Total Amount	6 ETH	11 ETH	9 ETH

Figure 2: The four pending Transfers are aggregated into a Transfer Root with three destinations.

A Transfer Root can contain thousands of Transfers yet be accounted for on layer-1 as a single bundle. This alleviates the layer-1 bottleneck and allows a large number of transfers to be passed through layer-1 to their destination rollups in a scalable way.

However, propagating a Transfer Root through layer-1 can be a slow process. This is primarily due to the exit time of the rollup that the Transfer Root originated on. In order to fulfill Transfers immediately, an external party can provide up-front liquidity on the destination rollup for a small fee, as described in this next section.

2.1.3 Transfer Bonds

The *Bonder* can verify that the Transfer was made on its origin rollup by running a verifier node for the rollup. The Bonder can then provide up-front liquidity on the destination rollup in order to fulfill the Transfer immediately. Eventually, when the Transfer has reached its destination, the Bonder's funds are restored.

The Bonder may take a small fee in exchange for locking up liquidity while the Transfer is propagating through the system.

The immediate liquidity provided by Transfer Bonds and scalability achieved with Transfer Roots enables Hop Bridge Tokens to be quickly and economically moved from rollup to rollup.

2.2 Automated Market Makers

Each Hop Bridge Token represents a layer-1 token and can be quickly and economically moved from rollup to rollup. Under normal network conditions, each Hop Bridge Token is worth exactly 1 of its L1 counterpart because it can be redeemed on layer-1 at any time.

However, third parties on each rollup are not likely to adopt Hop Bridge Tokens directly. It is more likely that the Canonical Tokens they adopt are produced by the rollup's Native Token Bridge or an Application-Specific Token Bridge as previously discussed. To complete the bridge between the layer-1 token and its Canonical layer-2 counterpart, an Automated Market Maker (AMM) can be deployed to enable swaps between each Hop Bridge Token and its corresponding Canonical Token (e.g., Hop ETH - Canonical L2 ETH).

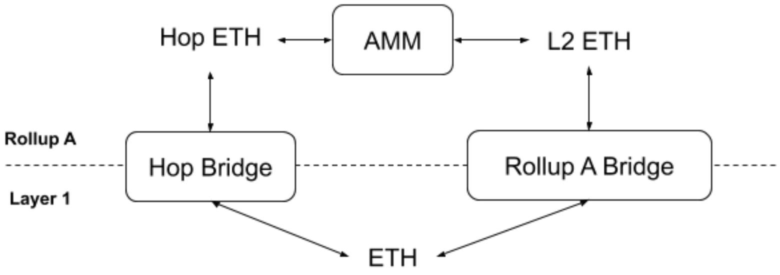


Figure 3: The AMM creates a market between Hop ETH and the Canonical ETH on the rollup.

This market provides a pricing mechanism for liquidity on a given rollup and also acts as an incentivization mechanism for Arbitrageurs to rebalance liquidity in response to market movements.

2.2.1 Arbitrageurs

Arbitrageurs are an external, unsanctioned group of actors in the Hop protocol. The Arbitrageur's role is to take advantage of price differences between layer-1 tokens and their Canonical layer-2 counterparts. By taking advantage of these price differences, Arbitrageurs effectively rebalance liquidity across the supported rollups.

Consider the following scenario with a layer-1 network and a single rollup supported by a Hop Bridge. A user has ETH on the rollup, and they want to move it to the layer-1 immediately. They use the Hop Bridge because using the Native Token Bridge would subject them to the rollup's exit time. First, their ETH will be swapped for Hop ETH (hETH) using the ETH:hETH AMM on the rollup. Then, the Hop ETH can be burned on the rollup and redeemed for layer-1 ETH.

Because exiting over the Hop Bridge involved selling ETH into the ETH:hETH market, it caused ETH to be priced at a small discount to Hop ETH. If this discount becomes large enough (e.g., 1.005 ETH/hETH), the Arbitrageur moves layer-1 ETH across the Hop Bridge to receive Hop ETH. The Hop ETH is then used to purchase the discounted layer-2 Canonical ETH. The Arbitrageur may now choose to move that layer-2 Canonical ETH back to layer-1. Because they do not want to trade back into the market they just arbitraged and lose their profits, they will exit via the Native Token Bridge and take on the liquidity lock up.

2.2.2 Liquidity Providers

Each AMM requires Liquidity Providers to contribute passive liquidity to the AMM's liquidity pool. In return, Liquidity Providers are rewarded with a small fee from each swap (e.g., 0.3%). Typically, Liquidity Providers also risk incurring what is known as "impermanent loss". Impermanent loss occurs when the assets in an AMM diverge in price. Because AMM pairs in the Hop protocol are always assets of similar value, Hop Liquidity Providers have a very low risk of impermanent loss under normal network conditions. Additionally, the AMM's price curve can be optimized for assets that trade within a narrow range [3].

2.3 Rollup-to-Rollup Transfers

With both the Hop Bridge Token and markets on each rollup to swap between the Hop Bridge Token and the Canonical Token, users can quickly and easily convert from one rollup's Canonical Token to the next. Rollup-to-rollup transfers through the Hop protocol are highly scalable because individual transfers do not require any layer-1 transactions.

Consider the following scenario where Alice has Rollup A Canonical ETH and wants Rollup B Canonical ETH:

1. Alice swaps her Rollup A Canonical ETH for Hop ETH using the AMM on Rollup A.

2. Alice then uses the Hop Bridge to send her Hop ETH from Rollup A to Rollup B.

3. Once the Bonder provides liquidity for her Transfer, Alice receives Hop ETH on Rollup B.

4. Alice can now swap her Hop ETH for Rollup B Canonical ETH using the AMM on Rollup B.

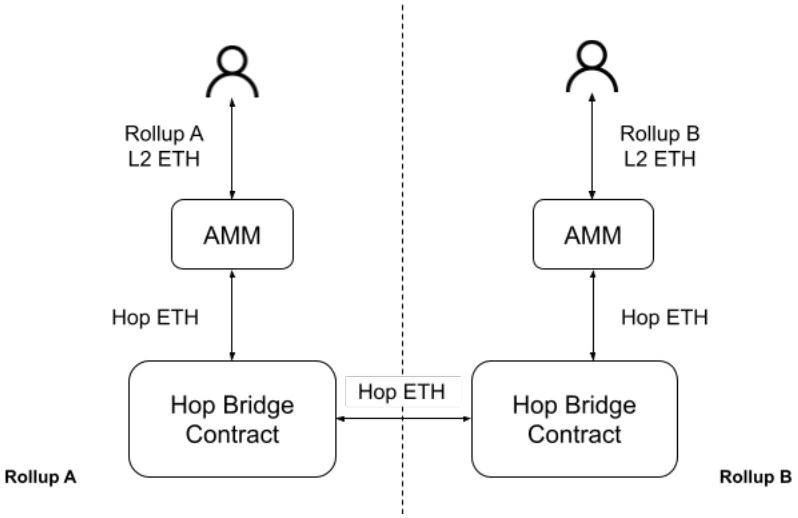


Figure 4: Users swap between the Canonical Token on each rollup using the Hop Token as an intermediary asset.

Eventually, her Transfer propagates through the L1 Hop Bridge, and the Bonder's liquidity is returned.

For convenience, Alice can also make her cross-rollup transfer by making a single transaction. She makes a call to the Hop Bridge, which performs a swap from Rollup A Canonical ETH to Hop ETH for Alice using the AMM and then sends the Hop ETH to its destination. This time, the Transfer is sent with instructions to automatically swap Hop ETH for Rollup B Canonical ETH at the destination. This convenience functionality also makes it easy for other smart contracts to interact directly with the Hop protocol and make cross-rollup transfers.

It is important to note that Alice's Transfer was completed with only layer-2 transactions in both cases. The layer-1 Hop Bridge strictly deals with batches of Transfers rather than individual Transfers themselves. This allows thousands of rollup-to-rollup Transfers to be completed with minimal layer-1 interaction.

3 Areas for Further Research

3.1 Decentralize the Bonder role

In its most basic form, the Bonder is a single, external party that verifies and fronts liquidity for Transfers and Transfer Roots. However, relying on a single Bonder to be available at all times and to have enough liquidity to meet demand is less than ideal. If the Bonder is unavailable, no funds are at risk, but Transfers propagate through the system at a much slower pace.

Allowing for a network of Bonders can create a more robust system presents its own challenges. Choosing which Bonder will be rewarded for bonding a particular Transfer or Transfer Root must be done in a way that does not frequently lead to failed transactions and does not allow a denial-of-service attack.

3.2 Broader Layer-2 Support

It may be possible for non-rollup layer-2 solutions to be compatible with the Hop protocol. Supporting a broader set of solutions will improve the flexibility and may improve the efficiency of the system. It also may be possible to support bridging to other layer-1s, but it will likely be challenging to do so without introducing new trust assumptions.

3.3 Limiting Liability

In the current construction, a Hop bridge is only as secure as the weakest supported rollup. This presents a systemic risk to participants in the system. While regular users moving tokens across the Hop bridge are only exposed momentarily, liquidity providers for the AMMs and some arbitrageurs have constant exposure to this risk. It may be possible to limit the liability that each rollup presents to the system, reducing the system's overall risk.

3.4 Enable Contract Calls

Smart contracts and externally owned accounts can send tokens across the Hop Bridge, but they cannot send data to or make contract calls on the destination rollup. This functionality is straightforward to implement, but the exact methodology and security risks must be carefully considered.

References

[1] Vitalik Buterin. An incomplete guide to rollups. `https://vitalik.ca/general/2021/01/05/rollup.html`, January 2020.

[2] Compound. Compound cash. `https://compound.cash/](https://compound.cash/`, December 2020.

[3] Michael Egorov. Stableswap - efficient mechanism for stablecoin liquidity. `https://www.curve.fi/stableswap-paper.pdf`, November 2019.

[4] Aviv Zohar Jona Harris. Flood & loot: A systemic attack on the lightning network. `https://arxiv.org/pdf/2006.08513.pdf`, June 2020.

[5] Cristina Perez-Sola, Alejandro Ranchal-Pedrosa, Jordi Herrera-Joancomart, Guillermo Navarro-Arribas, and Joaquin Garcia-Alfaro. Lockdown: Balance availability attack against lightning network channels. `https://eprint.iacr.org/2019/1149.pdf`, 2019.

[6] Dan Robinson. Htlcs considered harmful. http://diyhpl.us/wiki/transcripts/stanford-blockchain-conference/2019/htlcs-considered-harmful/, January 2019.

[7] taconator Ryan R. Fox, John M. Jones. bsip-0044. https://github.com/bitshares/bsips/blob/master/bsip-0044.md, August 2018.