SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Cryptocurrency Research

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper

Open Access

Comparing Blockchain Bridges: Poly Network vs. IBC

Apurba Pokharel[1*]

[1]RedChillies Lab, Inc., Wyoming, USA. E-mail: pokharel.apurba@gmail.com

Article Info

Volume 2, Issue 2, December 2022
Received : 10 September 2022
Accepted : 24 November 2022
Published : 05 December 2022
doi: 10.51483/IJCCR.2.2.2022.16-21

Abstract

Blockchain is a system where connected nodes work together to perform various activities within the system, and all of them maintain their own copy of a system ledger that stores transactions. With the emergence of new blockchain networks and the massive growth of users on the network the need for inter blockchain communication is now at an all-time high. At first glance the two existing bridging protocols, the Poly Network and the Inter Blockchain Communication (IBC) protocol, seem to be completely different, but a closer inspection reveals a connection between them that will most likely be seen in future blockchain bridges as well. This paper discusses their similarities and differences and ends by making a prediction about future blockchain bridges.

Keywords: *Blockchain, Blockchain network, Bridges, Poly Network, Inter Blockchain Communication*

1. Introduction

As more and more blockchain networks are being developed and their use case is ever on the rise, the need for blockchain bridges that links these solitary and private blockchain are on the rise. This paper aims at comparing two such bridges, Poly network, a bridge already in use and Inter Blockchain Communication (IBC) a new and advance protocol being developed. This paper will list out their similarities, their differences and will try and make a prediction about the future blockchain bridges.

2. Poly Network

In order to ensure authenticity of transactions the destination chains verifies the read / write transactions on the source chain (Poly Team, 1955). This is done via header synchronization and is the task of the Poly chain. The entire process can be described in the steps below:

- Relayers will forward read and write operation from the source chain to the destination chain, i.e., to the cross chain management contract on the destination chain.
- Moreover the destination chain must be provided with the operation (write or read) and a proof of inclusion of the operation on the source chain (Poly Team, 1955).
- The cross chain management contract will than validate the proof using Poly chain and if and only if the proof is valid will the operation be carried out (Poly Team, 1955).

[*] *Corresponding author: Apurba Pokharel, RedChillies Lab, Inc., Wyoming, USA. E-mail: pokharel.apurba@gmail.com*

The validation process can be explained as follows:

- Each node of Poly chain runs full nodes of all participant chains as a result of which the transactions on the participant chains will be available to Poly chain. Running a full node allows Poly chain to perform verification of every block (Poly Team, 1955).
- The Poly chain's output will contain block headers of each blockchain, and cross chain management contract will then be able to use these headers to verify the existence of source chain's header (in the proof) (Poly Team, 1955).
- So participant chain will be able to perform cross chain validation using the Poly chain.
- Poly chain's block only stores the hash of the block header of participant chains, and the respective block header can be provided when needed as each node of the Poly chain will run full nodes.

The verification process used by cross chain management contract can be summed up below:

Users send cross chain operation to source chain. The operation is mined on the source chain. Relayers pick up the cross chain operation and feeds it to the Poly chain, Poly chain will mine new block that contains the hash of the block header of the block mined on the source chain that contains the operation. Relayer will get the operation from the log and will forward the operation as well as proof to the cross chain management contract in the destination chain. The cross chain management contract will use Simple Payment Verification (SPV), is a light weight client that is used for verification of blockchain transactions) to get the source chain block header from nodes in Poly chain, get hash of block header from block mined in Poly chain, and confirm them. In order to prevent tampering with Poly chain's block and provide further security from attacks each block's header will store the merkle root of the merkle tree formed from combining all merkle root in the block header of Poly chain (Poly Team, 1955).

3. Inter Blockchain Communication

IBC is a communication protocol for heterogeneous ledgers (Christopher, 2020). Unlike Poly network, IBC has a different take on the blockchain bridging solution. Poly network uses what is known as a top down approach, i.e., almost any ledgers that have been designed and developed in their own way can communicate with others via Poly Network protocol, whereas IBC uses a bottom up approach wherein only the legers that have been designed from the ground up to be IBC compatible can communicate with other such ledgers. IBC has a certain requirement for the ledger that must be satisfied in order to use the protocol. Just like Poly network, IBC also relies on relayers to achieve cross chain communication and the relayers can read the state of the ledgers and submit data to another. IBC protocol defines three abstractions client, connection and channel in order to provide reliable, flow controlled and authenticated transfer of data packets between IBC modules on separate ledgers (Christopher, 2020).

3.1. Client

Defines the property needed for state verification. An algorithm called validity predicate is used to perform verification of state of the other ledger. Light client is a name given to the part of the IBC module that uses validity predicate for state verification (Christopher, 2020).

3.2. Connection

Connection object contains connection end on each ledger, each associated with a light client on the other ledger. Together they enable cross ledger state verification and packer relay through channels. Connections are established using a handshake protocol (Christopher, 2020).

3.3. Channel

Channel abstraction provides message delivery semantics: ordering, exactly once delivery and module permissioning. In and ordered channel packets are delivered exactly once in the order they were sent. An unordered channel however does not preserve the order of packets upon delivery. All channels provide exactly once packet delivery. The packets on one ledger are sent to the modules on other ledger as specified by the channel end object (Channel end is an object that holds all necessary information while message passing). Channels also contain negotiated encoding and multiplexing options (Christopher, 2020).

Poly network uses SPV and full nodes for verification whereas IBC uses light client for the same purpose.

The flow of data in IBC can be described in the steps below:

- Connection initiated between ledgers and Channels opened over the connection.
- IBC module will create the packet and will call the send Packet function. This function will perform various checks:
  - Check if the channel and connection are open.
  - Check the port.
  - Check if packet metadata matches what was agreed to when opening the channel.
  - Check if timeout height has not passed on the destination ledger.
  - Increases the sequence number counter (a number assigned to each sending packet to ensure reliable packet delivery).
  - Stores a constant size hash of commitment to the packet data and packet timeout.
- Relayers pick it up as they are continuously monitoring the state of the ledger and they pick up the IBC packets that have been committed to the ledgers.
- Relayers are responsible for creating the proof of inclusion as well and they do this using information such as consensus state, client, connection, channel and packet information.
- The recvPacket function is called by a module in order to receiver and process an IBC packet. This function performs various checks:
  - Checks the channel and connection are open.
  - Checks the calling module owns the receiving port.
  - Checks the packet metadata matches to what was agreed to when opening the channel.
  - Checks the packet sequence number is the next sequence that the channel end expects to receive (for ordered channel, for unordered channel this step is skipped).
  - Checks the timeout height has not passed.
  - Checks if the inclusion proof matches for the outgoing ledger's state.
  - Sets the acknowledgment value unique to the packet (this step performed for unordered channel only).
  - Increases the packet received sequence number within the channel end object (for ordered channel only).
  - Once the packet is received the acknowledgePacket is called. This function checks that:
  - Checks the channel and connection are open to acknowledge packets.
  - Checks the calling module owns the sending port.
  - Checks the packet metadata matches the channel and connection information.
  - Checks the packet was actually sent on this channel.
  - Checks the packet sequence is the next sequence, the channel end expects to acknowledge (for ordered channels).
  - Checks the inclusion proof of the packet acknowledgment data in the receiving ledger's state.
  - Deletes the packet commitment (cleaning up state and preventing replay).
  - Increments the next acknowledgment sequence (for ordered channels).

4. Why Poly and Why IBC?

4.1. Connection

Both the protocol upon brief studying depends on a relayer entity for cross chain communication. This intersection between the two different protocols suggested that the working of the protocols at the core was much similar than what is initially perceived. So, even though these protocols have different approaches to cross chain communication, ultimately they share a common similarity in relayers. And this similarity is what made this comparative study possible.

4.2. Market Cap of Poly

Poly network is an already running and fully developed protocol. Many ledgers have partnered with Poly network to enable cross chain transaction and is one of the most widely used blockchain bridging solution at the time of starting this research (2021). Though it has suffered the biggest hack of assets in crypto history the hacker has begun the process of returning all the funds and has been hired as the security analyst within the organization making the protocol less susceptible to future hacks. In a word it is battle tested.

4.3. Sound logic of IBC

IBC is what seems to be the second generation bridging solution that builds on top of the foundations laid by Poly network. Though at their core they are different one is a bottom up approach while the other is a top bottom approach but the similarities cannot be disregarded. IBC's sound logic and detailed attention to every detail (ordering, timeouts, and better verification process) is why IBC was chosen to be compared with Poly network.

5. Discussion

This section will describe the findings of this research.

5.1. Why the Bottom up Approach was Introduced in IBC?

Let us look at the scenario in Polkadot that justifies the introduction of the bottom up approach. Polkadot uses a top down approach.

- In order to pass message between para chains a channel is opened between them.
- The collator of sender chain will gossip this to both light and full nodes of relay chain.
- The gossiped message will have destination and time stamp among other attributes.
- When the collator on the receiving para chain gets the message it will perform initial verification and send the blocks to validators. (Collators run full node of the relay and para chain so they will have access to the gossiped message).
- The validators will validate the block and compress this block which then gets added to the relay chain.

In order to achieve this message passing, protocol named XCMP is used, the collators of Polkadot needs to run full nodes of the relay chain as well as the pair of para chain (*Poly Team*, 1955). The node specification for running a collator will be extremely high as it needs to run full nodes of both the relay and the pair of para chains. Also, the nodes need to be actively listening for events in the para chain and the relay chain. And if this sharded system exceeds its threshold for fault tolerance than the message passing will not work.

This is mitigated in IBC due to the bottom top approach. Due to the use of relayers in IBC which does not run full nodes rather uses validity predicate algorithm or a light client of the pair of ledgers over a channel in order to send cross chain messages.

5.2. Dependence on the use of Relayers

Distributed ledgers are by nature closed off to the internet and can be thought of as an intranet between the peers in the network. The current state of the ledger must be the same when replaying all the transaction from genesis block to the latest block at any given time. This is the reason why block chains aren't API compatible as replaying the same request to the API at another point in time would yield different response and would cause a fork in the ledger. This is also why ledgers aren't aware of one another meaning Ethereum network is unaware of the existence of Bitcoin network as they are closed off to one another and cannot communicate with each other. So, a literal middleman is required that can relay messages between these two closed off networks. Relayer entities were born out of need for such requirement in order to bridge these ledgers. Due to the property of distributed ledgers all future bridging solutions will have a relayer entity.

5.3. Additional Features of IBC

Due to the dependence on relayers the work of bridges are similar. All bridging solution will end up using relayers by necessity and so the new bridging solutions will just be upgraded versions of the old ones with new features and approaches but the major aspect, i.e., relaying will be the same. This applies to PolyNetwork and IBC where the former can be thought of as version 1 and the latter as version 2. The additional features provided by IBC are as:

5.3.1. Reliable Transfer of Packets

IBC module assigns sequence number to each packet being sent. This sequence number is checked by the IBC handler on the receiving ledger. The sequence number is contained by the channel end (A channel end is a data structure that stores metadata associated with one end of a channel on one of the participating ledger).

5.3.2. Ordered Channel

Ordered channel are channel where the data packets are sent and received in order. Ordered channel used sequence number for managing the proper order of data packets.

5.3.3. Reject Data Packets

The encoding options agreed to when opening a channel and the next Expected Sequence number represents the next data packet expected. If anyone if these do no match then the IBC module can reject data packets.

5.3.4. Timeout Packets

A packet being sent contains a timeout Height and a timeout Time Stamp. These attributes of the interface packet will allow the data packet sent to have a lifespan. If the timeout Height or timeout Time Stamp exceeds what is defined in the packet interface when committing it to the ledger than the packer will not be further processed and will be dropped.

5.3.5. Proof Verification (Validity Predicate)

IBC defines an algorithm called validity predicate that is used to verify the state of ledgers. IBC modules that implement validity predicate are called light clients. Using light clients, a module on one ledger can easily verify the state of another ledger.

5.4. Advance Relayer Algorithm of IBC

The relayer of IBC has more features than the relayer in Poly network. Since IBC supports ordered and unordered packet delivery so there are different algorithms defined that must be used to get the packets to be relayed as per the requirement.

These relayers can extract the packet to be relayed using two methods:

5.4.1. Event Based

The relayer should watch the source ledger for events emitted whenever packets are sent, then compose the packet using the data in the event log.

5.4.2. Query Based

The relayer should periodically query the send sequence on the source ledger, and keep the last sequence number relayed, so that any sequences in between the two are packets that need to be queried and then relayed

Packets in an unordered channel can most easily be relayed in an event-based fashion and packets in an ordered channel are relayed using the query based method.

Subsequently, the relayer should check whether the destination ledger has received the packet already by querying for the presence of an acknowledgment at the packet's sequence number, and if one is not yet present the relayer should relay the packet. The relayer in IBC relay more than just data packets. They also relay acknowledgment to the sending ledger. The relayer should watch the destination ledger for events emitted whenever packets are received and acknowledgments are written, then compose the acknowledgment using the data in the event log, check whether the packet commitment still exists on the source ledger (it will be deleted once the acknowledgment is relayed), and if so relay the acknowledgment to the source ledger.

6. Conclusion

This review paper compares the two blockchain bridging solutions, IBC and Poly Market. The paper highlights the similarities between these protocols by explaining the dependence on relayer entities for achieving cross chain communication as well as the difference of features provided by the protocols and the difference in the flow of data during a cross chain communication process.

This paper was written for the sole purpose of providing knowledge for anyone that maybe interested in how each of these protocols work under the hood and for their similarities and differences. The paper also makes a claim that says that all of the future bridging solutions will have to use a relayer entity to achieve cross chain communication. Only time will tell how this claim holds up but there may be new efficient ways that may be engineered in the future.

In conclusion, the paper clearly shows that IBC protocol is more advance, better planned, secure and effective (in theory as the protocol is still being developed at the time of writing this paper) than Poly Network which has had the bigger hack in crypto of $600 mn (which is being returned). All in all these protocols will act as the building blocks and have clearly advanced the bridging solutions by more than what was thought to be capable when we first had the question of how to bridge ledgers.

Acknowledgment

I would like to thank my team at RedChillies for making RedChillies the best place to work in. I would also like to thank my sister Shaili Regmi for critiquing my paper and Tejaswi Sapkota for always being a brother to me.

References

Christopher Goes. (2020). *The Interblockchain Communication Protocol: An Oveview,* Berlin, Germany.

Poly Team. (1955). PolyNetwork: An Interoperability Protocol for Heterogenous Blockchains.

Trans. Roy. Soc.. A247, 529-551, London.

Cite this article as: Apurba Pokharel (2022). Comparing Blockchain Bridges: Poly Network vs. IBC. *International Journal of Cryptocurrency Research,* 2(2), 16-21. doi: 10.51483 / IJCCR.2.2.2022.16-21.