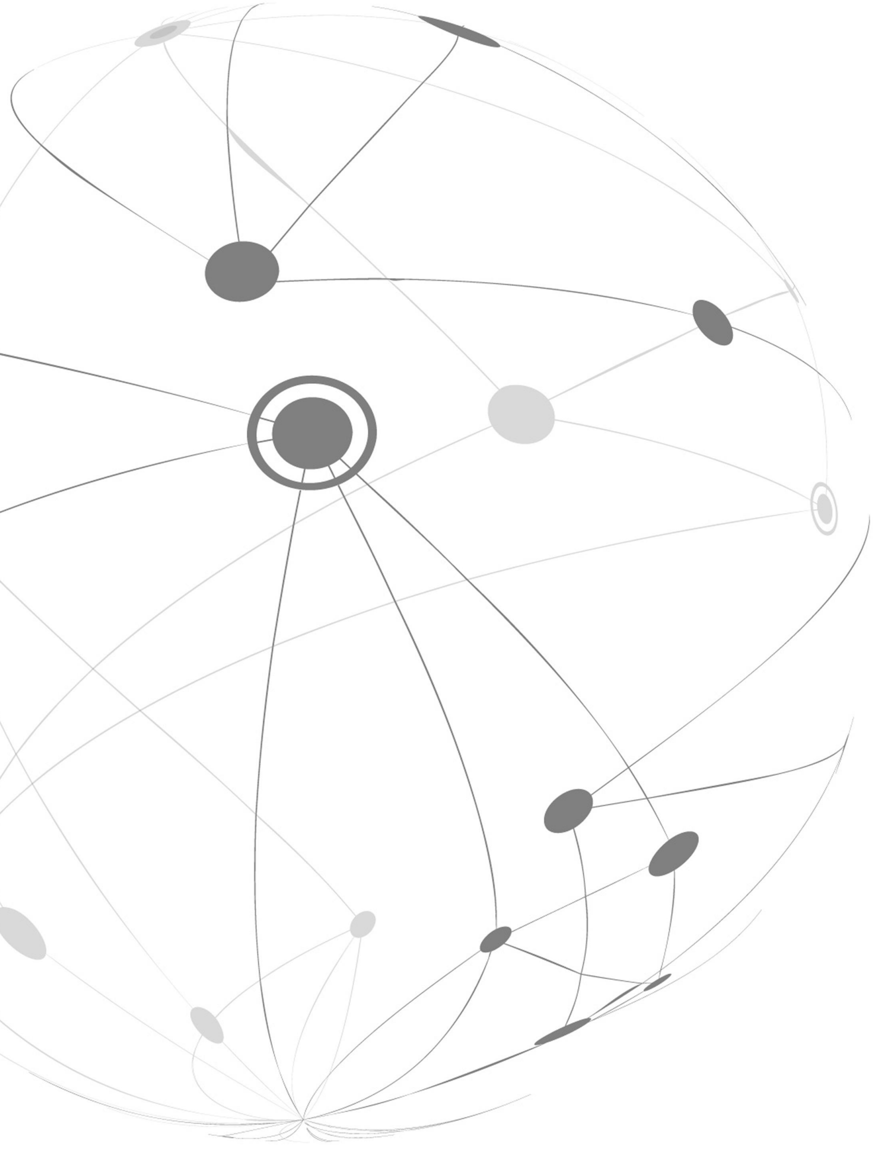
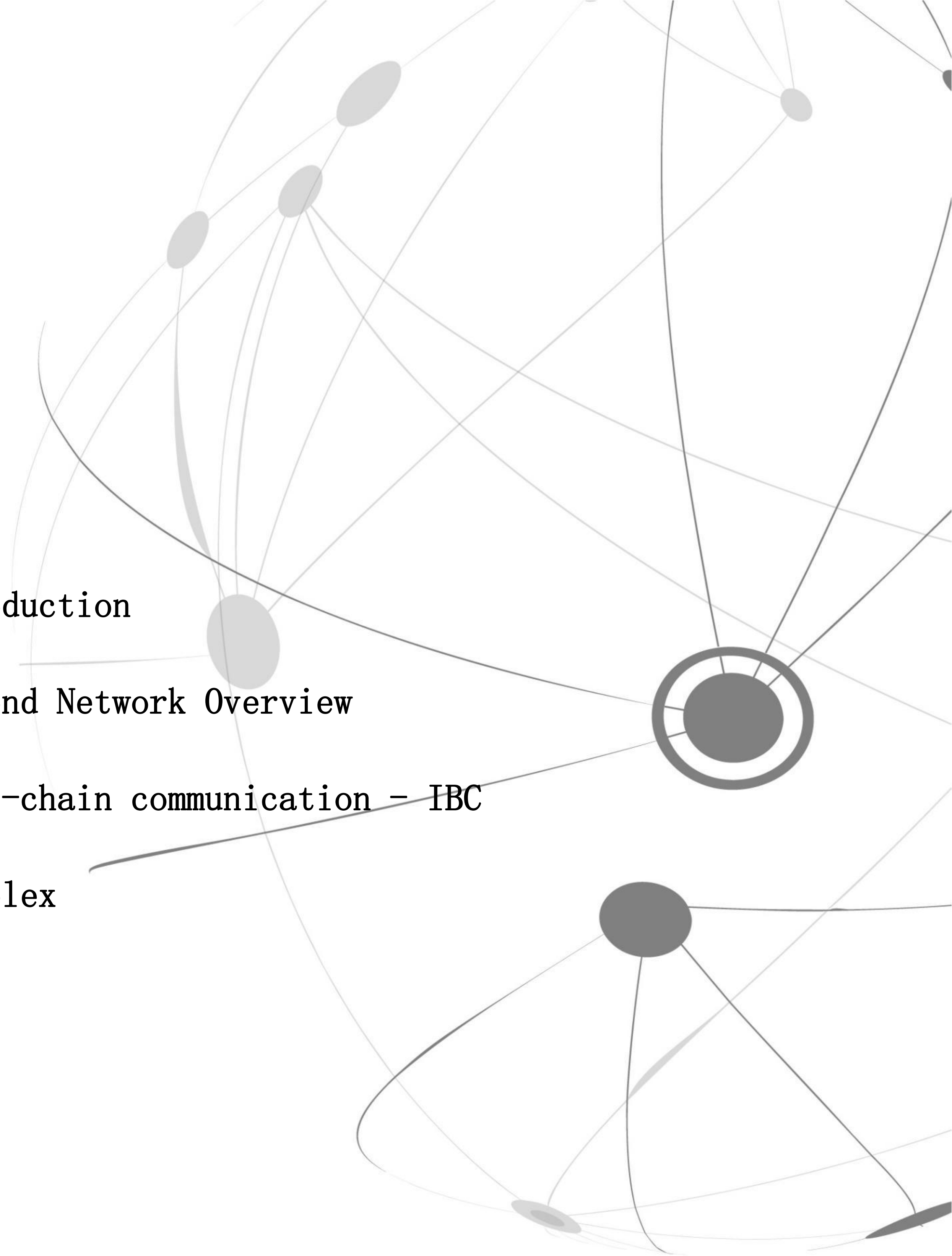
Diamond Network

WHITEPAPER V2.0



DISTRIBUTED LEDGER NETWORK



- Introduction
- Diamond Network Overview
- Cross-chain communication - IBC
- Examplex

Introduction

The success of a series of technologies such as the ecosystem of open source, sharing of decentralized file and public cryptocurrency, has inspired the people and make them understand that decentralized Internet protocols can fundamentally improve the socio-economic infrastructure. We've seen a blockchain application with expertise, such as Bitcoin (cryptocurrency), as well as a public-smart contract platform such as Ethereum, and a myriad of distributed applications based on EVM (Ethereum virtual machine) development.

However, to date, these blockchains have exposed a variety of deficiencies, including low overall energy efficiency, poor performance or, limitations and lack of mature governance mechanisms. In order to expand Bitcoin transaction throughput, many solutions such as Segregated-Witness and BitcoinNG (a new scalable protocol) have been developed. But these vertical scaling solutions are still limited by the capacity of a single physical machine, to ensure complete auditability. Lightning Networks can extend Bitcoin's transaction capacity by partially recording transactions outside the main chain. This method is well suited for micropayment and privacy protection payment channels, but it cannot be applied to more general expansion requirements.

The ideal solution is to allow multiple parallel blockchains to interact while maintaining their security features. It turns out that it is difficult to do this with the proof of workload, but it is not impossible. For example, merge mining allows the parent chain to be reused on the sub-chain while the work is completed. But the transaction must be verified sequentially by each node. And if most of the hashing power on the parent chain does not actively conduct the merge mining on the sub-chains, they are vulnerable to be attacked.

Here we will introduce the Diamond Network, a new blockchain network architecture that solves all of these problems. The Diamond Network is a network composed of many independent blockchains called "partitions." Partitioning runs under the support of BPOS, which is a Byzantine fault-tolerant security consensus engine with high-performance, consistent features and be able to stop malicious vandals under a strict fork-and-fork mechanism. BPOS's Byzantine fault-tolerant consensus algorithm is well suited for extending the public blockchain under the Proof of Entitlement (PoS) mechanism. Using the Blockchains of other consensus models, including Ethereum which is similar to POS and Bitcoin, can also be connected by the Diamond Network network using adaptive partitioning.

The first partition of the Diamond Network is called the Diamond Network hub. The Diamond Network Hub is a multi-asset equity proof cryptocurrency network that adapts and upgrades the network through a simple governance mechanism. In addition, the Diamond Network hub can be extended by hooking to other partitions.

The hub and partitions of the Diamond Network network can communicate via the Inter-Block Inter-Chain Communication (IBC) protocol, which is the Virtual Unser Datagram Protocol (UDP) or Transmission Control Protocol (TCP) for the blockchain. Tokens can be safely and quickly transferred from one partition to another without having to have exchange liquidity between the two partitions. Instead, all cross-partition token transfers go through the Diamond Network hub to track the total amount of tokens held in each partition. This hub will isolate each partition from other failed partitions. Because everyone can connect the new partition to the Diamond Network hub, the partition will be backward compatible with the new blockchain technology.

Interoperability between blockchains can be used to achieve with Diamond Network. This is a potentially valuable Internet where assets are released and controlled by different verifiers and seamlessly transfer and trade across cross-chain assets without relying on third parties that need to be trusted.

In this section, we will explain the BPOS consensus protocol and the interface used to build its application.

In the classic Byzantine fault-tolerant algorithm, each node has the same weight. At BPOS, nodes have different numbers of (non-negative) voting rights, and those with a significant amount of voting rights are called verifiers. The verifier participates in the consensus agreement by encrypting the signature, voting, or voting for the next block.

Note: Scores like 2⁄3 and 1⁄3 refer to the total voting weight, not the total verifiers, unless all verifiers have the same weight. And “>2⁄3” means "more than 2⁄3", “ ⩾ 1⁄3” means "1⁄3 or more".

CONSENSUS

BPOS is a partially synchronized Byzantine fault-tolerant consensus protocol, which is derived from the DLS consensus algorithm. BPOS is known for its simplicity, performance and forked accountability. The protocol requires that this set of verifiers be fixed and well known, and that each verifier has its public key authentication identity. These verifiers attempt to reach consensus on a block at the same time, and these blocks are a series of transaction records. The consensus of each block takes turns, and each turn will have a leader or proposer, who will initiate the block. The verifier then decides whether to accept the block in stages or whether to proceed to the next round to vote. Each round of proposers will choose from the list of verifiers in accordance with their voting rights.

BPOS uses optimal Byzantine fault tolerance using most voting (more than two-thirds) and locking mechanisms to ensure its security. These can guarantee:

If a vandal wants to cause a security issue, it must have more than one-third of the voting rights and submit more than two-third values.

If a group of verifiers successfully compromises security or has tried to do so, they will be identified by the protocol. The protocol includes voting on conflicting blocks and broadcasting those questionable votes.

In addition to its superior security, BPOS has outstanding performance. Taking a commercial cloud platform as an example, the BPOS consensus is based on 64-bit nodes distributed across seven data centers on five continents. It can process thousands of transactions per second with an order submission delay of 1-2 seconds. It is worth noting that even in extremely hostile environments, such as verifiers crash or broadcasting maliciously slammed votes, this performance of more than a thousand transactions per second can be maintained.

VERIFIER

In the classic Byzantine fault-tolerant algorithm, each node has the same weight. At BPOS, nodes have different numbers of (non-negative) voting rights, and those with a significant amount of voting rights are called verifiers. The verifier participates in the consensus agreement by encrypting the signature, voting, or voting for the next block.

The verifier's voting rights are determined from the outset, or the modification of voting rights is determined by the blockchain based on the application. For example, in a proof of interest application like the Diamond Network hub, voting rights can be determined by the number of tokens tied to the deposit.

LIGHT CLIENT

The main benefit of the BPOS consensus algorithm is its secure and easy client, making it an ideal choice for mobile and IoT use cases. The Bitcoin light client must synchronize the chain of block headers and find the one with the most proof of workload, while the BPOS light client only needs to be consistent with the changes in the verification group, and then simply verify the >⅔ pre-submitted in the latest block to determine the latest situation.

This authentication mechanism of simple light client also enables communication between blockchains.

PREVENT ATTACKS

BPOS has a variety of defenses to prevent obvious attacks, such as remote and non-interested double-spend attacks and censorship.

ABCI

The BPOS consensus algorithm is implemented in a program called BPOS. This program is a consensus engine independent from the application. It can transform any identified black box application into a distributed, replicable blockchain. BPOS can connect to other blockchain applications through the Application Blockchain Interface (ABCI). Moreover, the Application Blockchain Interface (ABCI) allows blockchain applications to be implemented in any language, not just to wirte the language used by this consensus engine. In addition, the Application Blockchain Interface (ABCI) also makes it possible to exchange the consensus layer of any existing blockchain stack.

We compared it to the well-known cryptocurrency Bitcoin. In the Bitcoin cryptocurrant blockchain, each node maintains a fully audited UTXO (Unused Transaction Output) database. If you want to create a Bitcoin-like system based on the Application Blockchain Interface (ABCI), then BPOS can do the following:

Share blocks and transactions between nodes
Create a canonical or unchangeable transaction order (blockchain)

At the same time, ABCI applications can also:

Maintain the UTXO database
Verify the cryptographic signature of the transaction
Prevent the occurrence of non-existent balances from being traded
Allow customers to access the UTXO database

Overview of Diamond Network

The Diamond Network is an independent parallel blockchain network in which each blockchain runs through a classical Byzantine fault-tolerant consensus algorithm such as BPOS.

The first blockchain in the network will be the Diamond Network hub. The Diamond Network hub connects many other blockchains (or partitions) through a new blockchain communication protocol. The Diamond Network hub tracks countless types of tokens and records the total number of tokens in each connected partition. Tokens can be safely and quickly moved from one partition to another without currency exchange between them, as token transfers between all partitions pass through the Diamond Network hub.

This architecture addresses many issues that today's blockchain domain is facing, including the interoperability, scalability and the ability to seamlessly upgrade of application. For example, partitions derived from Bitcoind, Go-Ethereum, CryptoNote, ZCash or other blockchain systems can be anchored to the Diamond Network hub. These partitions allow the Diamond Network to scale indefinitely to meet the needs of global transactions. In addition, the partition is also fully applicable to distributed exchanges, and the exchange also supports partition operations.

The Diamond Network is more than just a single distributed ledger, and the Diamond Network hub is not the center of a closed courtyard or universe. We are designing a set of protocols for the open network of distributed ledgers, which will be based on cryptography, robust economics, consensus theory, transparency and accountability principles, and become the new foundation for the future financial system.

GOVERNANCE

A distributed public account book should have a set of articles of association and governance systems. Bitcoin relies on the Bitcoin Foundation and mining to collaborate to update. But this is a slow-responding governance system. Ethereum is using hard forks to ETH and ETC to solve The DAO hackers, mainly because no social contract or mechanism was set up to make such decisions.

The verifier and principal of the Diamond Network hub can vote on the proposal, changing the system parameters that are pre-set by default (such as block transfer fee limits), collaboratively updating, and revising the readable articles of association to govern the hub system of Diamond Network. The articles of association allow stakeholders to come together to address issues such as theft and vulnerabilities (such as The DAO incident) and to get a solution faster and more specific solutions.

Each division can also develop its own set of articles of association and governance mechanisms. For example, the articles of association of the Diamond Network hub can be set to enforce the unchangeable nature of the hub (cannot be rolled back, except for vulnerabilities generated by Diamond Network hub nodes), and each partition can set its own rollback policy.

The Diamond Network enables interoperability between different partitions of the system, giving customers the freedom and potential to experiment without the need for a license (new technology).

BPOS- BYZANTINE FAULT TOLERANCE

The Diamond Network hub is the first public blockchain in the Diamond Network that runs through BPOS's Byzantine consensus algorithm. The BPOS open source project was created in 2014 to address the speed, scalability and environmental issues of the Bitcoin workload proof consensus algorithm. By adopting and improving the proven Byzantine algorithm (developed at the Massachusetts Institute of Technology in 1988), BPOS became the first team to demonstrate the cryptocurrency of POS in a concept. This mechanism can solve the "no interest" problem faced by NXT and BitShares, the first generation of equity proof cryptocurrency.

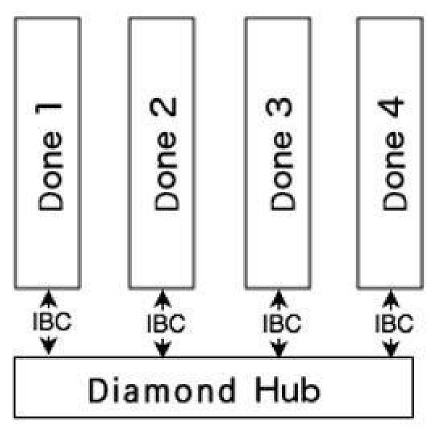
Today, all Bitcoin mobile wallets virtually use reliable servers for transaction verification. This is because the workload proof mechanism needs to be confirmed multiple times before the transaction is deemed to be irreversible. Dual payment attacks have also appeared in services such as Coinbase.

Unlike other blockchain consensus systems, BPOS provides instant, provably secure mobile client payment verification. Because BPOS is designed to be completely non-forked, the mobile wallet can receive transaction confirmations in real time, thus truly achieving a trusted payment method on the smartphone. This also greatly affects IoT applications.

The verifier role in the Diamond Network is similar to the Bitcoin miners, but they use cryptographic signatures to vote. The verifier is a secure machine dedicated to submitting blocks. Non-verifiers can entrust an equity token to any verifier to earn a certain block fee and reward. But if the verifier is hacked or violates the agreement, the token will be at risk of being punished (cut). The provable security mechanism of the BPOS Byzantine Consensus and the collateral guarantees of stakeholders (verifiers and principals) provide provable and quantifiable security for nodes and even light clients.

Hub and Partition

Here we will describe a new decentralized and scalable model. The Diamond Network runs numerous blockchains through the BPOS mechanism. Though the goal of the existing proposal is to create a "single blockchain" that includes all trading orders worldwide, the Diamond Network allows many blockchains to remain interoperable while running in parallel. On this basis, the Diamond Network hub manages a number of independent blockchains (sometimes called "fragments", with reference to "fragments" from well-known database extension techniques). Fragments on the hub will continuously submit the latest blocks, which allow the hub to synchronize the state of each partition. Similarly, each partition will be consistent with the state of the hub (although the partitions will not be synchronized with each other unless indirectly through the hub). The message is proven to be accepted and sent by issuing a Merkel certificate, to pass the message from one partition to another. This mechanism is called "inter-blockchain communication" or simply "IBC" mechanism.



Any partition can be a hub to build acyclic chart. But for the sake of clarity, we only describe such a simple configuration with only one hub and many non-hub partitions.

HUB

The Diamond Network hub is a blockchain that carries a variety of distributed ledger assets, where tokens can be held by individuals or partitions themselves. These tokens can be transferred from one partition to another via a special IBC packet, e.g. "coin packet" is transferred from one partition to another. The hub is responsible for keeping the total amount of tokens in each zone unchanged. IBC token packet transaction must be executed by the sender, hub, and block recipient.

Because the Diamond Network hub acts as a central token book throughout the system, its security is extremely important. Although each partition may be a BPOS blockchain – just pass four, (or fewer verifiers to ensure security without the Byzantine fault tolerance consensus). But the Diamond Network hub must be secured by a global decentralized verification team, and this verification team must be able to withstand the most serious attacks, such as regional network points.

PARTITION

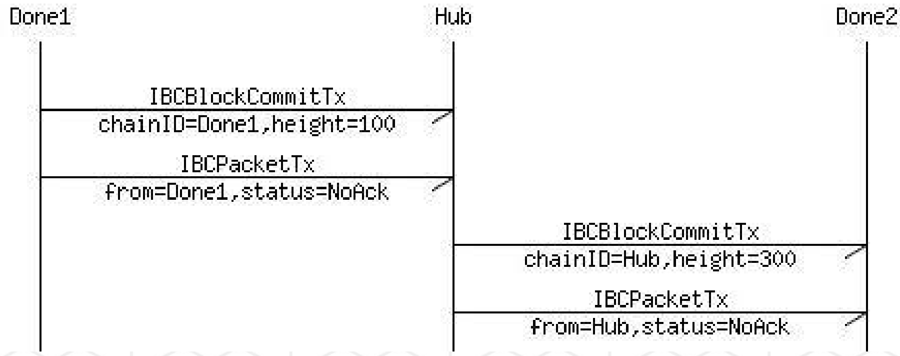
The partition of Diamond Network is an independent blockchain that enables IBC message exchange with the Diamond Network hub. From the perspective of hub, a partition is a multi-asset, dynamic membership multi-signature account that can be used to send and receive tokens through IBC packets. Just like a cryptocurrency account, a partition cannot transfer tokens that exceed its holdings, but can receive tokens from other people who own tokens. Partitions may be designated as "sources" of one or more tokens, giving them the power to increase the supply of tokens.

The tokens of the Diamond Network hub can be used as a chip that the block verifier connects to the hub. Although under the BPOS fork liability system, a double payment attack on a partition can result in a reduction in the number of tokens, if more than 2⁄3 of the votes in the partition have been occurred the Byzantine issue, the partition can be submitted to an invalid state. The Diamond Network hub does not validate or execute transactions submitted to other partitions, so it is the responsibility of the user to send tokens to a reliable partition. In the future, the management system of the Diamond Network hub may solve the partition failure problem by improving the proposal. For example, when an attack is detected, the token transfer initiated by some partitions (or all partitions) will be suspended to achieve an emergency disconnect (i.e., temporarily suspend the token transfer).

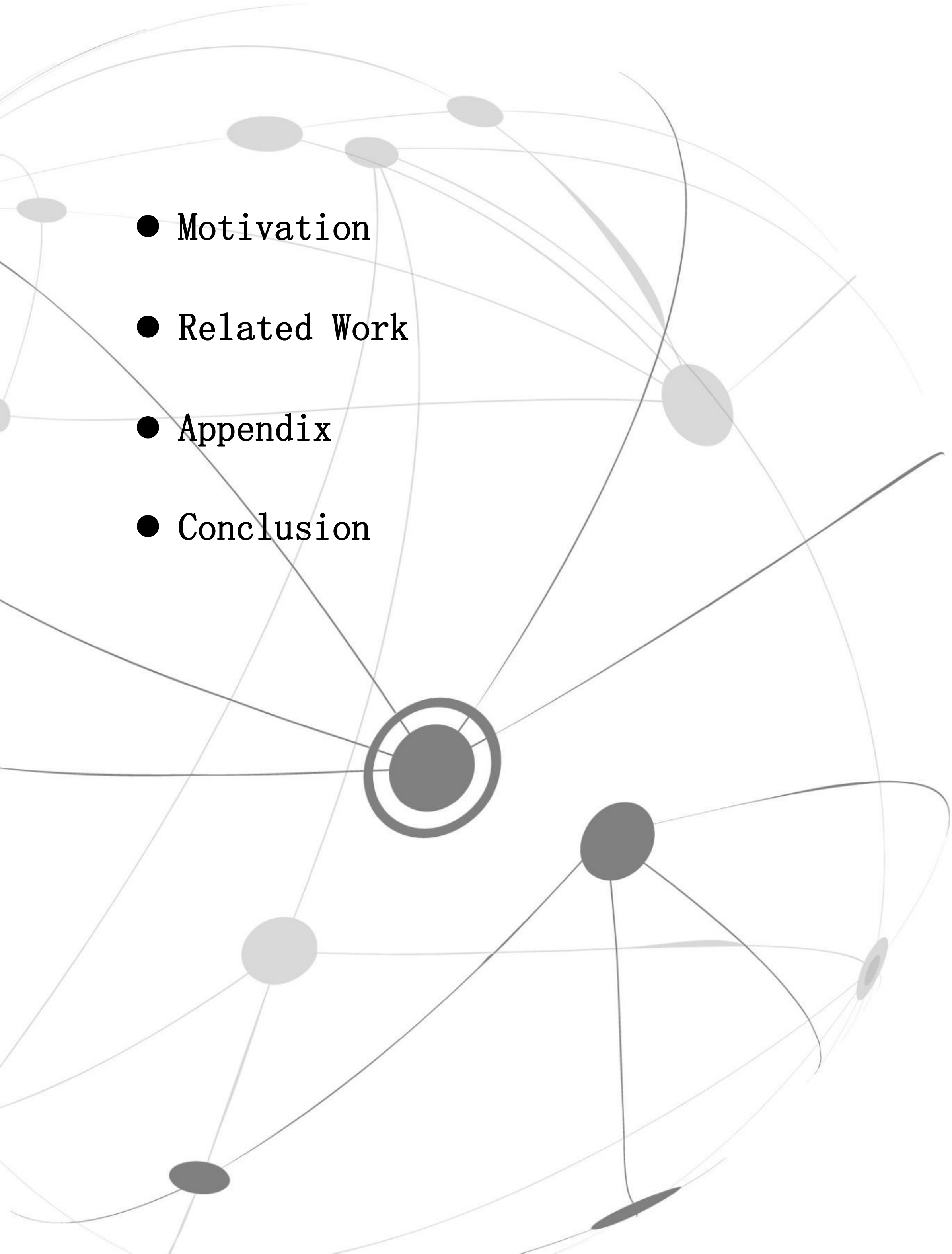
Cross-chain Communication - IBC

Now let's introduce the method of communication between the hub and the partition. If there are now three blockchains, "Partition 1", "Partition 2", and "Hub", we want "Partition 1" to generate a packet and send it to "Partition 2" via "Hub". In order to transfer a packet from one blockchain to another, a certificate is required to be issued on the receiver's blockchain to clarify that the sender has initiated a packet to the specified destination. The proof that the receiver wants to verify must be consistent with the sender's block header. This mechanism is similar to the mechanism used in the side chain. It requires two interacting chains to "know" the other party by transmitting the proof data element (transaction) in both directions.
The IBC protocol can be naturally defined as the use of two transactions: one is the IBCBlockDommitTx transaction, which allows the blockchain to prove its latest block hash to any observer; the other is the IBCPacketTx transaction, which can prove that a packet is actually sent by the sender's application to the hash of the most recent block by Merkle-proof.

By separating the IBC mechanism into two separate transactions, which are IBCBlockDommitTx transaction and the IBCPacketTx transaction. We can let the receiver's local fee market mechanism decide which packets to be recognized, while at the same time ensuring the sender's complete freedom, letting it decide for itself the number of packets that can be transmitted.



In the above case, in order to update the block hash of "Partition 1" on the "hub" (or the block hash of the "hub" on "Partition 2"), the block hash of "Partition 1" of the IBCBlockDommitTx transaction must be posted to the "hub" (or the hash value of the "hub" block of the transaction is posted to "Partition 2").



- Motivation
- Related Work
- Appendix
- Conclusion

Examples

DISTRIBUTED EXCHANGE

Bitcoin uses a lot of replication to increase the security of distributed ledgers. In a similar way, we can run an exchange on the blockchain to reduce the likelihood of internal and external attacks. We call this as decentralized exchange.

Today, the decentralized exchanges is based on "cross-chain atomic transactions" (AXC transactions) according to cryptocurrency community. Through the AXC transaction, two users on two different chains can initiate two transfer transactions. The transaction is either submitted together on two books, or both books are not executed (i.e. the atomicity of the transaction). For example, two users can realize the transactions between Bitcoin and Ethereum (or any two tokens on different books) through AXC transactions, even though Bitcoin and Ethereum's blockchains are not connected to each other. Users of exchange in the AXC trading model neither need to trust each other nor rely on the transaction matching service. The drawback is that the transaction can be made only when both parties are online at the same time.

Another decentralized exchange is a distributed exchange with a separate blockchain that replicates in large numbers. Users of such exchanges can submit orders with limit price and close their computers. And transactions can be performed while the user is offline. The blockchain will represent the trader to complete the match and trade.

A centralized exchange can build a trading account with a large volume of trading with limit price to attract more traders. In the field of exchanges, liquidity triggers more liquidity. So the network effect of the exchange business has become more apparent. Currently, the cryptocurrency exchange Poloniex ranks first with a transaction volume of $20 million per 24 hours. And Bitfinex ranks second with a transaction volume of $5 million per 24 hours. Under this powerful network effect, the volume of AXC-based decentralized exchanges is unlikely to exceed the centralized exchanges. If a decentralized exchange wants to compete with a centralized exchange, it is required to support the operation of deep trading account composed of orders with limit price. It can be only achieved by decentralized exchanges based on blockchain.

The fast trade execution provided by BPOS is another big advantage. Diamond Network's internal network can quickly and accurately determine the finality without sacrificing consistency, to complete the transaction quickly – for both trade order transactions and IBC (cross-blockchain communication) tokens and other networks transaction.

In summary, according to the existing situation of cryptocurrency exchange, a major application of the Diamond Network is the decentralized exchange (called Diamond Network DEX). Its trading throughput and commissioning delays are comparable to those of centralized trading. Traders can submit orders with limit price when the parties are offline. And, based on BPOS, the

Diamond Network hub and IBC, traders can quickly complete funds in and out of exchanges and other networks.

AS A HOOK TO OTHER CRYPTOCURRENCIES

A privileged partition can be a token source hooked to other cryptocurrencies. This hook is similar to the relationship between the Diamond Network hub and the partition. Both must update each other's latest blockchain to verify that the token has been transferred from one party to the other. The "bridge partition" hooked on the Diamond Network network is synchronized with the center and other cryptocurrencies. This indirect "bridge partition" can keep the hub logic simple. And it is not necessary to understand other chain consensus strategies, such as the Bitcoin workload to prove the mining mechanism.

Full Liability System of Hooked Area

The risk of such linked contracts is that a group of malicious verifier may appear. If the Byzantine voting rights exceed 1⁄3, it will cause a fork, which is to pick up the Ethercoin from the Ethereum bridge-joint contract, and also keep the hooks in the bridged partition unchanged. Even if the Byzantine voting rights exceed 2⁄3, someone may directly steal the Ethercoin by sending the Ethercoin to the account in the bridge-joint contract by breaking the bridging logic from the original bridging partition.

If this bridging method is completely designed as a responsibility system, it is possible to solve this problem. For example, all IBC packages at the hub and starting point may need to be approved by the bridge partition first, that is, all state transitions of the bridge partition will be validated by the bridge-joint contract in the hub or starting point. The hub and starting point should allow the verifier of the bridging partition to provide collateral, and the contract's token transfer needs to be delayed (and the collateral unbinding time is long enough) so that the individual auditor has time to initiate any question. We will open up the design description and implementation of this system as a proposal for future Diamond Network improvement, so that the management system of the Diamond Network hub will be approved.

Send tokens to the Diamond Network hub

The verifier of each hooked bridge partition runs a special ABCI bridging application on the blockchain based on the BPOS formula, but also runs a "full node" of the original blockchain.

When the new block is dug out in the original blockchain, the verifier of bridge zone will agree on the local perspective by signing and sharing the hints of the starting point blockchain. When a bridged partition receives payment from the original blockchain (such as a sufficient number of acknowledgments on the chain of PoW mechanisms such as Ethereum or Bitcoin), a balance with the corresponding account should be created on the bridged partition.

In the case of Ethereum, the bridge partition can share the same verifier with the Diamond Network hub. In the Ethereum aspect (original blockchain), it is allowed that a bridge engagement will be sent by the Ethernet owner to the bridge of the Ethereum's bridging partition. Once the bridge is engaged and the Ethereum is received, the Ethereum cannot be withdrawn unless the corresponding IBC packet is received from the bridged partition. The bridge joins the verification group following the bridge partition, which may be the same as the verifier group of the Diamond Network hub.

In the case of Bitcoin, the concepts are similar except that instead of a bridge-joint contract. Each UTXO will be limited by a threshold multi-signature P2SH database. Due to the limitations of the P2SH system, the signer cannot be the same as the verifier group of the Diamond Network hub.

Picking Up Tokens from the Diamond Network Hub

The Ethereum on the bridged partition ("Bridged-joint Ethereum") can be transferred in and out between the hubs. And after the transfer to a specific Ethereum pickup address, the "bridged-joint Ethereum" that was transferred out is completely deleted. An IBC message can prove the transaction on the bridged partition, and this message will be posted to the Ethereum bridge-joint contarct so that the Ethereum is picked up.

As far as Bitcoin is concerned, the rigorous trading script system makes the image conversion mechanism of IBC currency difficult to be implemented. Each UTXO has its own specific script, so each UTXO must be migrated to the new UTXO when the Bitcoin compliance signer changes. One solution is to compress and decompress UTXO-set as required to keep the total number of UTXOs down.

MULTI-PURPOSE INTEGRATION

The Diamond Network partition can run any application logic. The application is set when the partition is created, and can be updated continuously by the administrator. This flexibility allows the Diamond Network partition to act as a hook carrier for other cryptocurrencies, such as Ethereum or Bitcoin. And it can also be hooked to derivatives of these blockchains, and will be distinguished in the verification process and initial allocation by using the same code base. This allows multiple existing cryptocurrency frameworks to work, such as Ethereum, Zerocash, Bitcoin, CryptoNote, etc., combined with BPOS to become a better-performing consensus engine in a common network, providing more interaction opportunities between platforms. In addition, as a multi-asset blockchain, each transaction may contain multiple input and output items, each of which can be any token, making the Diamond Network directly a decentralized exchange. Of course, assuming the trade orders are matched through other platforms. The alternative is to have the partition as a distributed fault-tolerant exchange (including the purchase and sale accounts), which is a rigorous improvement over the centralized cryptocurrency exchange – a case that the current transaction is often attacked in the past.

Partitioning can also be used as a block-chain enterprise and government system, a specific service that was originally run by one or more organizations is now running on a partition as an ABCI application, so that to maintain the security and interactivity of the public Diamond Network, without giving up control of the underlying services. Therefore, the Diamond Network can provide an excellent operating environment for those who want to use blockchain technology and are not willing to give up control to distributed third parties.

EXPANSION OF ETHEREUM

As we all know, the expansion is a problem that has been plaguing Ethereum. Currently the Ethereum node processes every transaction on the node and stores all state references.

The BPOS submission block is faster than the Ethereum workload, so Ethereum virtual machine partitions driven by BPOS consensus and running on bridging Ether can enhance the performance of the Taifang blockchain. In addition, although the Diamond Network hub and IBC parcel mechanism cannot implement contract logic per second, it can be used to coordinate token circulation between Ethereum contracts in different partition, and lay a foundation for expansion of Ethereum based on token by fragmentation.

MITIGATING NETWORK PARTITIONING PROBLEMS

Some people think that the consensus algorithm like BPOS that supports consistency has a big problem that, network partitioning will result in no more partitions having more than 2⁄3 of voting rights (such as more than 1⁄3 voting rights are offline), which will break the consensus. The Diamond Network architecture can alleviate this problem. It can use the global center. At the same time, each partition implements the regional autonomy, and then the voting rights of each partition are allocated according to the normal geographical location. For example, the general paradigm may be for individual cities or regions, allowing them to run their own partitions while sharing a common hub (such as the Diamond Network hub). And during the interruption caused by the temporary network partition, it can continue to maintain the regional autonomy activities. Note that in the process of designing a robust federated fault-tolerant system, you can consider the characteristics of real geographic, political, and network topologies.

FEDERATED NAME RESOLUTION SYSTEM

NameCoin is one of the first blockchains to try to solve name resolution problems with Bitcoin technology. However, there are some shortcomings in this program.

For example, we can verify @satoshi by Namecoin, which is registered with a specific public key at some point in the past. However, it is not known whether the public key has been updated, unless download all before the name was last updated. This is due to the limitations of the Merkel model in the Bitcoin UTXO trading model, where only transactions (rather than variable application states) are added to the block hash with Merkel. It allows us to use the update to prove the existence of the name, not non-existent. Therefore, we must rely on the full node to clarify the most recent value of this name, or spend a lot of resources to download the entire blockchain.

The independence of the proof of work will still lead to problems in the verification of light Client. The Light Client must download a full backup of all block headers in the blockchain (or at least all block headers updated from their last name). It means that bandwidth needs to be linearly extended over time. In addition, in the workload proof system, the name change on the blockchain needs to be verified by additional workload verification verification, which may take an hour on Bitcoin.

With BPOS, we only need the block hash signed by the quorum verifier (by voting rights) and the Merkel certificate of the current value associated with the name. This makes it possible to verify the value of the simple, fast and secure light passenger name.

In the Diamond Network, we can take advantage of this concept and extend it. Each name registration on the Diamond Network can have a related top-level domain name (TLD), such as ".com" or ".org", and each name registration partition has its own management and registration rules.

Motivation

LIMIT ON THE NUMBER OF VERIFIERS

Unlike Bitcoin or other workload proof blockchains, the BPOS blockchain slows down as the number of verifiers increases due to increased communication complexity. Fortunately, we can support enough verifiers to implement a reliable globally distributed blockchain with very fast transaction confirmation time. And as bandwidth, storage, and parallel computing capacity are increasing, we will be able to support more verifiers in the future.

On Creation Day, the maximum number of verifiers will be set to 100. This number will increase by 13% for 10 years and eventually to 300.

BECOMING THE VERIFIER AFTER CREATION DATE

Token holders can become verifiers by signing and submitting BondTx transactions. The amount of tokens collateralized cannot be zero. Anyone can become a verifier at any time, unless the current number of verifier groups exceeds the maximum. In this case, the transaction is valid only if the number of tokens held is greater than the minimum number of valid tokens in the existing verifier, where the valid token includes the entrusted token. When a new verifier replaces an existing verifier in this manner, the existing verifier will be offline and all of its tokens and delegated tokens will go into the unbound state.

PUNISHMENT FOR THE VERIFIER

Any verifier who intentionally or unintentionally deviates from the accreditation agreement must be imposed certain penalties. Some evidence is immediately admissible, such as a double signature at the same height and round, or a violation of the “pre-voting lock” (the rules of the BPOS consensus protocol). Such evidence will cause the verifier to lose its good reputation, and its bound tokens and the proportion of shares in the reserve pool (collectively referred to as “equity”) will be significantly reduced.

Sometimes, the verifier will not be available due to a regional network outage, power failure, or other reasons. If the verifier's commit vote is not included in the blockchain are more than ValidatorTimeoutMaxAbsent times in the ValidatorTimeoutWindow block at any point in the past, the verifier will go offline and reduce the euiqty of ValidatorTimeoutPenalty (default 1%).

Some “malicious” behaviors did not produce clear evidence on the blockchain. In these cases, if there is a majority of consensus, the verifier can coordinate outside the band to force these malicious verifiers to time out.

If the Diamond Network hub has a suspension because more than 1⁄3 of the voting rights are offline, or more than 1⁄3 of the voting rights enter the malicious behavior of the blockchain after being reviewed, then the hub must be recovered with the hard fork reorganization protocol. (See "Forking and Reviewing Attacks" for details).

TRANSACTION FEE

Verifiers of Diamond Network hub can accept any kind of token or combination as a transaction fee. Each verifier can set the exchange rate and select the transaction it wants. As long as it does not exceed BlockGasLimit, every ValidatorPayoutPeriod (default is 1 hour) will be allocated according to the proportion of tokens bounded by the stakeholders.

Among the transaction fees charged, ReserveTax (default 2%) will be deposited in the reserve pool to increase the reserve and increase the security and value of the Diamond Network hub. These funds can also be allocated in accordance with the decisions of the governance system.

Token holders who delegate voting rights to other verifiers will pay a commission to the client, and this fee can be set by each verifier.

MOTIVATING HACKER

The security of the Diamond Network hub depends on the security of the underlying verifier and the client's delegation choice. To encourage discovery of vulnerabilities in early reporting, the Diamond Network hub encouraged hackers to post successful vulnerabilities through ReportHackTx transactions, saying, "This verifier was compromised, please send the bonus to this address." In this case, the verifier and the principal will be hung and idle. Each person's

HackPunishmentRatio (default in 5%) token will be cut, and the HackRewardRatio (default in 5%) token will be sent to the hacker's bounty address as a reward. The verifier must use its backup key to recover the remaining tokens.

In order to prevent this feature from being abused to transfer unauthorized tokens, the ratio of tokens (authorized and unlicensed) will remain unchanged before and after ReportHackTx, and the hacker's bounty will include some unauthorized tokens (if there is).

SPECIFICATION OF GOVERNANCE

The Diamond Network hub is managed by a distributed organization and requires a clear governance mechanism to coordinate changes to the blockchain, such as system parameter variables, as well as software upgrades and constitutional revisions.

All verifiers are responsible for voting on all proposals. Failure to vote on the proposal in a timely manner will result in the verifier being automatically deactivated for a period of time. This time is called AbsenteeismPenaltyPeriod (default as 1 week).

The principal automatically inherits the voting rights of the verifiers it delegates. This vote can be manually overwritten. Unbound tokens have no voting rights.

Each proposal requires a margin for the token, which may be a combination of one or more tokens (including tokens). For each proposal, the voter can vote to remove the deposit. If more than half of the voters choose to take the deposit (for example, because the proposal is spam), then the deposit will be deposited in the reserve pool, except for the burning token s.

For each proposal, voters can choose from the following options:

- Agree
- Strongly agree
- Oppose
- Strongly oppose
- Abstain

Decision of adopting (or not adopting) a proposal requires a strict majority vote “agree” or “strongly agree” (or “oppose” and “strongly oppose”), but more than one-third vote “strongly oppose” or “strongly support”, the decision of majority can be rejected. If the votes of majority are rejected, then each of them will lose the VetoPenaltyFeeBlocks (the default is the block value of the day, except for taxes) as a penalty, and the party that veto most decisions will also lose the additional VetoPenalty token (The default is 0.1. %) as punishment.

PROPOSAL OF PARAMETER CAHNGES

Any parameters defined here can be changed after the ParameterChangeProposal is passed.

PROPOSAL OF BOUNTY

After BountyProposal has been passed, tokens can be issued additionally and pool funds can be reserved as a bounty.

TEXT PROPOSAL

All other proposals, such as those used to update the agreement, will be coordinated through the general TextProposal.

Related Work

In the past few years, there has been a lot of innovation in blockchain consensus and scalability. In this section, some important innovations will be selected for a simple analysis.

CONSENSUS SYSTEM

Classic Byzantine Fault Tolerance

In the early 1980s, the consensus mechanism of malicious participants began to be studied. At that time, Leslie Lamport coined the term "byzantine fault tolerance" to refer to the malicious behaviors of those who attempted to misbehave, which was different from the "crash failure". The latter is just a process crash. Early solutions for synchronous networks have also been explored. There is an upper limit to network information lag, but the actual use is in highly controlled environments, such as precision flight instruments and data centers that use atomic clock synchronization. Until the late 1990s, Practical Byzantine Fault Tolerance (PBFT) was gradually promoted as an effective, partially synchronized consensus algorithm. It can tolerate 1⁄3 participants with malicious behavior. PBFT became the standard algorithm and spawned various versions, including the algorithm recently proposed by IBM and used in the Hyperledger superbook.

Compared to PBFT, the main benefit of the BPOS consensus is that it has improved and simplified underlying structure, some of which has followed the results of the blockchain paradigm. In BPOS, blocks must be submitted in order, which eliminates complexity and saves on communication costs associated with state changes in PBFT. In the Diamond Network and many cryptocurrencies, if the block N itself is not committed, then the block N+i (i>=1) after it cannot be submitted. If the communication bandwidth limitation causes block N not to be submitted to Diamond Network done, then using communication bandwidth for sharing votes to block N+i is a waste. If block N is not committed due to a network partition or a node drop, then N+i cannot be committed anyway.

In addition, the state of Merkel hash to record application can be used to package the transaction into chunks, rather than the timing summaries of PBFT checking mechanism. This allows light Client to submit proof of transaction and cross-chain communication faster.

Many features other than PBFT features are also optimized in BPOS. For example, the block submitted by the verifier is divided into multiple parts, which are Merkelized and then broadcasted between the nodes. Its broadcast performance can be improved through this way. Moreover, BPOS does not make any assumptions about point-to-point connections, as long as the network between the peers is not disconnected, it will function properly.

Delegated stake of BitShare

In BitShares, the relevant party selects "Witness" to submit the transaction order and submit; the relevant party selects "Principal" to coordinate software updates and parameter changes. Although BitShare achieves high performance in an ideal environment: 100k tx/s, with lag of 1 second. Each piece has only a single signature, and the finality of the transaction is slightly longer than the block time. A standard protocol is still under development. Stakeholders can remove or replace verifiers with malicious behavior on a daily basis, but unlike margin mechanism of BPOS PoS's. BitShares does not require a verifier or agent to submit a deposit. If a double-flower attack occurs, the deposit will not be reduced.

BitShares is not the first blockchain to use proof-of-stake (PoS), but it has made a huge contribution to the research and advancement of PoS in the blockchain, especially in DPoS, that is delegated proof-of-stake. In BitShares, the relevant party selects "Witness" to submit the transaction order and submit; the relevant party selects "Principal" to coordinate software updates and parameter changes. Although BitShare achieves high performance in an ideal environment: 100k tx/s, 1 second lag. Each piece has only a single signature, and the finality of the transaction is slightly longer than the block time. A standard protocol is still under development. Stakeholders can remove or replace verifiers with malicious behavior on a daily basis, but unlike BPOS PoS's margin mechanism, BitShares does not require a verifier or agent to submit a deposit. If a double-flower attack occurs, the deposit will not be reduced.

Stellar

Based on Ripple's solution, Stellar optimizes the federal Byzantine protocol model, where participation in the consensus process does not constitute a fixed global process. Instead, each process node organizes one or more "arbitration pieces", each of which constitutes a set of trusted processes. The "quorum" in Stellar is defined as a set of arbiters containing at least one node. Thus agreement can be reached.

The security of the Stellar mechanism relies on the assumption that the intersection of any two arbitrations is non-empty. At the same time, the availability of a node requires that at least one "arbitration piece" consist entirely of honest nodes. This requires a compromise on the size of the “quorum”: it is difficult to reach consensus if people are too few; and it is hard to trust everyone if people are too many. It may be difficult to balance without making significant assumptions about trust. In addition, the node must maintain a certain number of arbitrators to obtain sufficient fault tolerance (or any "complete node", dependencies of most of the results), and provide a hierarchical configuration strategy similar to the Border Gateway Protocol (Border Gateway Protocol, BGP). BGP is used by Internet Service Providers (ISPs) to establish global routing tables and is also used by browsers to manage Transport Layer Security (TLS) certificates. They are notorious for being insecure.

The BPOS-based PoS criticism in Stellar's paper can be mitigated by the token strategy described in this article. This paper proposes a new token called “token” that represents the costs and rewards generated in the course of future transactions. The advantage of BPOS PoS is that its principle is relatively simple, while still fully guaranteeing and proving security.

BitcoinNG

BitcoinNG is an improvement to Bitcoin that allows vertical expansion, such as increasing the size of the block to avoid negative economic consequences, such as a disproportionate impact on miners. This improvement is achieved by separating the leader election from the transaction broadcast: the leader is first elected by the PoW of the "micro-block micro-blocks", and then the leader can broadcast the transaction until the next new "micro-block". This reduces the bandwidth requirements required to win a PoW tournament, making the miner more competitive, and speeding up the submission of the transaction by allowing the last miner to submit a microblock.

Casper

Casper is the PoS consensus algorithm proposed by Ethereum. Its main mode of operation is the consistency of “predictive bets”. By having the verifiers based on other bets they currently see, iteratively annotates which blocks they think will be submitted into the blockchain. Ultimateity can be achieved immediately.

Compared to BPOS, Casper's main advantage may be to provide "availability beyond consistency" - consensus is not required to exceed 50% of voting rights - perhaps at the expense of submit speed or implementation complexity.

HORIZONTAL EXPANSION

Interledger Protocol

The Interledger Protocol (ILP) is not a strict extension. It provides a specified interaction across different ledger systems through a loosely coupled bilateral relationship network. Like the Lightning Network, the purpose of ILP is to implement payment, but it pays particular attention to the type of cross-ledger payment and extends the processing mechanism of atomic transactions so that the processing of the transaction not only supports hash locks, but also includes a quorum of notaries (called the atomic transport protocol). The atomic mechanism that the latter implements among ledgers is similar to BPOS's light customer SPV mechanism, so it is necessary to compare the differences between ILP and Diamond Network / IBC, as described below.

1. ILP does not support changes to connector notaries, nor does it allow for flexible weights between notaries. In addition, IBC is specifically designed for blockchains, verifiers can have different weights, and members can change at any time as the blockchain evolves.

2. Same as the Lightning Network, the confirmation can be sent to the initiator only when the recipients in the ILP are online. In IBC token transmission, the set of verifiers in the blockchain where the recipient is located is responsible for providing confirmation, rather than receiving the user himself.

3. The biggest difference is that the ILP connector does not need to be responsible for maintaining the authority of the payment status. However, in the Diamond Network, the verifier of the hub is responsible for the IBC token transmission status and the authority of the number of tokens held in each done. Allowing a safe and asymmetric exchange of tokens from done is an essential innovation. The ILP connector in the Diamond Network can be seen as a persistent and secure blockchain ledger: the Diamond Network hub.

4. Cross-account payments within the ILP require the support of an exchange's instruction set. Because there is no a token transfer from one ledger to another, only the transfer of market equivalents can be achieved.

Side Chain

Side chain is a mechanism for extending the performance of Bitcoin networks by using a "two-way hook" to replace the blockchain with the Bitcoin blockchain. (Two-way hooks are equivalent to bridging, and are called "bridges" in the Diamond Network to distinguish them from the market). The side chain allows Bitcoin to be easily moved between the Bitcoin blockchain and the side chain and allows experimentation with new functions on the side chain. In the Diamond Network Hub, the side chain and Bitcoin are light clients to each other, using SPV proof when moving between Bitcoin blockchain and side chain. Of course, because Bitcoin uses PoW, the Bitcoin-centric side chain suffers from many problems and risks caused by PoW as a consensus mechanism. Moreover, this is a Bitcoin-maximizing solution that does not natively support a variety of token and done network topologies like the Diamond Network. However, the core mechanism of the two-way hook is in principle the same as that used by the Diamond Network.

Expanding Efforts of Ethereum

Ethereum is currently working on many different strategies to partition the state of the Ethereum blockchain to address the need for scalability. The goal of these efforts is to maintain the current Ethereum virtual machine's abstraction layer above the shared state space. At present, a number of research work is underway.

Diamond Network vs Ethereum 2.0 Mauve

Diamond Network and Ethereum 2.0 Mauve have different design concepts.

- Diamond Network is for tokens and Mauve is about expanding computing power.
- Diamond Network is not limited to EVM, so even different VM can interact.
- Diamond Network lets the creator of done determine the verifier.
- Anyone can create a new done in the Diamond Network (unless the manager decides otherwise).
- Isolation between the hub and done is invalid, so the global token invariant can be maintained.

UNIVERSAL EXPANSION

Lighting Network

The Lightning Network was designed as a token transfer network that runs on the top of the Bitcoin blockchain (and other public blockchains) , and transfer from the consensus ledger to the so-called "payment channel" by shifting most transactions. This is achieved by a chain cryptocurrency script that enables both parties to enter a stateful contract held by both parties, update the state by sharing a digital signature, and finally issue evidence on the blockchain after the contract is over. First of all, it is welcomed by the cross-chain atom exchange transaction. By opening a payment channel with multiple parties, participants of the Lightning Network can become a central point to provide routing for other people's payments, resulting in a fully connected payment channel network at the cost of funds tied to the payment channel.

While Lightning Networks can easily span multiple independent blockchains and achieve value transfer through the trading market, it does not enable asymmetric token transactions from one blockchain to another. The main advantage of the Diamond Network described here is the direct token exchange. In other words, we hope that payment channels and lightning networks will be widely adopted along with our token transfer mechanism, saving costs and protecting privacy.

Isolating Verifier

Isolation Witness is a Bitcoin Improvement Recommendation BIP, which is designed to increase the transaction throughput per block by a factor of 2 or 3 while enabling new nodes to synchronize blocks faster. The highlight of this solution is how it allows soft fork upgrades under the limitations of Bitcoin's current protocol (for example, clients with older versions of the software will continue to run after the upgrade). BPOS has no design limitations as a new protocol, so it has different extension priorities. BPOS's looping algorithm is based primarily on cryptographic signatures rather than mining. This algorithm allows horizontal expansion through multiple parallel blockchains, while more conventional, more frequent block submissions also allow for vertical expansion.

Appendix

FORKING ACCOUNTABILITY

A well-designed consensus agreement should provide some protection for the system beyond fault tolerance or consensus errors. It is especially necessary in financial systems that can achieve substantial economic returns through Byzantine behavior. Forking accountability is a very important safeguard mechanism that a process that causes a consensus error (e.g. having the protocol client begin accepting different values—that is, forking) is identified and punished according to the protocol rules, and even be transferred to the judicial system for disposal. But when the judicial system is unreliable or the litigation costs are extremely expensive, in order to let the verifiers participate in this mechanism, the system will force them to establish a security deposit, and the deposit will be fined or cut once the malicious behavior is detected.

Note that this is different from Bitcoin, which is a regular occurrence due to the probabilistic nature of network asynchronous and local hash collisions. Because in many cases, the malicious forks and forks caused by non-synchronization are indistinguishable. Bitcoin can hardly perform forked accountability accurately unless the miner pays hidden opportunity costs for the isolated blocks.

BPOS CONSENSUS

We divided the voting phase into two stages: pre-voting and pre-submission. A vote can be used both for a specific block and for Nil. We call the pre-voting sum of a single block of more than 2⁄3 in the same round and the pre-submission sum of a single block of more than 2⁄3 in the same round as Dommit. If the pre-submission of Nil in the same round exceeds 2⁄3, they will proceed to the next round.

Note that the strict certainty in the protocol raises a weak synchronization hypothesis because the initiator of the error must be detected and skipped. Verifiers will wait for a period of time before pre-voting for Nil, which is called as timeout proposal. And the wait time for this timeout proposal will also increase as each round progress. Each round is completely asynchronous, and only when the verifier listens to more than 2⁄3 of the online vote can enter the next round in the process. In fact, it requires extremely powerful obstacles to thwart this weak synchronization hypothesis (resulting in no consensus, unable to submit blocks), and the difficulty of doing so by the random value of timeout proposal by each verifier cab be increased.

Another additional constraint, or locking convention, ensures that the network eventually submits only one block at each height. Any malicious act that attempts to submit more than one block at a given height will be identified. Firstly, a pre-submission of a block must be considered justified and submitted in the form of Dolka. If the verifier is ready to pre-submit a block in the R_1 round, we say that they locked the block, and then Dolka, which is used to verify the new pre-commit action for the R_2 round, must enter the R_Dolka round, where R_1 < R_Dolka <= R_2. Secondly, verifiers must propose and/or pre-vote for the blocks they lock. These two conditions work together to ensure that the verifier cannot perform a pre-submission operation without adequate justification for its legitimacy, and that the verifier who has completed the pre-submission can no longer vote for the pre-submission contribution of other things. This not only ensures the security of the consensus algorithm, but also ensures its activity.

BPOS LIGHT CLIENT

Because the generation of a side chain (a fork) means that at least 1⁄3 of the security interest is penalized, the BPOS Proof of Entitlement (BPOS-PoS) cancels the requirement to synchronize all block headers. Of course, the penalty is also required to share the evidence of the fork, so the light client must store any block hashes it witnesses. In addition, the light client can be periodically synchronized with the verifier group's changes to avoid remote attacks (but other solutions are also possible).

Similar to Ethereum, BPOS enables applications to embed a global Merkel root hash in each block, making it easy to verify state queries, such as querying account balances, values in smart contracts, or the existence of unused transaction output (UTXO), which is determined by the nature of the application.

DEFENSE AGAINST LONG-RANGE ATTACKS

Assuming that there is a sufficiently flexible broadcast network collection and a set of static verifiers, then any blockchain forks can be detected and the margin submitted by the verifier who initiated the attack will be fined. This new method, first proposed by Vitalik Buterin in 2014, solves the problem of "no relevant interest" in other rights proof cryptocurrencies. But since the verifier group must be able to be changed, some of the original verifiers will release the bond binding over a longer period of time, which gives them the freedom to create new chains from the creation block, and because they don’t have a locked margin and they will not have to pay any fees for this behavior. This type of attack is called a long-range attack (LRA), and the latter can punish the verifier who initiated a fork in the deposit binding compared to the short-range attack (assuming a BPOS-like consensus) Fork Accountability Byzantine Fault Tolerance Algorithm). Therefore, long-range attacks are often considered as a dangerous blow to the proof of rights mechanism.

Fortunately, long-range attack (LRA) can be mitigated in the following ways. Firstly, for the unbind verifier (retrieving the mortgage deposit and no longer obtaining the fee from the participation consensus), the margin cannot be transferred within a certain period of time, and it can also be called the "unbinding period". This cycle may be several weeks or months. Secondly, for the security of the light client, regarding the first time it connects to the network, it must verify the latest block hash or several best block hashes based on the trusted source. This situation is sometimes referred to as "weak subjectivity." Finally, in order to ensure security, frequent synchronization with the latest verifier group is required, and the duration is the same as the unbinding period. This ensures that the Lightweight client knows the change of the verifier group before losing any rights due to the verifier's untied funds, otherwise the unbound verifier will start creating a new block after the height of its binding, to implement a long-range attack to trick the client (assuming it can control enough early private keys).

It is noted that, using such a way to combat remote attacks (LRA) requires a thorough inspection of the proof-of-work original security module. In Proof of Work (PoW), a light client can easily synchronize with the current height of a trusted found block at any time by running a proof of effort in each block header. However, in order to combat remote attacks (LRA), we need to track the changes of the verifier group be on a regular basis, which must verify the information collected from the network based on reliable sources when first coming online. Admittedly, the latter requirement is similar to that of Bitcoin, and its protocols and software must also be obtained from reliable sources.

The above methods for preventing remote attacks are better suited for the blockchain verifier node and the full node driven by BPOS, because these nodes are required to maintain a connection with the network. These methods are equally applicable to light client that wants to synchronize frequently with the network. However, for light clients that do not want frequent access to the Internet or blockchain networks, there is another way to solve the problem of long-range attacks. The non-verifier node can use the token as a margin during a long unbinding period (such as longer than the verifier's unbinding period), and provides the current validity of second-party proof and solutions of past block hashes for the light client. Although these tokens are not valuable for the security of the blockchain consensus, they can provide a strong guarantee for the light client. If historical block hash queries are supported in Ethereum, then anyone can bind their tokens with specific smart contracts and provide proof of payment services to effectively develop a market against LRA security issue for light clients.

OVERCOMING FORKS AND REVIEWING ATTACKS

Due to the definition of the submitted block process, any node with no less than 1⁄3 of the voting rights after the union can stop the blockchain operation by offline or not broadcasting the ballot. Such a union can also review specific transactions by rejecting blocks containing these transactions, although this would result in most block proposals being rejected, resulting in a slower block submission rate, reducing its usefulness and value. Malicious alliances may still broadcast ballots on a continuous basis, forcing them stop by blocking the blockchain, or using a combination of any of these attacks. Eventually, it will cause the blockchain to fork by double signature or violation of the locking rules.

If a globally active perpetrator is involved, the network will be segmented by a method that may result in a slower verification of the subset of people. This is not just a limitation of BPOS, but rather a limitation of all consensus agreements that are controlled by active hostiles.

For these types of attacks, the verifier's subset should be coordinated externally to sign a reorganization proposal that selects a fork (and all evidence associated with it) with the initial subset of the signed verifier. The verifier who signed such a reorganization proposal will waive the deposit that belongs to him on all other forks. The client shall verify the signature and any relevant evidence in the reorganization proposal and make a judgment or prompt the end user to make a decision. For example, a mobile wallet app should give users a security warning when they are likely to accept any reorganization proposal signed by more than half of the initial verification people.

When more than 1⁄3 of the voting rights are dishonest, a non-synchronized Byzantine fault-tolerant algorithm can't reach a consensus. However, forks assume that no less than 1⁄3 of voting rights have become dishonest due to improper double signatures or lock changes. Therefore, signing a reorganization proposal is a coordination issue, and no asynchronous protocol can solve this problem (that is, it is automatic and does not consider the reliability of the underlying network). At present, we have left the coordination of the reorganization proposal to the users through the social consensus of the Internet media. The verifier must ensure that there are no network segmentation issues before signing the reorganization proposal to avoid signing two conflicting reorganization proposals.

Assuming that the external coordination medium and protocol are reliable, there will be fewer concerns about forks than for review attacks.

In addition to the bifurcation and censorship system that requires more than 1⁄3 of Byzantine voting rights to be initiated, more than 2⁄3 of the joint voting rights may be submitted in an arbitrary, invalid state. This is a problem specific to any consensus system of Byzantine fault-tolerant algorithms. Unlike double signatures that using a simple verifiable proof to create a fork, detecting an invalid state submit requires a non-authentication node to verify the entire block, which means that the non-authenticated node will keep a copy of the local state and execute each transaction. And then calculate the root cause of the state for themselves. Once detected, the only way to deal with such failures is social consensus. For example, in the case of Bitcoin problems, whether due to software breaches (as in March 2013), or due to the ineffective status of miners byzantine behavior (as July 2015), by merchants, development The social consensus established by the close-knit community of miners, miners and other organizations will allow them to participate in the work of repairing the network in accordance with the division of labor. In addition, since the verifier identity of the BPOS blockchain is identifiable, the submission of invalid status can actually be punished by law or other external legal systems, if necessary.

ABCI DESCRIPTIONS

ABCI consists of three main types of information that are passed from the consensus engine to the application, and then the application responds with the appropriate response message.

AppendTx information is the primary delivery medium for applications. Every transaction in the blockchain is passed through by this information. The application requires to validate each transaction, which is done by receiving AppendTx information for the current state, application protocol, and transaction password credentials. Verified transactions will require to update the application state by adding values to the key-value store or updating the UTXO database.

CheckTx information is similar to AppendTx information, but it is only for transaction verification. BPOS's memory pool will first validate the transaction with CheckTx and will only pass valid transactions to other nodes. The application checks the transaction serial number and returns an error based on CheckTx if the serial number expires.

The Dommit information is an encrypted submission used to calculate the current application state that will be stored in the next block header. This has convenient features. The inconsistency of the state will be like causing a program error, causing the blockchain to fork. This also simplifies the development of secure light clients, because the Merkel hash proof can be verified by checking the block hash, which is signed by the specified number of verified people (by voting rights).

In addition, the ABCI information allows the application to keep track of changes to the verifier group and have the application receive block information such as height and submission of votes.

The ABCI request/response is a simple Protobuf message.

AppendTx

- Command line parameters:

- Data ([ ]byte): Transaction request information

- Returns:

- Code (uint32): Reply code

- Data ([ ]byte): The result byte, if any

- Log (string): Error message

- Use: Submit and execute a transaction. If transaction is valid, then return to CodeType.OK

CheckTx

- Parameter of command line
- Data ([ ]byte): Transaction request information
- Returns:
- Code (uint32): Reply code
- Data ([ ]byte): Result byte, if any
- Log (string): Error information

Use: Verify a transaction. This information should not change the application state. The transaction is first run through CheckTx before being broadcasted to other nodes. You can initiate a semi-stated CheckTx and clear the state on Dommit or BeginBlock to allow the sequence to execute related transactions in the same block.

Dommit

- Return value:
- Data ([ ]byte): Merkel root value
- Log (string): Debug or error message
- Use: Returns the current application status

Query

- Command line parameters:
- Data ([]byte): Request data
- Return value:
- Code (uint32): Reply code
- Data ([]byte): Query reply byte
- Log (string): Debug or error message

Flush

- Use: Refresh the reply queue. Application that applies types.Application doesn't need to implement this information -- this is handled by the project.

Info

- Return value:
- Data ([]byte): Information byte string
- Use: Returns information about the state of the application. Application specification.

SetOption

- Parameters:
- Key (string): set parameters
- Value (string): parameter value
- Return value:
- Log (string): Debug or error message

Use: For example, the connection to the memory pool can be set to "mode" (mode) and the value is "mempool" (memory pool). Or for a consensus connection, set the key to "mode" and the value to "consensus". Other options are tailored to suit your specific application.

InitChain

- **Parameters:**
- Validators ([]Validator): Initialize the Creator Authenticator
- Use: Called when the creation block is created

BeginBlock

- Parameters:
- Height (uint64): The height of the block at the beginning

Use: Provides a signal for the beginning of a new block. Called before the additional transaction (AppendTxs).

EndBlock

- Parameters:
- Height (uint64): Height of block at the end

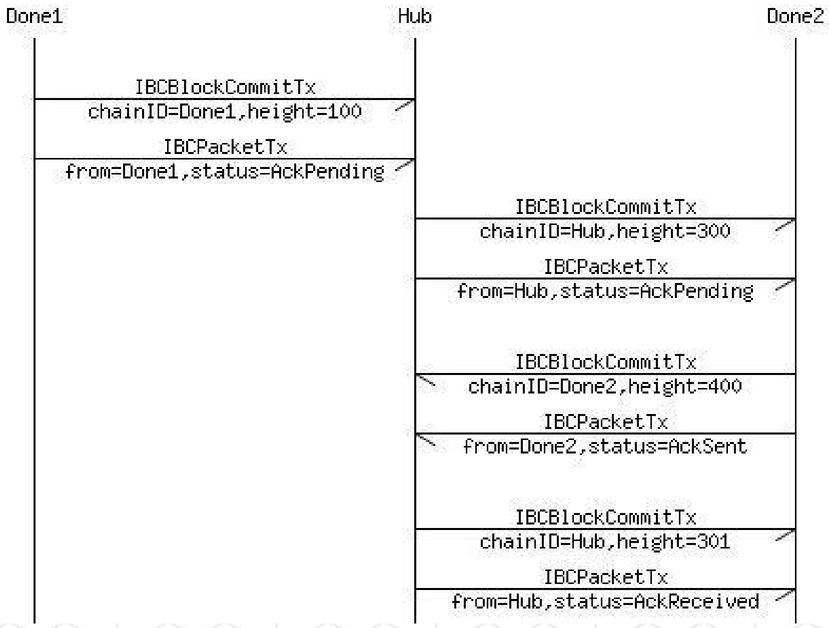
Return value:

- Validators ([]Validator): The verifier with the change of the new vote (return to zero)
- Use: Provides a signal for the end of the block. Called after all transactions before each commit.

IBC PACKET DELIVERY CONFIRMATION

The sender has many reasons why requiring to provide a packet delivery confirmation for receiving chain. For example, if the destination chain is expected to go wrong, the sender may not be able to understand the state of the destination chain. Or, when the destination chain may encounter the attack of a denial of service due to a violent increase in receiving packets, the sender will want to set the packet timeout period (with MaxHeight, the maximum packet domain).

In these cases, the sender can request delivery confirmation by setting the initial packet status on AckPending. The delivery is then confirmed by the receiving chain through a Merkel hash containing a simplified IBCPacket application.



Firstly, an IBCBlockDommit and IBCPacketTx are uploaded to the "hub" to prove the existence of the IBCPacket on "Partition 1". Suppose that the value of IBCPacketTx is as follows:

- **FromChainID**: “done1”
- **FromBlockHeight**: 100 (supposed)
- **Packet**: an **IBCPacket**:
    - **Header**: an **IBCPacketHeader**:
        - **SrcChainID**: “done1”
        - **DstChainID**: “done2”
        - **Number**: 200 (say)
        - **Status**: **AckPending**
        - **Type**: “coin”
        - **MaxHeight**: 350 (“hub” is supposed as current height of 300)
    - **Payload**: <a payload byte of "token">
- **FromChainID**: “done1”
- **FromBlockHeight**: 100 (suposed)
- **Packet**: an **IBCPacket**:
    - **Header**: an **IBCPacketHeader**:
        - **SrcChainID**: “done1”
        - **DstChainID**: “done2”
        - **Number**: 200 (suposed)
        - **Status**: **AckPending**
        - **Type**: “coin”
        - **MaxHeight**: 350 (current height of "hub" is supposed as 300)
    - **Payload**: < a payload byte of "token">

Secondly, an IBCBlockDommit and IBCPacketTx are transmitted on both "Partition 2" to prove the existence of the IBCPacket on the "hub". Supposed that the value of IBCPacketTx is as follows:

- **FromChainID**: “Hub”
- **FromBlockHeight**: 300
- **Packet**: an **IBCPacket**:
    - **Header**: an **IBCPacketHeader**:
        - **SrcChainID**: “done1”
        - **DstChainID**: “done2”
        - **Number**: 200
        - **Status**: **AckPending**
        - **Type**: “coin”
        - **MaxHeight**: 350
    - **Payload**: <a payload byte identical to "token">
- **FromChainID**: “Hub”
- **FromBlockHeight**: 300
- **Packet**: an **IBCPacket**:
    - **Header**: an **IBCPacketHeader**:
        - **SrcChainID**: “done1”
        - **DstChainID**: “done2”
        - **Number**: 200

    - **Status**: **AckPending**
    - **Type**: “coin”
    - **MaxHeight**: 350
  - **Payload**: < a payload byte identical to "token" >

Next, "done2" must add the abbreviated version of AckSent's latest status package to the application state hash. IBCBlockDommitand and IBCPacketTx are transferred to the "hub" to prove that the simplified IBCPacket exists on "Partition 2". Suppose the values of IBCPacketTx are as follows:

- **FromChainID**: “done2”
- **FromBlockHeight**: 400 (supposed)
- **Packet**: an **IBCPacket**:
  - **Header**: an **IBCPacketHeader**:
    - **SrcChainID**: “done1”
    - **DstChainID**: “done2”
    - **Number**: 200
    - **Status**: **AckSent**
    - **Type**: “coin”
    - **MaxHeight**: 350
  - **PayloadHash**: < A hash value of the payload byte identical to the “token”>
- **FromChainID**: “done2”
- **FromBlockHeight**: 400 (supposed)
- **Packet**: an **IBCPacket**:
  - **Header**: an **IBCPacketHeader**:
    - **SrcChainID**: “done1”
    - **DstChainID**: “done2”
    - **Number**: 200
    - **Status**: **AckSent**
    - **Type**: “coin”
    - **MaxHeight**: 350
  - **PayloadHash**: < A hash value of the payload byte identical to the “token”>

Finally, the "hub" must update the packet status from AckPending to AckReceived. The proof of this new completion status should be returned to "Partition 2". Supposed that the value of IBCPacketTx is as follows:

- **FromChainID**: “Hub”
- **FromBlockHeight**: 301
- **Packet**: an **IBCPacket**:
  - **Header**: an **IBCPacketHeader**:
    - **SrcChainID**: “done1”
    - **DstChainID**: “done2”
    - **Number**: 200
    - **Status**: **AckReceived**
    - **Type**: “coin”
    - **MaxHeight**: 350

- PayloadHash: <The hash bytes of the same “coin” payload>
- **FromChainID**: “Hub”
- **FromBlockHeight**: 301
- **Packet**: an **IBCPacket**:
    - **Header**: an **IBCPacketHeader**:
        - **SrcChainID**: “done1”
        - **DstChainID**: “done2”
        - **Number**: 200
        - **Status**: **AckReceived**
        - **Type**: “coin”
        - **MaxHeight**: 350
    - **PayloadHash**: < The hash bytes of the same “token” payload >

At the same time, "Partition 1" assumes that the delivery of the "Token" package has been submitted successfully, unless there is evidence on the "hub" can give the opposite proof. In the above example, if the "hub" does not receive the AckSent state of the 350th block from "Partition 2", it will automatically set it to Timeout. Evidence of this timeout can be posted back to "done1" and all tokens will be returned.

DESCRIPTION OF MERKEL TREE AND MERKEL PROOF

Two Merkel trees supported by the BPOS/Diamond Network ecosystem: simple trees and IAVL+ trees.

IBCBlockDommitTx

The IBCBlockDommitTx transaction consists mainly of:

- **ChainID (string)**: blockchain ID
- **BlockHash ([]byte)**: block hash byte, which includes the application hash Heckergen
- **BlockPartsHeader (PartSetHeader)**: The header byte set in the block section, used only to verify the voting signature
- **BlockHeight (int)**: Submit the height
- **BlockRound (int)**: Submit the round
- **Dommit ([]Vote)**: BPOS pre-submit vote submitted by more than 2⁄3 of the included block
- **ValidatorsHash ([]byte)**: Merkel root hash of the new validation group
- **ValidatorsHashProof (SimpleProof)**: A simple tree Merkel proof that proves the verifier hash in the block hash
- **AppHash ([]byte):** IAVL tree, the Merkel root hash of the application state
- **AppHashProof (SimpleProof):** A simple version of the Merkel tree that validates the application hash in the block hash **AppHash** against the **BlockHash**

IBCPacketTx

IBCPacket consists of the following items:

- **Header (IBCPacketHeader):** Packet header
- **Payload ([ ]byte):** Packet payload byte. Optional.
- **PayloadHash ([ ]byte):** Packet byte hash. Optional.

One of the Payload or PayloadHash must exists. The hash of IBCPacket is a simple version of Merkel with two items, which are the head and payload. IBCPackets without a full payload are called abbreviated packages.

IBCPacketHeader consists of the following items:

SrcChainID (string): Source blockchain ID DstChainID (string): **Target blockchain ID Number (int):** Unique number of all packets Status (enum): Can be any of **AckPending, AckSent, AckReceived, NoAck,** or **any Type (string) of Timeout**: The type is determined by the application. Diamond Network retains the "coin" package type. **MaxHeight (int)**: If the state is not **NoAckWante**d or **AckReceived** given by this height, the state is time out. Optional.

An **IBCPacketTx** transaction is composed of:

- **FromChainID (string)**: The ID of the blockchain provided to this packet is not necessary for the source
- **FromBlockHeight (int)**: The height of blockchain in which the following packet will be included (Merkle-ized) in the block-hash of the source chain
- **Packet (IBCPacket)**: A packet of data, whose status may be one of **AckPending**, **AckSent**, **AckReceived**, **NoAck**, or **Timeout**
- **PacketProof (IAVLProof)**: A proof of IAVLTree Merkle is for verifying the packet's hash against the **AppHash** of the source chain at given height

The sequence of sending packets from "done1" to "done2" via "Hub" is described in the {Figure X} function. Firstly, an IBCPacketTx will prove to "Hub" that the packet is included in the application state of "done1". Then, another IBCPacketTx will prove to the "done2" that the packet is included in the "Hub" application state. In this process, the fields of IBCPacketTx are the same: SrcChainID is always "done1", and DstChainID is always "done2".

The **PacketProof** must have the correct Merkle-proof path, as follows:

```
IBC/<SrcChainID>/<DstChainID>/<Number>
```

When "done1" wants to transfer the packet to "done2" through "Hub", the IBCPacket data is the same regardless of whether the packet is Merkelized in "done1", "Hub" or "done2". The only variable field is the Status for tracking delivery.

Conclusion

The Diamond Network has emerged as an interactive hub for cross-chain interconnection, from which the value islands of the blockchain can communicate with one another, ultimately realizing the vision of decentralized cross-chain Internet.

Simple version of Merkel tree

The simple version of the Merkel tree is based on a basic static list. If the number of items is not the power of 2, then some leaves will be on different layers. The simple tree tries to make the sides of the tree at the same height, but the left side may be slightly larger. This Merkel tree is used for Merkelization of a block transaction, and the top element is the root of the application state.

A SimpleTree with 7 elements

IAVL+Tree

The purpose of the IAVL+ data structure is to permanently store the key pair in the application state so that the determined Merkel root hash can be efficiently computed. The balance of this tree is achieved by a variant of the AVL algorithm, all running O (log(n)).

In the AVL tree, the heights of the two subtrees of any node at most have one difference. Whenever this happens, it is contrary to the update. The tree will again reach equilibrium by creating a new O(log(n)) node (pointing to the unmodified node on the old tree). In the initial AVL algorithm, internal nodes can also retain key value pairs. The AVL+ algorithm (note that there is a "+" sign) modifies the AVL algorithm to keep all values on the leaf nodes, and only uses branch-nodes to store the keys. This simplifies the algorithm while maintaining a shorter Merkel hash trajectory pair.

The AVL+ tree is similar to the Pear tree of Ethereum. There is also a certain compromise. The key does not need to generate a hash before it is embedded in the IAVL+ tree, so this provides a faster command iteration for the key space, which can benefit many applications. The logic implementation is simpler, requiring only two types of nodes, internal nodes and leaf nodes. As a balanced binary tree, Merkel proved to be shorter on average. On the other hand, Merkel root of the IAVL+ tree has updates depending on the command. We will support additional valid Merkel trees, such as the Ethereum Pap tree when binary variables are available.

At the same time, only branch-nodes are needed to store the keys. This simplifies the algorithm while maintaining a shorter Merkel hash trajectory pair.

The AVL+ tree is similar to the Pear tree of Ethereum. There is also a certain compromise. The key does not need to generate a hash before it is embedded in the IAVL+ tree, so this provides a faster command iteration for the key space, which can benefit many applications. The logic implementation is simpler, requiring only two types of nodes: internal nodes and leaf nodes. As a balanced binary tree, Merkel is proved to be shorter on average. On the other hand, Merkel root of the IAVL+ tree depends on the updates of command. We will support additional valid Merkel trees, such as the Ethereum Pap tree when binary variables are available.

TRANSACTION TYPE

In the standard implementation, transactions are flooded into the Diamond Network Hub application through the ABCI interface.

Diamond Network Hub will receive several major transaction types including SendTx, BondTx, UnbondTx, ReportHackTx, SlashTx, Burn Token Tx, ProposalCreateTx, and ProposalVoteTx (send transactions, bind transactions, unbind transactions, attack report transactions, cut transactions) , token burning transactions, creation of proposal transactions, and proposal voting transactions), which does not require to be explained and will be filed in future versions of the document. Here we mainly list two main IBC transaction types: IBCBlockDommitTx and IBCPacketTx (ie IBC block submission transaction and IBC packet transaction)