Performance Test and Bottlenecks of IBC (Inter-Blockchain Communication)

Amin Rezaei [1]

Abstract

Fast growth of decentralized technologies and espeically blockchains is starting a trend to transform how finanical and economical concepts happen, and tries to actually democratize the process by removing the centralized entities. There are still a lot of issues to solve in this sector like scalability usually measured as max transactions per second that blockchain can handle. There is a more essential problem laying around on this ecosystem, and it is the absence of connection to outer world and consequently to other blockchains. Solutions like Oracles, reduce the degree of security by providing another network or trusted party as a middleware, which changes the trust assumptions of initial blockchain. IBC (ibc) comes to provide inter-blockchain communication natively through CosmosSDK (csd). In this work we will review performance of IBC in several cases and how scalable it is.

1. Intro

In this section we will briefly discuss the basis knowledge needed to understand for the rest of the paper. Basically discussing what blockchains are, what are the communication challenges and the ways that they are practically been tried to be solved.

1.1. What is a Blockchain?

A Blockchain is a decentralized system that has a single important goal of maintaining a secure ledger across the systems. This ledger is simply a mapping between addresses and their balances and optionally other data based on the usecase of that blockchain. There are various ways of securing this ledger which are known as consensus algorithms. Two popular methods in consensus are Proof of Work (PoW (pow), e.g. Bitcoin network (bit), ETH v1.0 (eth)) and Proof of Stake (PoS (pos), e.g. ETH v2.0). PoW scheme secures the network by posing a computational challenge that upon solving the miner can generate a block that will be verified by other miners on the network. In PoS scheme, the validator roles are introduced in which they deposit funds as stakes, trying to secure the chain with these funds. In each validation round, based on some algorithm a leader generates a block of transactions and gets rewarded. However if a validator actually tries to manipulate the ledger in malicious manner, the network of validators will detect and that validator will risk losing the funds put at stake. All of the consensus algorithms are trying to provide a solution to Byzantine Fault Tolerance (BFT) problem. So if at least two-third of the network is honest, everything will be secure and working.

1.2. Essential Need of Data

Emerging financial capabilities of blockchains through smart contracts that can run arbitrary logic and provide secure and verifiable state changes, brings the need for connection to outside world. Blockchains act as a closed network and there are no connections with outerworld other than smart contract inputs. Obviously, there are several hundreds of blockchains on the market right now. So what happens when a user wants to transfer funds between these blockchains? This is the most simple usecase of outer-world data and needs to be solved. Solutions like Oraclized Bridges try to tackle this problem by introducing a new role as Oracle in these usecases. Oracle (ora) can be a network of nodes or just a trusted party that tries to collect data from outer-world and provide it to the target blockchains. There are two cases here: 1. The oracle is a trusted party: This actually contradicts with the philosophy of decentralization, all the effort is done to provide a trustless system, why trust a 3rd party here? 2. The oracle is a network of decentralized nodes most probably another blockchain. The question here is how secure is that blockchain? If we route through that network, we are most probably lowering security to the security of that network.

[1]University of Waterloo. Correspondence to: Amin Rezaei <a27rezae@uwaterloo.ca>.

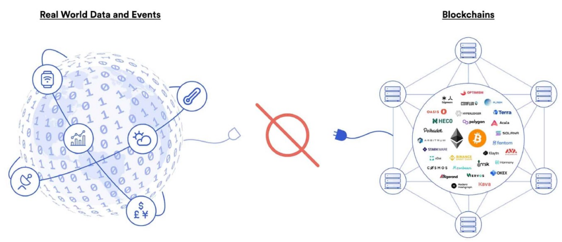

Figure 1. No connectivity between blockchain and outer-world

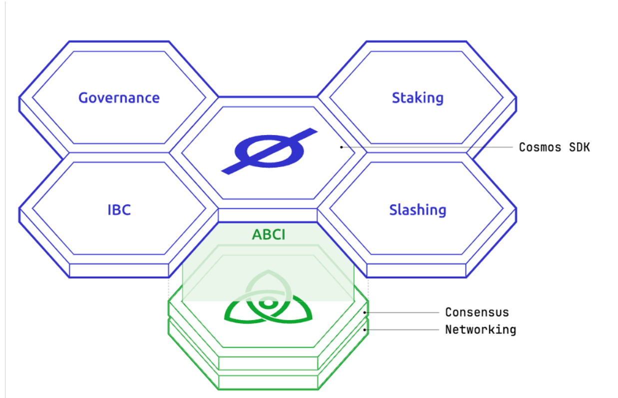

Figure 2. Cosmos SDK

1.3. Are there better alternatives?

One possible solution that comes to mind is that, what if we implement the ability of blockchains talking to each other at core of their system? Although this solution looks promising, the adoption rate and practicality of it is still in question. However, this is the basis idea of Cosmos Blockchain Network (cos). The approach is to provide basis code for blockchains and abstract protocols that can be implemented and populated by each blockchain based on those usecases which is in this case provided by CosmosSDK and CometBFT(cbf)/Tendermint(tdm). Inter-blockchain communication is a standard in this SDK and abstractions, so blockchains made within this ecosystem with provided tooling, will natively support communication among each other using IBC connections and channels.

2. Cosmos Ecosystem

Cosmos aims to be the network of independent application-specific blockchains that are interconnected to each other with connections defined in Inter-Blockchain Communication (IBC) protocol. This is done by providing Cosmos SDK which can be easily utilized to develop custom application logic on top of existing blockchain basis, without worrying about consensus and its security. Figure 2 shows the concept visually.

2.1. Consensus

Consensus is handled using CometBFT a.a.k.a Tendermint which is the consensus algorithm and engine at the core of Cosmos Ecosystem. It also defines the Application BlockChain Interface (ABCI), which abstracts the concepts and allows everybody to develop in any environment they like by just implementing the needed interface for communications between application logic and consensus engine. The algorithm is basically a PoS consensus which is secured by stakes and blocks are commited if 2/3 of network reaches agreement on it.

2.2. IBC

Let's take a look at definition by the official Cosmos website:

> The Inter-Blockchain Communication Protocol (IBC) is an open-source protocol to handle authentication and transport of data between blockchains. IBC allows heterogeneous chains to trustlessly communicate with each other to exchange data, messages, and tokens.

As obvious, the power of IBC is allowing any data, message and funds to be moved trustlessly and natively through these channels which makes it a great candidate in terms of security.

2.3. Relayers

For IBC to work, there is one additional role needed in network, Relayers. The source and destination chains actually have light clients that are like lightweight nodes which enable verifying the consensus state of the other chain and making sure of transaction inclusion. But still, these chains are not connected like two servers/computers in real-life networks we use. Relayers are the entities that scan the networks for IBC transactions and as soon as they spot them, they collect the needed data and proof and submit it to the target chain. There are some fees included with these operations and that's why the relayers should have finanical incentives. Relayers are also the entities that help chains at the first step to be able to establish an IBC channel and light-clients.

2.4. Packet Flows

Figure 3 shows the flow for the case that a packet is sent from A to B and a relayer has successfully transferred the message which is eventually committed to both of the blockchains. However in Figure 4 we see the case that the packet is not delivered to B, and after timeout a relayer triggered the revert of the transaction on the source chain so the funds are restored to the user.

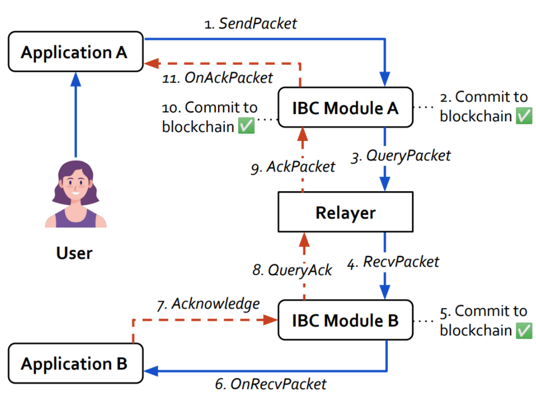

Figure 3. Successful IBC Packet Flow

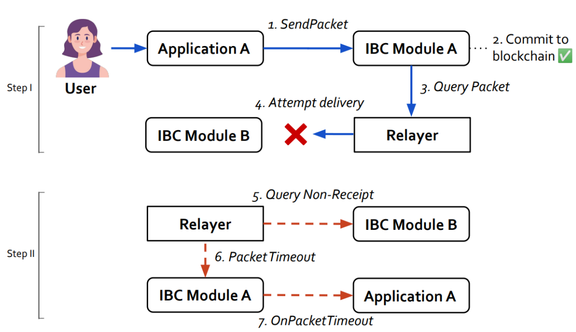

Figure 4. Failed/Timed-Out IBC Packet Flow

3. Challenges

Biggest challenge in the blockchain connectivity aside than Cosmos SDK, is that the connection between blockchains are costy and by emerging number of different blockchains N-to-N connection between them are impossible. Cosmos tries to solve this problem by connecting all blockchains to another blockchain that is called Hub. So every two blockchains that need to talk to each other should perform a two-hop IBC, first sending a message from source to Hub and then from Hub to destination. The process is depicted in Figure 5.

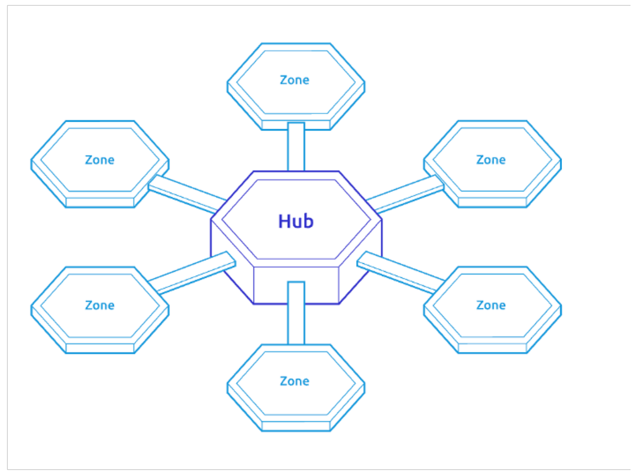

Figure 5. Connectivity Flow in Cosmos Ecosystem

It's obvious that this design doesn't scale and the Hub is acting like a centralized entity that could represent single point of failure. However, the design's future introduces multiple hubs to solve the problem but that would be costy to introduce new hubs to solve the connectivity. Our goal in this work is to show that this design is not optimal and will become a bottleneck.

Another big challenge that we encounter in this process, is the fact that these technologies are pretty new and experimental so the documentation is weak, there are no proper benchmarking suites and spawning decentralized networks is not a straightforward task and requires a lot of resources.

4. Prior Work

In (Chervinski et al., 2023), a benchmark against simple IBC (1-hop) is taken place which we will first review it. Our work is similar to theirs but aims for a diffrent goal (2-hop). Their workflow design is depicted in Figure 6.

As shown in Figure 7, Figure 8 and Figure 9, it's obvious that even on a single hop connection the solution doesn't scale and has issues. So Imagine this, a single hub will be the host to transfer cross-chain messages between thousands of blockchains. It's already obvious that this is practically not scalable. Interesting fact is

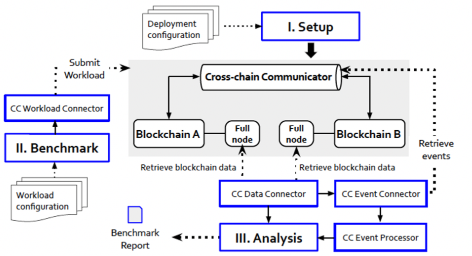

Figure 6. 1-hop Benchmark System Design

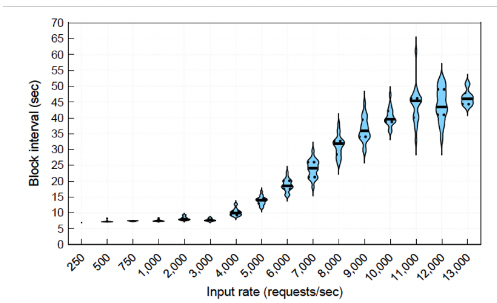

Figure 7. Latency Benchmark on 1-hop

that Figure 9 shows that by adding more relayers for same channel, the network throughput is reduced. An efficient design should've actually promoted this feature and by adding relayers the scalability should've been increased. Otherwise, what if some relayers maliciously try to stall the IBC transactions?

5. Our Design And Results

5.1. Design

We introduce Multi-Chain Setup, it is similar to the prior work's architecture, but tries to benchmark different concept. In our setup instead of 2 chains we have N ones and the benchmark workflow is as follows:

1. Prepare node codes that support IBC packets
2. Setup Configurations and Launch Chains
3. Setup Accounts
4. Start resource monitoring script on servers
5. Begin construction of connections and channels between hub and zones
6. Configure the relayer(s) to be able to relay packets between changes

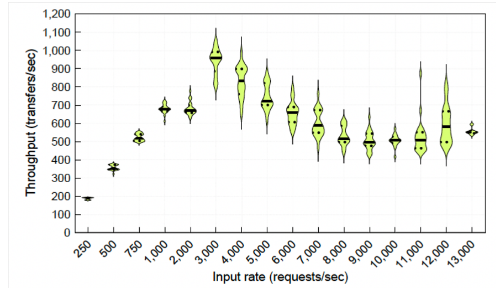

Figure 8. Throughput Benchmark on 2-hop

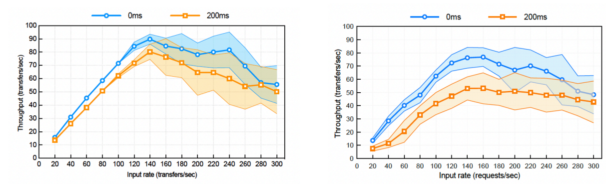

Figure 9. Throughtput with 1 relayer vs 2 relayers

7. Send Two Hop transfer (Chain i -> Hub -> Chain j) requests with varying rates to identify limits of the channels and hub
8. Collect the data:
   (a) TPS and Blockchain specific analysis (Retrieving all stored blocks and analyzing the rates)
   (b) Resource analysis (CPU, Memory, Disk)

Due to hardware limitation, we prepare a simple setup consisting of 5 Blockchains each with 1 Node, and a single relayer trying to deliver the messages. We develop scripts to deploy, prepare channels and start relayer. Eventually, another script starts to flood the network with TXs to cause a bottleneck to appear. We collect the blocks and also process information for further analysis. The main issue we encountered during tests were crashing of relayer code and not allowing us to flood at high speed to test the real limits of the blockchain, so most of our results depict bottleneck at the relayer part. Choice of another stable relayer with higher-end hardware can lead to our main point with the same scripts and setup we provide. All the data mentioned in the setup are collected and available in the repository, but we only depict important results.

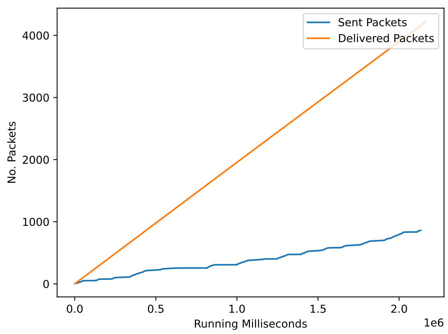

Figure 10. Packets Sent vs Delivered by Relayer

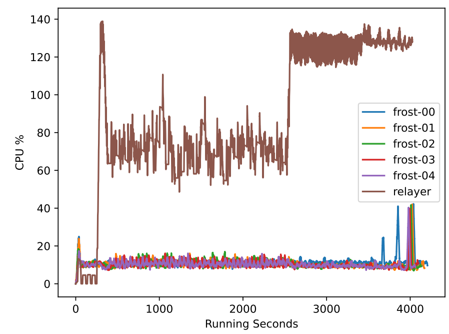

Figure 12. CPU utilization by nodes and relayer (>100% means different threads are active)

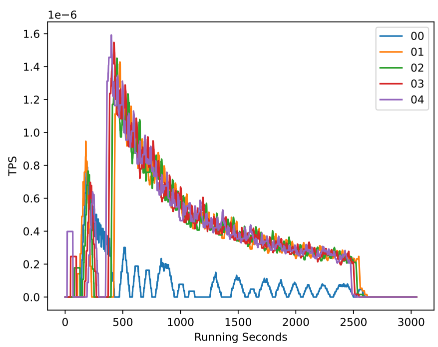

Figure 11. TPS for all chains

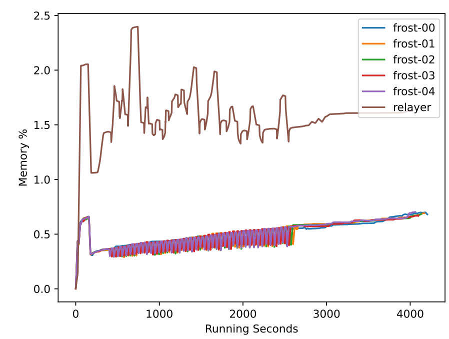

Figure 13. Memory usage by nodes and relayer

5.2. Results

Analyzing the results, we see that obviously the relayer is bottleneck here, the opensource implementation was not completely opensource and we applied a hack so recover from its crashes. Chain frost-00 is the chain acting as a hub and other chains are using it as a transport chain. Also the channel establishment for all chains first time takes about 4 minutes. Assuming that this value is for chains that are actually with one node, Imagine how hard and time taking it would be for chains active with hundreds of validators. The TPS on hub chain is signifcantly lower as significant amount of packets are probably timed out due to delayed delivery. Also, the scaling of chains is in question as TPS drops by time, although the load of messages is constant.

6. Conclusion

Although our setup was not big, but it already showed the bottlenecks of chains and relayers. With a better hardware setup, it's possible to also add more nodes and more blockchains so the test would reflect much more real-world situation. Our proposed solution is to not rely on central Hub, instead utilizing multi-hop transactions that can go through different routes. The blockchains should be connected like in network, and a router can determine a path that packet can travel. All code and data is available on: https://github.com/AminRezaei0x443

References

Bitcoin. https://bitcoin.org/en/.

Cometbft. https://cometbft.com/.

Cosmos network. https://cosmos.network.

Cosmos sdk. https://v1.cosmos.network/sdk.

Ethereum. https://ethereum.org/.

Ibc. https://cosmos.network/ibc/.

What is a blockchain oracle? https://chain.link/education/blockchain-oracles.

Proof of stake. https://en.wikipedia.org/wiki/Proof_of_stake.

Proof of work. https://en.wikipedia.org/wiki/Proof_of_work.

Tendermint. https://tendermint.com/.

Chervinski, J. O., Kreutz, D., Xu, X., and Yu, J. Analyzing the performance of the inter-blockchain communication protocol, 2023.