

This post outlines an in-protocol

cross-shard messaging system by proposing changes to the block validity and fork choice rules. This is a part of the CBC Casper sharding proposal, but can be easily applied to later phases of Eth2.0. A description of this can be found in the [CBC Casper draft paper](#) in Section 4.6

1. Objective

Eth2.0 supports cross-shard messaging on an out-of-protocol basis, meaning that users have to manually perform some operations to deliver a cross-shard message. The protocol handles the delivery of the receipt of the transaction from a shard chain to the beacon chain, and the user has to create a transaction on the recipient shard that refers to this receipt.

[

image

952×440 21.3 KB

](https://ethresear.ch/uploads/default/original/2X/e/e77798d734620817e32e26701c7b8976bedd269b.png)

The goal is to have a protocol which handles the end-to-end delivery of the cross-shard message, in which the user only has to generate a transaction on the first shard chain.

[

image

959×454 27.8 KB

](https://ethresear.ch/uploads/default/original/2X/8/87058648f17ec3d961b51c9d8b4833499943bb4c.png)

2. Proposal

2.1. Block Structure

Blocks now contain 2 new logs, one for sent messages (in this block) and received messages (in this block) each.

2.2. Messages

Messages are objects that contain:

- sender_shard_ID
- recipient_shard_ID
- final_destination_shard_ID
- target_block

on the recipient shard

- blocks_to_live

parameter

- payload_tx

Any cross-shard message is going to have to make 2 hops - Shard A to Beacon Chain, and Beacon Chain to Shard B. sender_shard_ID

and recipient_shard_ID

are for the specific hop, and final_destination_shard_ID

identifies Shard B.

The message can be received in blocks on the recipient shard between heights height(target_block)+1

and height(target_block)+blocks_to_live

.

[

image

980×287 12.5 KB

](https://ethresear.ch/uploads/default/original/2X/1/16f8ef8713adaea98a3fccda746e99e57a58cf30.png)

In the above figure, the `target_block`

is `b2`

and `blocks_to_live`

is 2. This message can be received only in blocks `b3`

& `b4`

.

2.3. Block Validity & Fork Choice

The atomicity of sends and receives needs to be maintained; either a message is sent in the sender shard and received in the recipient shard (eventually), or it does not appear anywhere.

I'll skip the obvious ones like: each message can only be received once, chains can only receive messages destined for themselves, etc.

2.3.1. Shard Chains

The shard chain fork choice follows the beacon chain fork choice, and these modifications enforce the send-receive atomicity:

- If the beacon chain sends a shard some message, then the `target_block`

is in the fork choice on the shard chain.

- If the beacon chain receives some message from a shard, the sender block is in the fork choice on the shard.
- If the beacon chain does not receive a shard chain message by its expiry, then the sender block is orphaned in the shard.
- Any shard chain block that doesn't receive a yet-unreceived message by its `blocks_to_live`

expiry is orphaned.

The below figure illustrates the first rule:

[

image

1118×510 18.5 KB

](https://ethresear.ch/uploads/default/original/2X/2/257d1443ed4b7d918629fec8d220d3af569a820d.png)

The below figure illustrates the second rule:

[

image

1095×482 18.4 KB

](https://ethresear.ch/uploads/default/original/2X/3/317a5ce5d533d649c88ea23ef30365a7052700d9.png)

2.3.2. Beacon Chain

The beacon chain block validity rule enforces that messages received in a block are immediately redirected and sent to the final destination. If a message is in the receive log of a block, it must also appear on the sent log of the block with:

- sender_shard_ID

identifying the beacon chain

- recipient_shard_ID

as the final_destination_shard_ID

of the received message

- appropriate target_block

and blocks_to_live

on the recipient shard

Since the shard chain fork choices are dependent on the beacon chain fork choice, the beacon chain fork choice must follow this rule to avoid chaos in the shards:

- The beacon chain cannot send to/receive from one fork of a shard and later send to/receive from a conflicting fork of the shard. ("send to a fork" refers to the fork containing the target_block

)

The purpose of this fork choice rule is to avoid the below situation:

[

image

1077×380 19.4 KB

](<https://ethresear.ch/uploads/default/original/2X/3/3d03781b3b392d5b6c68e9863a4d4716b84b4325.png>)

3. How Are Messages Traveling Between Shards?

Messages can get from the beacon chain to the shards/from shards to the beacon chain naturally because of the way validators are distributed:

- Messages from the beacon chain to a shard chain are seen by all validators on the shard, since they are also validators on the beacon chain. Validators on the shard chain also have all necessary information for executing the fork choice.
- Messages sent from a shard to the beacon chain are collected by the respective crosslink committee and included in the crosslink information on the beacon chain. It is important to note that not all beacon chain validators have access to necessary information for executing the beacon chain fork choice and block validity rules. The requirement is that atleast one of the validators on the shard chain provides fraud proofs in case the stated rules are broken.