LayerZero

In the cryptocurrency ecosystem there are a lot of different chains that can't easily communicate with each other. Until recently communication between chains needed a trusted centralized authority or smart-contract intermediaries that involve an intermediary token and require a second transaction to complete the initial transaction between two chains.

LayerZero comes to solve that problem providing an interface to allow supported chains to communicate between each other seamlessly without needing to trust the intermediaries involved in the transaction. Every supported chain has a LayerZero endpoint that consists of Communicator, Validator, Network, and Libraries. The first three modules comprise the core functionality of the Endpoint, while each new chain supported by LayerZero is added as an additional Library. LayerZero works by using an independent oracle and relayer to transfer the transaction and proof between both chains ensuring valid delivery in a trustless way.

In the next few paragraphs I'll describe how LayerZero manages to ensure valid delivery in a trustless way using every component mentioned until now.

First the user application on the first chain executes some actions as part of a transaction. A step included in the transaction is the transmission of a message over LayerZero with valid delivery conditioned on it.

Then the Communicator module constructs a LayerZero packet containing dst and payload, and sends it, along with the transaction and relayer arguments, to the Validator module. When the Validator module receives the information it sends the transaction and dst to the Network module. This step notifies the Network module that the block header for the current block on the first chain needs to be sent to the second chain. At the same time the Validator module forwards the packet, transaction, and relayer arguments to the Relayer, notifying the Relayer that the transaction proof for the transaction needs to be prefetched and eventually sent to the second chain.

The Network module sends dst and the block ID of the current transaction to the Oracle. This notifies the Oracle to fetch the block header for the current block on the first chain and send it to the second chain. In the event that multiple LayerZero transactions occurred in the same block, this is done once.

The Oracle reads the block header from the first chain. Then the Relayer reads the transaction proof associated with the transaction from the first chain, and stores it off-chain.

After that the Oracle confirms that the block corresponding to block header is stably committed on the first chain and then sends it to the Network module on the second chain. On the second chain the Network module sends the block hash to the Validator.

Then the Validator module forwards the block hash to the Relayer. After receiving the block hash, the Relayer sends a list of any Packets, transactions and proofs that match the current block. In the event that multiple users simultaneously send messages between the same endpoints, there may be multiple packets and associated transaction proofs within the same block.

The Validator module uses the received transaction proofs in conjunction with the block headers stored by the Network module to validate whether the associated transaction is valid and committed. If the block header and transaction proof do not match, then the message is discarded. If they do match, then the Packet is sent to the Communicator module. Lastly the Communicator module emits the Packet to the dApp on the second chain, thus completing the transaction between chains.