TOP Network Technical Whitepaper

1 Introduction

Since emerging just over a decade ago, the blockchain space has experienced tremendous growth. With the promises of decentralization, transparency, P2P commerce, digital trust, and brand-new business models, blockchain has inspired an entire movement. However, while blockchain technology undoubtedly holds great promise for the future, many have been left wondering why it is yet to produce any world changing applications. Unfortunately, current blockchain technologies have still not matured to a point where commercial applications are viable, except for a few select uses cases.

As the pioneer, Satoshi Nakamoto introduced the world to the basic concepts of decentralization and blockchain through Bitcoin. Years after, Ethereum was released to expand upon Bitcoin's limited uses cases with the revolutionary concept of smart contracts. This brought to the forefront the concept of Decentralized Applications, or DApps. However, Ethereum was later found to be too slow for most applications, which lead to flurry of new projects attempting to solve its issues, including projects like EOS and Tron. While improving upon their predecessor in terms of scalability, many of these next gen public chains were rushed, and sacrificed decentralization for scalability.

Even with the modest scalability improvements in recent years, blockchain platforms have yet to pro-duce any large-scale massively adopted DApps. Broadly speaking, this is because current blockchain technologies still suffer from several flaws, including: lack of true scalability, lack of support for real-time applications, lack of developer tooling, lack of support for complex business workflows, and a lack of initial users.

This paper introduces TOP Network, a full-stack blockchain platform which addresses these inadequacies and builds upon the trials and errors of previous blockchain iterations. The platform consists of three main layers: the infrastructure layer, service layer, and application layer. Together, these layers can support the development and operations of DApps of any scale.

2 The TOP Network Blockchain Platform

TOP network is a full-stack platform for DApp development and deployment. The aim is to build an ecosystem providing everything a developer would need to easily build scalable decentralized applications. This is accomplished through a comprehensive stack consisting of three layers.

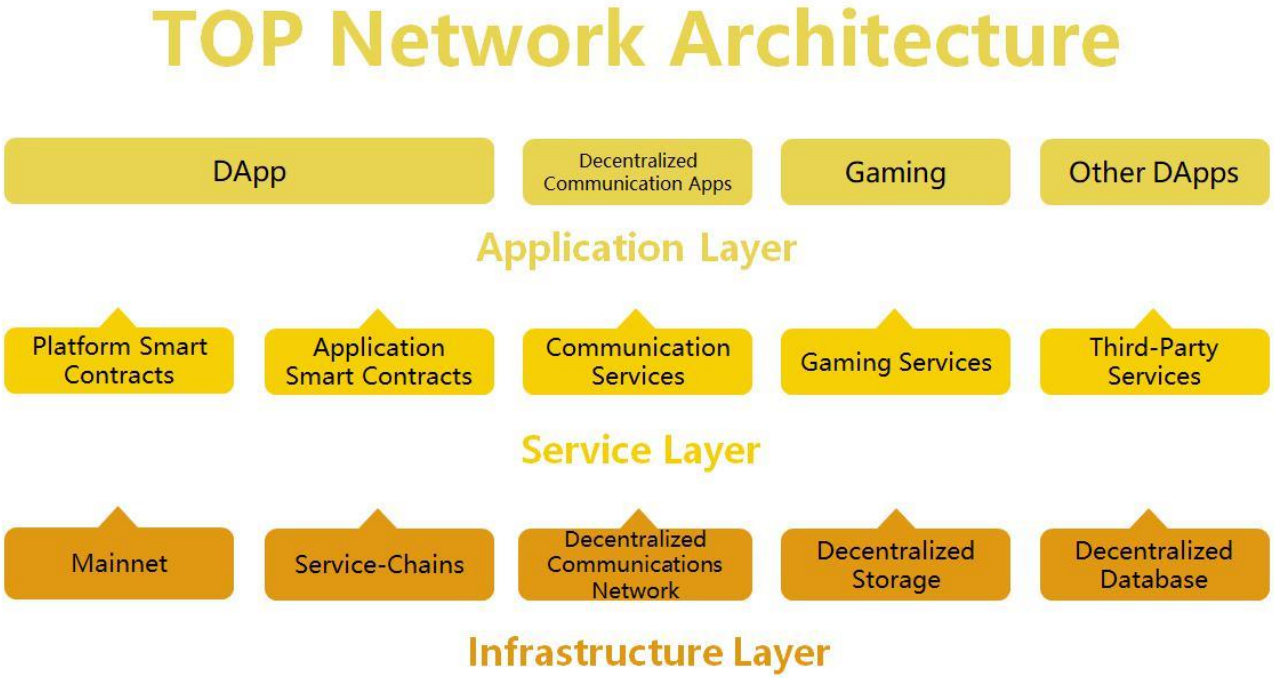


Fig. 1. TOP Architecture

2.1 Infrastructure Layer

From observing the past attempts of other projects, it has become clear that the only way to scale blockchains while maintaining all of decentralization, security, and scalability is horizontal scale-out techniques like sharding. Projects such as Quarkchain and Zilliqa were the first to attempt blockchain sharding. While providing transactional sharding, these and other similar projects have yet to develop a comprehensive sharding design which shards in terms of not only transactions, but also state and net-work capacity. As a result, bottlenecks still exist which inhibit scalability

Through extensive R&D, TOP Network has developed the first comprehensive multi-layer sharding de-sign, including the more difficult full-state sharding. However, we realized from the beginning that on-chain scaling is not sufficient in isolation. As a result, TOP has pioneered a multi-layer scaling paradigm, which integrates scaling techniques in layer-0, layer-1, and layer-2. With breakthroughs in the areas of P2P networking, blockchain sharding, multi-chain architectures, and more, TOP Network supports true horizontal scale-out. In all, the Infrastructure Layer includes a fully sharded high-performance PoS-based public chain with a novel parallel pBFT consensus mechanism, pluggable service-chains, distributed file storage and distributed database

technology, and built in layer-2 scaling. This is all sup-ported by an underlying custom built and highly optimized “P2P Internet” networking architecture in “layer-0”.

2.2 Service Layer

Currently, building blockchain applications is very difficult. Only a select few developers have the abilities required to develop well-made DApps, which has likely stunted the growth of the DApp space. If DApps are to become as widespread as traditional apps, comprehensive sets of developer tools are required. TOP Network provides service layer frameworks and tooling which aid in the development process by abstracting the underlying blockchain aspects. Furthermore, TOP provides multiple types of smart contracts, including Platform Contracts for secure asset transfer, and flexible Application Level Contracts for complex business logic. Finally, the Service Layer includes industry specific development frameworks similar to what is found in the traditional application development scene.

2.3 Application Layer

Often the most difficult part of building an App is garnering enough users to reach a critical mass where the business model begins to work. Unfortunately, this is even greater a challenge in the blockchain world, as there are virtually no users to draw from. TOP Network is uniquely positioned in the space thanks to a large pre-existing userbase. The application layer of TOP Network consists of an application ecosystem drawing from a large existing userbase which can be leveraged by third-party developers.

These layers offer the three most vital components of an application platform: a scalable infrastructure, comprehensive developer tooling, and an application ecosystem with an abundance of users. This is how TOP Network is able to meet the needs of real-world businesses and large-scale applications.

3 TOP Chain Infrastructure

TOP Chain is the first fully sharded, entirely permissionless public chain providing high TPS, near instant confirmation times, and low to zero gas fees. Building a chain with these features is very difficult and requires a multitude of techniques. As a result, we have developed a multi-layer scaling stratagem to accomplish these specifications.

3.1 P2P Network Architecture: Layer-0

At the base layer of every decentralized blockchain system lies a means of communication via a P2P network. This base layer could be called layer-0 of the blockchain stack. Most blockchain projects do not innovate in layer-0, and instead simply reuse popular implementations of P2P networks.

Unfortunately, P2P networks have a big downside; they are not as fast or as efficient when compared to centralized networks. To overcome this, we designed our own customized P2P network architecture. The guiding principle in our design process was to build a P2P network architecture that over time could become nearly as fast as a centralized network. This required a number of new innovations along with some tweaks to existing protocols.

3.1.1 P2P Overlay Internet

In TOP's ecosystem, there are numerous components and sub-networks which all must communicate and transfer data between each other. If the P2P network through which these various parts interact is weak, the entire platform is affected. Popular P2P network implementations have been optimized for certain use cases over the years but are not quite adequate for such an ecosystem of many distinct yet connected networks requiring near real-time latency. This is particularly true for a sharded system where there is constant interaction between shards. In fact, the bandwidth used for cross-shard communication can quickly become a bottleneck.

We have accounted for this by building a "P2P Internet", wherein instead of one monolithic network, there are many P2P networks organized in a hierarchical fashion, similar to the design of the Internet. Furthermore, we have developed optimized data transfer and gossip protocols to minimize bandwidth consumption and increase the efficiency of node discovery. Finally, we embed geographical information into the P2P network structure and implement smart-routing to reduce latency.

3.1.2 Kademlia DHT

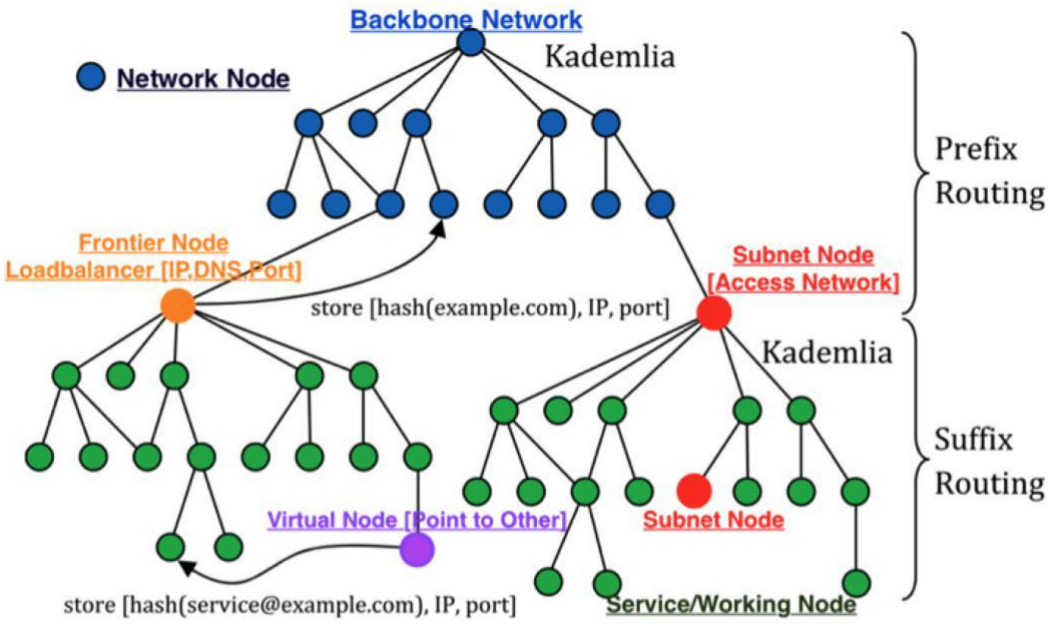


Fig. 2. Hierarchical Network

The basis for each P2P network on TOP Network is a modified Kademlia DHT implementation. How-ever, instead of using a single Kad network for communication between all nodes and services, we use many layered Kad networks.

The relationship between sub P2P networks is hierarchical. Each sub-network has information about the layer directly below it. The network hierarchy is organized into Zones, Clusters, and then Shards. Each Zone is a Kad network with Clusters and Shards as “nodes.” Clusters and Shards are sub Kad networks made up of physical nodes. Zones are logical divisions, while each Cluster and Shard consists of a physical P2P Kad network of nodes. Essentially, each sub-network is arranged as if it were a node in a Kad network. We say that each P2P network within TOP is virtualized as a single node within a larger overarching Kad network.

3.1.3 XIP address

As mentioned, the network architecture of TOP is very similar to that of the Internet. The Internet is one large network of networks, while TOP is one big network of P2P networks. On the Internet, every device is assigned a routable IP address. We use a similar concept in our P2P Internet through a 128-bit construct we call an XIP address. Just as an IP address allows for the routing of a packet to any device in a network within the Internet, an XIP address allows any TOP node to route a packet to any node within a P2P network residing in the P2P Internet.

When a node is first elected and joins the network, it is assigned a unique XIP address which maps to a Kad NodeID. This mapping is stored in an election block on-chain. From a high-level, to search or connect with any peer in the overall network, a node simply uses the XIP address. Under the hood, the XIP address is translated using XRouting-Table—a local table stored by each node—which is used to find the correct sub-network based on the XIP. After this translation, the request is sent to the closest NodeIDs to the target NodeID in the destination sub-network. At this point, a regular Kad lookup commences, which returns the IP/Port of the target node.

3.1.4 Zones and Geographical Clustering

In traditional Kad networks, each node is given a random unique NodeID, and a notion of distance between two NodeIDs is introduced through the XOR metric. Nodes cluster around peers with simi-lar NodeIDs (small XOR distance). This “distance” is not based on geographical location, and so two “close” NodeIDs could actually be on other sides of the world.

This poses a problem. Nodes which interact often are not grouped as such, and so latency can be high. This is particularly problematic in terms of consensus. If a single Kad network is used as the basis for a sharded system, nodes within a particular consensus committee are randomly positioned around the world. This has negative consequences for confirmation times.

To remedy this, we make a change to the NodeID. When a node first joins the network, general geo-graphical information such as its country code and network location are determined. This information is then encoded and built into the NodeID. Therefore, NodeIDs with the same country codes will generally be placed close to one another, now both in terms of XOR distance and geographical distance.

Zones represent the top level of this geographical partitioning. Each Zone is a Kad network which spans a certain geographical region. For instance, one Zone may cover North America, while another may span Europe. Nodes within Clusters and Shards in a particular Zone are thus located in the same general region, which vastly improves latency.

3.1.5 Optimized Gossip and Data Transfer Protocol

In blockchain networks such as the underlying Bitcoin P2P network, block and transaction propagation occurs through a gossip protocol. Basically, nodes store the information of a large number of nodes, and send out blocks or transactions to a configurable number of peers. The receiving peers then relay the transaction or block through the same process. This continues until every node has received the information. This simple form of gossip is robust, as the broadcasting has high redundancy. However, it is heavy on network resources for the same reason.

TOP also uses a gossip protocol to disseminate information, but with some optimizations. Packets sent within TOP Network are broken into two parts: the header and the body. The header is just a few bytes in size, while the body is usually much larger. Instead of disseminating the entire packet to a large number of nodes in a particular sub-network, the body is sent using the most efficient method of Hierarchical-Group Multicasting, while the header is sent using a Reliable Gossip protocol. This greatly reduces the number of times that the body must be transmitted during the gossip process, saving network bandwidth consumption.

3.1.6 Reliable UDP

Most P2P networks, including Kademlia, use UDP as the underlying transport protocol. UDP is fast and has low-overhead, which is ideal for the quick iterative lookup routing scheme used in Kad networks. However, UDP is connectionless and unreliable. For most applications, UDP on its own cannot guaran-tee quality of service.

There are some UDP based protocols such as Google's QUIC which are reliable. However, QUIC is overly robust for our needs, and so introduces unnecessary overhead. Instead of using a pre-existing reliable UDP protocol, we developed our own which we call X-UDP. This protocol is customized to only include the necessities, and so maintains the low-overhead and speed of UDP.

3.1.7 Smart Routing

Connections made through Kademlia are end-to-end, and not necessarily optimal in terms of latency. To improve QoS, we introduced a smart-routing scheme which has some similarities to the routing mechanisms in the Chord DHT. When connecting to another node, the lookup process will eventually provide the details required to make a direct connection. In addition, several paths will be tested through a forwarding scheme similar to Chord. The latency between each path is compared, and the route with the lowest latency is used.

3.2 TOP Chain Data Structure

TOP Chain is made up of unique data structures specifically designed for sharding and efficient data access and storage. The three main data structures used are Core Objects, the Unit Lattice, and the Block Lattice.

3.2.1 Core Objects

TOP follows an object-oriented programming paradigm. Each user account and each smart contract are represented by what we call "Core Objects".

3.2.1.1 Accounts

A user account is specified by an address on TOP Chain. An account is an object containing state information and logic, including things like the Balance, Properties which store data, and Actions for each property. The account balance can also be treated as a special type of property of the account object. An action could be a system-level function such as a transfer, or a customized action governed by a smart contract deployed by the account owner. The account object and its methods can be easily and flexibly extended with new properties and customized actions.

3.2.1.2 Property

A property of an account is a user-defined data object, which is added as a key-value pair. The key is an arbitrary String whose value can be any data type, such as an Integer, String, List, HashMap and so on. The account balance, a special property, exists by default when the account is generated.

3.2.1.3 Action

An action provides flexible processing capabilities for a property. An action can be a built-in system function such as Hash and Vote, or a user-defined smart contract. Users can trigger an action by sending a transaction or a message.

3.2.1.4 Message

A message is an instruction that initiates an action on a property of an account. Data attached to a message contains the property, action, input parameters and output parameters. For security purposes, a message cannot alter the account balance.

A message is a special kind of transaction. An account sending high-frequency messages is subject to flow control and has to pay gas fees, or the account owner may be forced to perform proof of work (POW).

3.2.1.5 Transaction

A transaction refers to an instruction that initiates an asset transfer from one account to another. A transaction can also contain an action that will be triggered when the transaction is being processed.

3.2.2 Two-Layer Lattice

The blockchain state, which is made up of all user account objects and smart contract account objects, is stored across Unit-Lattice and Block-Lattice data structures. Technically, these lattices are a form of Directed Acyclic Graph (DAG). However, unlike other DAG based projects such as IOTA, the lattice data structure is much more organized and well suited for a sharded architecture.

3.2.2.1 Unit Lattice

A transaction or a message sent to or from an account may change the value of a certain property and consequently the state of the account. Each transaction or message applied to an account is called a transaction record, or a Unit. When TOP Chain stores the Units of an account, each Unit is linked to the previous Unit. All of the past events on the account as well as the latest status of all its properties can be retrieved through the process of navigating through all the Units of an account from the genesis Unit (that is, the initial state of all properties when the account is created), to the lastest Unit. Essentially, all Units of an account are linked together to form a tiny chain, which is named a Unit Chain. The

collection of Unit Chains of all accounts within a Shard is called a Unit Lattice.

One major benefit of this design is that consensus can be done on the account level simultaneously, which adds another opportunity for parallelization, and therefore faster throughput. This is in contrast to using a single chain per shard which records transactions from many accounts sequentially.

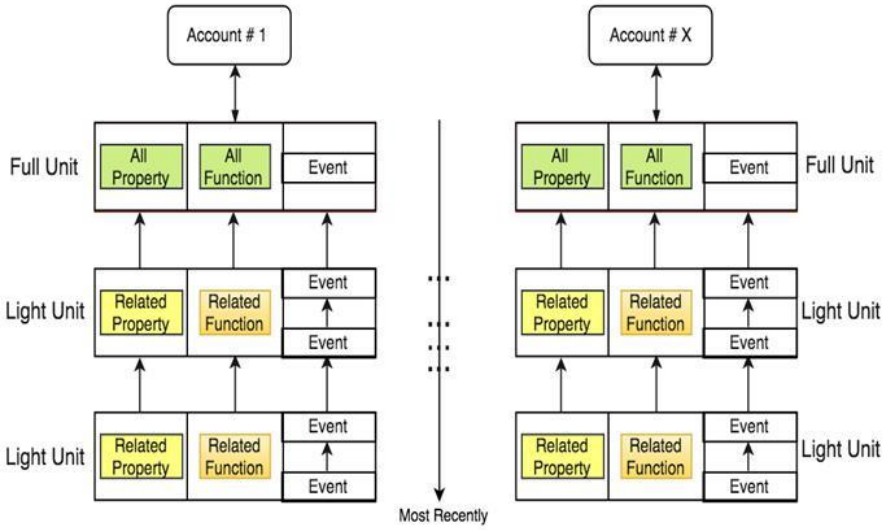


Fig. 3. Unit Lattice

As show in figure 3, the last Unit contains all of the information that proves the validity of the final state values of all properties in an account. As a result, the Unit Chain can be safely pruned to keep only the last Unit. TOP Chain can prune all Unit Chains within a Shard in real-time without affecting the whole system, significantly reducing the footprint of the entire blockchain.

3.2.2.2 Block-Lattice

Block-Lattice acts as a batching mechanism for consensus and cross-zone synchronization. The most recently updated Units are packaged into Table Blocks, and then linked into chains for each account sub-space. The set of these chains within a Zone forms the Block-Lattice data structure. Table Blocks are used for cross-shard synchronization, while the Block-Lattice is used for cross-zone synchronization.

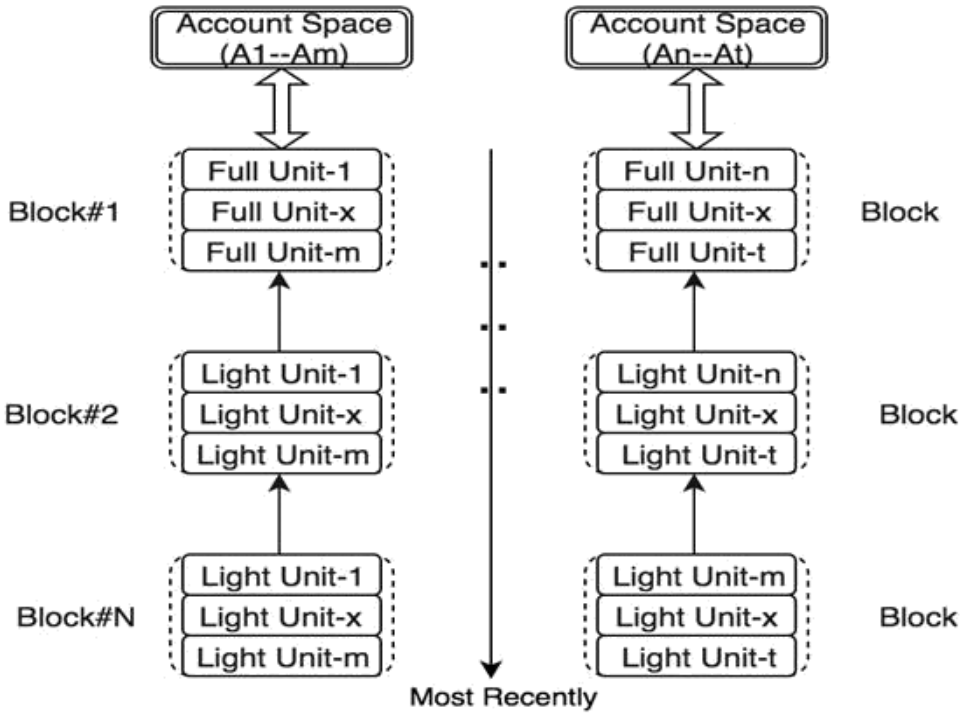


Fig. 4. Block Lattice

3.3 Sharding: Layer-1

Layer-1 scaling refers to direct performance improvements in the blockchain itself. Of all the on-chain scalability solutions, sharding has gained the most attention, and is generally considered the best route for a permissionless public blockchain. However, most sharding attempts thus far have only sharded in terms of computation, and not state and network capacity.

TOP has developed a comprehensive multi-level dynamic sharding paradigm which shards in all three aspects. This is facilitated by the underlying network infrastructure in layer-0, along with a layered con-sensus network, VRF-based sortition algorithms, a two-layer Lattice-DAG data structure, and a parallel pBFT-PoS* consensus mechanism. These innovations allow TOP Chain to achieve linear scalability while remaining secure and totally permissionless.

3.3.1 Overall Design

TOP adopts a multi-chain architecture. The entire structure of TOP Chain consists of up to 255 level-1 chains, which include the main chain and service chains. Each level-1 chain can itself support up to 255 level-2 sub-chains, and so on. For instance, a developer could branch their own level-2 chain off of the main VPN service chain specifically to support their business.

A chain at any Level can have a maximum of 255 Clusters, and each Cluster can support up to 127 Shards. Currently, the number of Shards per Cluster on the main chain is limited to 4, which means there can be a total of 1024 shards on the main chain.

Level-1 chains are distinct loosely coupled chains. Transactions are routed to the appropriate level-1

chain based on the purpose of the transaction. If it's for asset transfer, the transaction is routed to the main chain, while if it's for a type of service transaction, it is routed to the relevant service chain.

Each level-1 Beacon Chain is populated from a common set of nodes. Communication between each level-1 chain network is supported by TOP's P2P Internet architecture, allowing for efficient communi-cation between nodes across chain networks.

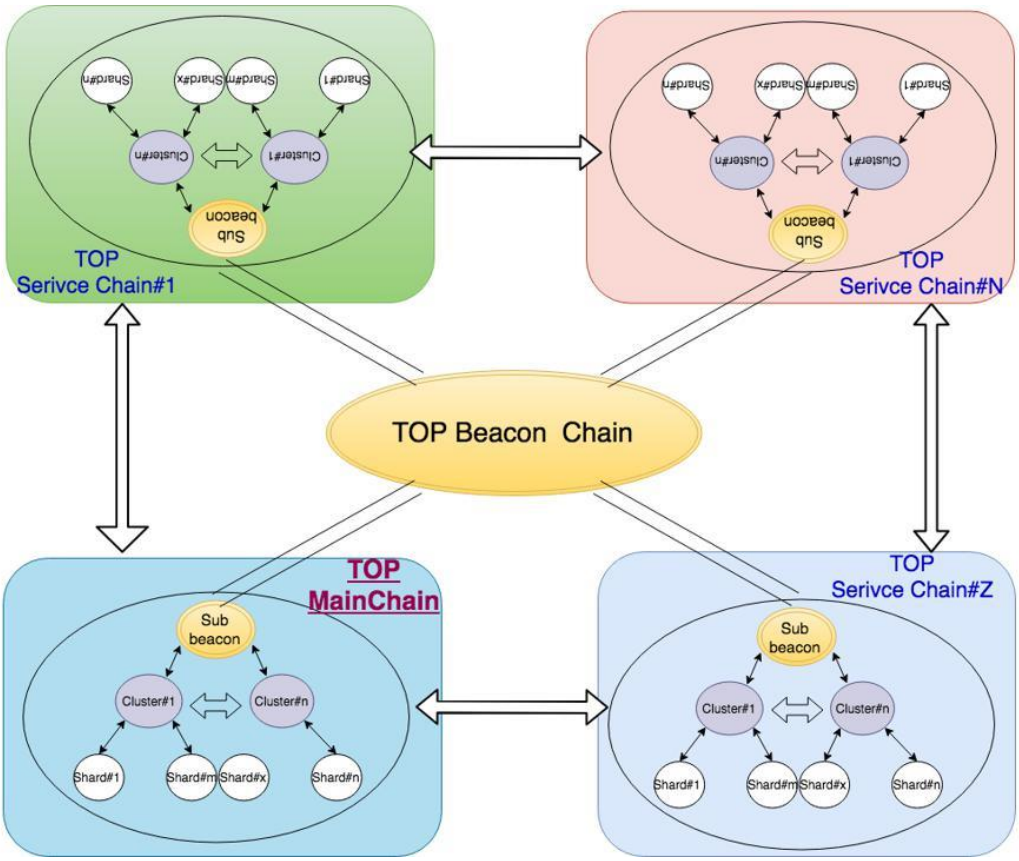


Fig. 5. TOP Chain Structure

3.3.2 The Beacon Chain

Like some other sharding architectures, we use a Beacon Chain as part of our design. However, unlike other Beacon Chain implementations, TOP's Beacon Chain is very lightweight, and is not involved in confirming the result of the consensus process through methods such as cross-linking. This is the case for Beacon Chains in other sharding projects, which results in them becoming bottlenecks in the system, as all Shards ultimately rely on a single chain to help confirm transactions.

3.3.2.1 Beacon Chain Purposes

TOP's Beacon Chain has numerous roles, acting as the coordinator and archive for the system. The Beacon Chain handles node registration and elections, along with staking, voting, slashing, and work-load logging. These parameters and processes are logged and handled through Beacon Chain smart contracts. The Beacon Chain also acts as a global clock for the whole system through the production of timing blocks at regular intervals.

3.3.2.2 Beacon Chain Nodes

In many sharding systems, the Beacon Chain is run by nodes who must specifically register to act as Beacon Chain nodes. This can result in a shortage of Beacon Chain nodes, as usually most of the mining rewards are distributed to nodes processing transactions. This can ultimately lead to centralization of the Beacon Chain.

On TOP Chain, every node is a potential Beacon Chain node by default. The nodes with the highest stake have a higher probability of selection via VRF-FTS algorithm, which means Beacon Chain operations are generally the most secure.

To add Beacon Chain blocks and execute Beacon Chain smart contracts, a subset of 256 nodes from the entire pool of nodes is randomly selected for each consensus round.

3.3.3 VRF-FTS Random Partitioning

Security is one of the biggest challenges involved with sharding. By dividing the network into smaller groups, it becomes easier for malicious entities to take control of shards in what's called a Single-Shard Takeover Attack. To prevent this, the sharding process is done randomly. With random sharding, ma-licious entities are not able to direct their nodes into any specific shard, making an attack much more difficult.

To generate randomness in a decentralized manner, TOP makes use of a Verifiable Random Function. VRFs are cryptographic constructs which allow for the creation of unbiasable and publicly verifiable random seeds. These random seeds generated via VRF are used along with a weighted Follow-The-Satoshi (FTS) algorithm to sort Validators into shards, and Advanced Nodes into Clusters.

3.3.4 Dynamic Sharding

While random partitioning helps prevent attackers from directly inserting their nodes into a particular shard, it is not sufficient to prevent adaptive adversaries from slowly corrupting a shard. For instance, it is possible that given enough time, a malicious entity could bribe all the nodes in a particular shard. If the node makeup of a shard remains static, this type of bribing attack is feasible. To prevent this, TOP employs dynamic sharding. Every so often, a few nodes from each shard are reshuffled into other shards. Over time, shards will have completely different nodes then they had previously. As only a few nodes are shuffled around at a time, consensus can continue uninterrupted while the newly moved nodes sync to the state of their new shard.

This same process occurs in the Routing/Audit Network as well, where Advanced Nodes are continu-ously shuffled between Clusters. This two-layer dynamic sharding scheme renders adaptive adversary attacks practically impossible.

3.3.5 Multi-Level Sharding

When it comes to sharding, the goal is to achieve linear scalability. This means that scalability increases linearly with increasing node count. For this to occur, the amount of work each node must do should not strongly depend on the total number of nodes in the system, or the global volume of transactions.

To accomplish this, all of a blockchain's resources must be sharded, including state(storage), computation(transaction validation and smart contract execution), and networking(block propagation, cross-shard communication etc). If, for instance, only computation is sharded, then storage or bandwidth will eventually become a bottleneck.

To reach the goal of a fully sharded system, we developed a novel multi-layered sharding architecture. Each blockchain resource is sharded in multiple ways, which aids in increasing scalability and keeping node requirements low.

3.3.6 Two-Layer State Sharding

The state of TOP Chain consists of all user accounts and all smart contracts. Every user account and every smart-contract are represented by an account object. Each account object contains multiple proper-ties, associated functions, and a mini NoSQL database. This state is sharded in two ways.

First, account objects are divided between shards, meaning nodes within a shard only need to store the state of a subset of the total account space. While this already achieves partitioning of the global state, it is not quite sufficient for a few reasons. Since Validator nodes only store the state of the account subspace associated with that shard, they cannot adequately verify transactions sent from other shards unless they know that the state of the sending account was properly updated.

To account for this, we use table blocks, which store the latest state information of recently changed accounts. There are 1024 table blocks which are divided into the current number of shards. The account space is mapped so that each table block is responsible for an equally sized account subspace. Finding the table block associated with an account can be quickly determined using the hash of the account number. This allows shards to quickly pull relevant state information from other shards and check if balances have been updated properly before committing a transaction.

Table blocks are also used to increase throughput by batching transactions. Multiple transactions from the same account, along with transactions from accounts within the same account subspace, can be included together in one table block. These blocks can be validated in one consensus round, potentially increasing throughput drastically.

3.3.7 Three-Layer Compute Sharding

Computational resources are needed for transaction validation and smart-contract execution, which are both pBFT consensus processes. We use a three-layer design to facilitate secure compute sharding. When a transaction is sent to a shard for validation, a random subset within that shard is chosen to per-form a pBFT consensus check. The rest of the shard monitors the subset and aids with the subsequent synchronization process. Advanced nodes in the governing cluster are also involved in the pBFT con-sensus through a secondary audit process. The concept of three-layer compute sharding will become clearer after reading section 4 on TOP's consensus mechanism.

3.3.8 Three-Layer Network Sharding

The network is divided into three levels. At the top level is Zones, followed by Clusters within the Routing Network, and then Shards within the Core Network. Each Zone is a network of Clusters and Shards, and each Cluster and Shard is also a network. Network operations within each of these levels is confined to a specific network. For instance, network communication within a Shard or Cluster always remains between the nodes in that Shard or Cluster. This three-layer partitioning of the network helps to ensure that bandwidth requirements do not grow significantly as the size of the overall network grows.

3.3.9 Three Layer Consensus Network

The TOP Chain consensus network is divided into three layers: the Edge Network, Routing Network, and Core Network.

The Edge Network acts as the access point for clients. All transactions are first sent to Edge Nodes in the Edge Network before being relayed to the Routing Network.
The Routing Network consists of Advanced Nodes randomly partitioned into Clusters. This layer handles cross-shard communication and synchronization while also acting as a secondary audit network for transaction validation carried out in the Core Network.
The Core Network consists of shards made up of Validator Nodes. This is where transaction val-idation takes places. Within each shard, Validator Nodes validate and confirm transactions using a parallel pBFT algorithm.

A tiered network is used for multiple reasons. First, splitting duties among multiple types of nodes helps keep node requirements low. The Routing Network handles most of the bandwidth heavy op-erations such as cross-shard communication, which then allows Validator nodes to have relatively low requirements. In addition, the Edge Network helps protect the Routing and Core Network from spam attacks, as clients can only send transactions directly to Edge Nodes.

3.4 Service-Chains: Layer-2

Each business has varying needs and workflows which cannot be satisfied by a single chain. TOP in-troduces service chains, which are pluggable chains built for specific use cases. For instance, there will be a VPN service chain, d-storage service chain etc. Businesses can easily deploy their own personal service chain to fit the needs of their application.

By default, each service chain has the same sharded architecture as the main chain, consisting of Shards, Clusters, and a Beacon Chain. However, service chains allow for the execution of complex business logic through application level smart-contracts, while the main chain only handles asset transfer and system level functions.

Service chains are slightly different from the popular side-chain concept. With sidechains, there is a two-way pegging process in which tokens are locked on the main chain, and then freed on the sidechain. With service chains, assets are free to move back and forth to the main chain and other service chains. Each account is able to store multiple types of assets, and directly receive any native token issued on TOP. Additionally, account numbers are globally unique between the main chain and all service chains.

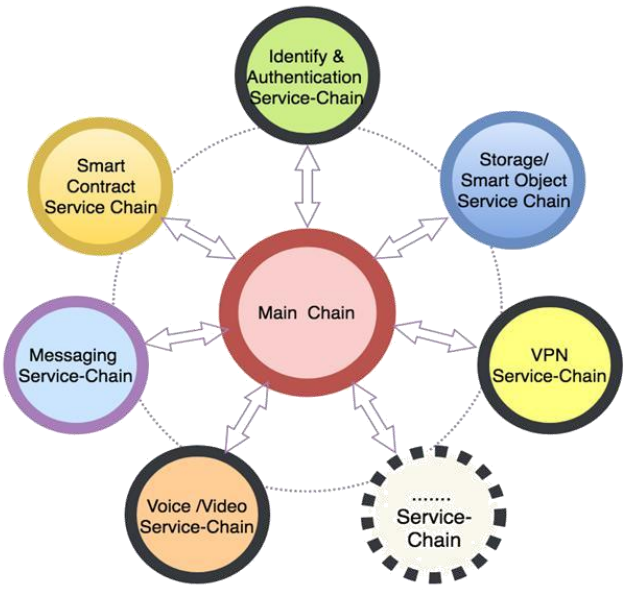


Fig. 6. Service Chains

3.4.1 Service Chain Development Framework

TOP provides comprehensive service chain development and deployment frameworks. Most DApp de-velopers do not have any experience developing blockchains, and so the process of developing service chains for their businesses should be abstracted as much as possible. The TOP service chain devel-opment framework is a turn-key solution allowing developers to create service chains with all of the important features built-in. The service chain development framework includes the following built-in features:

-A P2P network interconnecting the service chain with the main chain and all other service chains.

-An account system which is valid globally between the main chain and all service chains.

-A native token which is recognized between the main chain and all service chains.

-Common asset operations for TOP and the associated service chain native token including: asset lock, unlock or deposit, in addition to transfers between the main chain and service chains.

-Multiple consensus mechanisms to choose from.

-Application level smart contracts, which are integrated with on-chain storage as well as distributed off-chain storage.

Developers can customize the service chain parameters to meet the needs of their business during the application process. For instance, the minimum gas for a transaction, or the minimum number of nodes required for a shard can be configured. For now, developers will choose the underlying consensus mechanism of the service chain from a list, which includes TOP's hpPBFT-PoS* consensus mechanism, or other consensus mechanisms like PoW. In the future, bigger businesses may aim to develop their own customized consensus mechanisms.

3.4.2 Service Chain Deployment Framework

To actually deploy a service chain, TOP provides a one-click deployment framework. Developers need not worry about the underlying details involved with deploying a service chain and can instead focus on the high-level functionality. All developers need to do is submit an application by calling a dedicated smart contract deployed on the Beacon Chain. Along with the application, developers need to deposit a certain amount of TOP Tokens, and also specify the configurations parameters of the service-chain, including things like which consensus mechanism will be used, the minimal amount of nodes in a shard, etc. After successfully submitting an application, the Beacon Chain smart contract will issue a Chain-ID along with the service chain native token by broadcasting to all existing service chain and main chain nodes.

TOP's service chain framework goes further by addressing one of the main problems with sidechains and similar concepts, which is garnering nodes. After completing the application process, the TOP elec-tion contract on the main Beacon Chain will bootstrap the service chain by assigning nodes to act as the Beacon Chain nodes of the service chain. Once the service chain Beacon Chain begins running, it will gain all the functionality of the main Beacon Chain, including core system level smart contracts like the election contract, asset operations, and so on.

Subsequently, the service chain Beacon Chain can accept node registrations, and begin assigning new nodes to Shards and Clusters in a similar manner to the main chain. These nodes will validate and execute service transactions and application level smart contracts, which run the necessary business logic. While the main chain can help bootstrap a service chain by providing the initial core Beacon Chain nodes, to gain more nodes the service chain will need to host useful applications. If the service chain is popular, it will be easier to obtain more nodes, while if it has no transaction volume, it may be harder to convince additional nodes to join.

3.5 One-Way State Channels: Layer-2

In the past several years, there has been much discussion revolving around layer-2 scaling. Numerous projects have been working on technologies like state-channels for years. However, layer-2 solutions have yet to really take hold, mostly due to the difficulty of implementation. In terms of state-channels, the crux of the issue is that current attempts are aiming for total generality. While this seems ideal, it brings about many challenges and inefficiencies that make actually using state-channel solutions infeasible.

TOP implements built-in state-channels with the goal of increasing throughput for specific uses cases. In particular, TOP's state-channels are only used for applications which involve low-value high-frequency micro transactions. The state-channels are built atop service chains, and slightly differ based on the specific use case. We'll use the VPN service chain to demonstrate how TOP's state-channel solution works.

The participants in a VPN service chain transaction are a VPN client, a VPN service provider, and VPN Edge Node relays. When a VPN session is initiated, these actors form a virtual consensus network. Deployed on this virtual consensus network is what we call a virtual smart contract, which is stored locally by each node for the duration of the session. The virtual smart-contract can pull information from the VPN service-chain, such as the current balance of the VPN client.

A VPN session is billed based off of billing units, which are usually around 1Mb of bandwidth. This is an exceedingly small amount in terms of value. Executing a transaction for each billing unit could require hundreds of transactions for just a single session. Instead, the virtual smart contract keeps track of the number of billing units consumed during a particular session, and at the end submits a single transaction to the service chain. The client, service node, and Edge relays will periodically submit to the virtual smart contract the number of billing units consumed or relayed. The virtual network consisting of session participants will perform a small consensus check using what we call Proof-of-Bandwidth (PoB).

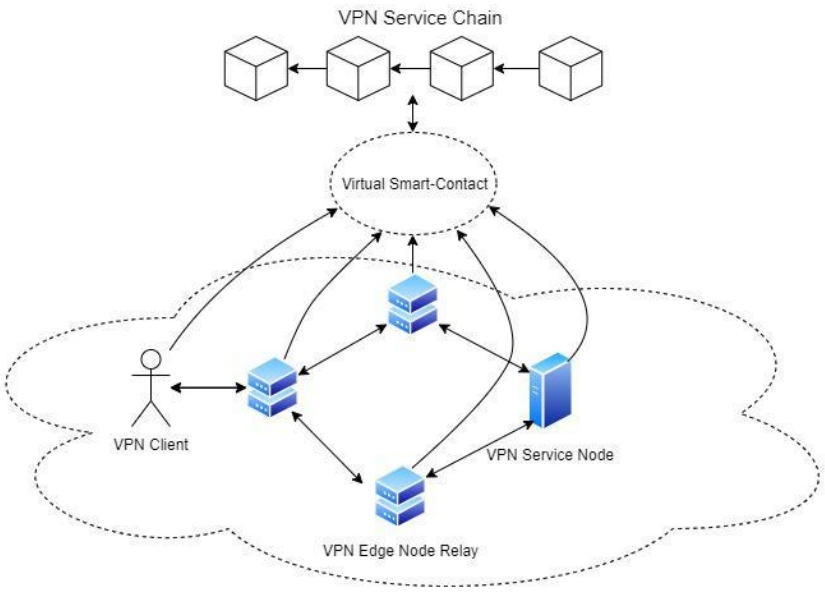


Fig. 7. VPN State-Channel Session

The major difference between TOP's state-channels and those of other projects is the settlement and dispute process. In usual state-channel implementations, the settlement and dispute process are complex. Each participant in the state-channel must agree completely on every state transition. If one of the participants goes offline, or there is a disagreement, the channel cannot be settled until there is a resolution. In this scenario, users are presented a "challenge window," where they can submit proof of the state, they think is valid. This window of time can be relatively long, which can negate the reduction in latency gained from the state-channels in the first place.

TOP forgoes this dispute process by forming one-way state-channels. Essentially, the client has the final say. If a discrepancy is uncovered from the consensus check, the state-channels is closed, and the client pays the amount last logged in the virtual smart-contract. So, for instance, if the client submits that it used 90Mb, while the service-node submits a contribution of 92Mb, the client will only pay for 90Mb. Since each billing unit is so small in value, the service-node does not stand to lose any significant amount. The discrepancy in which the channel will close is configurable by the service node. For instance, if the service provider is a big data center with excess bandwidth, it may be content in losing a few extra Mb in the case of dishonest client. An individual service provider may be more stringent and will close the channel as soon as there is a discrepancy of just a 1 or 2 Mb.

Each service chain has slightly difference consensus checks, but the general process is the same. State-channels help to vastly increase scalability. Since this functionality is built-in, developers do not need to fuss with integration and can instead simply enjoy the speed increases.

3.6 Distributed Storage

Blockchains are very inefficient databases. Storing a large file on a blockchain will always cost far more than a centralized counterpart, largely due to the extreme redundancy blockchains employ. However, almost all applications produce and must store large amounts of files or data. Using a blockchain for all storage needs would be prohibitively expensive for DApp developers and would increase the size of the blockchain astronomically. This is why many DApps on other blockchain systems resort to using centralized storage solutions. To account for this, TOP provides distributed storage options which allow large amounts of data to be stored in a decentralized manner, without storing everything directly on-chain.

3.6.1 Distributed File Storage

Applications developers often need to host large files which cannot be uploaded to the blockchain. TOP provides a distributed file storage network similar to IPFS as part of the core infrastructure. As with IPFS, files are segmented and encrypted and then stored redundantly across several nodes. Hashes of these chunks are stored on-chain to ensure authenticity and to prevent tampering. TOP's networking infrastructure is well suited for this kind of application, as this type of distributed file storage makes heavy use of DHTs to store and access file chunks.

3.6.2 Distributed Database

File storage is not quite the same as a database. Databases store related data in a structured format. TOP's infrastructure includes a distributed database system allowing DApp developers to store more complicated forms of data in a decentralized manner. Each account has an associated mini NoSQL like database stored on-chain. These mini databases can store multiple complex data structures, such as Hash Maps, Lists, and so on.

3.7 Token Economics

The monetary policy of the TOP ecosystem was designed with real-world considerations in mind. As with all real-world monetary policies, there are mechanisms for both inflation and deflation in the TOP ecosystem. On-chain governance gives the community a say in the specific parameters used, as should be the case in a decentralized, democratic system.

In regard to the fee structures, the token economics on TOP were devised with several general design ideas in mind. From a high level, the ultimate goal of TOP's token economics system is to be very developer and user friendly. To accomplish this, we made ordinary Shard transactions as free as possible, while it was determined that Beacon Chain transaction need fees to prevent spam attacks. As far as regular users, the concepts of disk and gas resources should be abstracted as much as possible.

While free shard transactions are desirable, it introduces some extra challenges. The main issue is the use of system storage resources as a result of spam transactions. To account for this, we developed several deterrents, including a system where the user who creates the transaction data owns it and must pay for it.

3.7.1 Token Supply

The total initial max supply of TOP tokens is 14.4 Billion. However, this number changes from both inflationary and deflationary mechanisms. These act to produce a healthy monetary policy, which is not overly inflationary or deflationary.

3.7.1.1 Mining Rewards

Block rewards for mining nodes are produced through inflation. Each block produced by the miners has an associated award, usually called the block reward. The annual number of tokens allotted for mining rewards each year is determined as a percentage of the total supply. Every year, the number of tokens produced for mining rewards decreases. The exact numbers for mining rewards are configurable through on-chain governance.

3.7.1.2 Burning Mechanisms

To offset the inflation associated with mining rewards, TOP adds several means of burning tokens. Burning means to remove tokens from the supply forever, which is a deflationary mechanism. There are several instances of this in the TOP ecosystem.

On TOP, users can issue TEP-10 Native tokens. Unlike contract tokens, these tokens live on-chain in a similar manner to the main TOP token. If TOP token is the “world reserve currency,” TEP-10 tokens are like smaller regional currencies traded against TOP. To issue a TEP-10 token, the user must deposit a sum of TOP tokens. These tokens are burned, thus lowering the total supply.

Transaction fees are also burned. In other blockchains such as Bitcoin and Etheruem, one of the the main source of miner income is transaction fees. On TOP, miners receive enough rewards from tokens issued for block rewards, and so all transaction fees can be burned.

For most transactions, there are no fees. However, TOP does charge accounts handling fees when sending some types of special transactions to the Beacon Chain. For instance, things like node registration, which require Beacon Chain transactions, charge a small handling fee which is burned. As for regular transactions, if an account runs out of gas, a small fee is charged and burned to offset the missing gas requirement.

3.7.2 Minimum Balance Requirement

TOP accounts must have a minimum balance of 0.1 TOP token before sending transactions. Additionally,if the account balance is less than 100 TOP, no free transaction can be sent; account balance more than 100 TOP, you can send limited free transactions. Again, this is to prevent attackers from creating many accounts and then sending spam transactions from these accounts.

3.7.3 System Resources

Transactions on TOP Chain can be divided into Shard transactions and Beacon Chain transactions. Both types of transactions consume on-chain resources, and there are two types of resources on TOP Chain. The first is temporarily occupied resources such as NET / CPU / RAM, which are unified and abstracted

into one type of resource called gas. The other major resource is DISK, which is a long-term occupied resource.

Each account—including each contract account and each ordinary account—has free daily gas and DISK resource quotas. To obtain higher daily quotas, the account must deposit TOP tokens. Gas re-sources are restored to the maximum amount every 24 hours. Disk resources can be restored only when users delete the associated data stored on the chain. TOP tokens deposited to obtain gas resources can be redeemed at any time and are credited to the account 24 hours after initiating the redemption. TOP tokens deposited for DISK space can only be redeemed in return for deleting the associated disk space.

For Beacon Chain transactions, in addition to consuming gas and disk resources, there will be a handling fee, which will be burned. This small fee is necessary to protect the Beacon Chain from spam attacks.

3.7.4 MaxTxFee

The MaxTxFee is the maximum fee an initiator is willing to be charged for a transaction. When an account does not have enough gas to send a transaction, TOP is deducted from their MaxTxFee balance—which must be greater than or equal to 0.1 TOP—to offset the gas costs. This fee is then burned. The GAS to TOP conversation ratio is currently set to:

$$1\ GAS = \frac{1}{10000}\ TOP$$

The minimum MaxTxFee parameter can be changed via on-chain governance. Most users will have enough free gas for their daily needs, while those who send many transactions may run out of gas and need to pay an amount up to the small MaxTxFee.

3.7.5 GAS Consumption Rules

As mentioned, each account on TOP is either a regular user account or a contract account. There are different rules for gas consumption depending on whether regular accounts or contract accounts are involved.

1. For transactions from ordinary account to ordinary account, only the initiator is charged.
2. For transactions from ordinary account to contract account, the initiator will be charged first, and the receiver's contract account can set the proportion of gas assumed.
3. For transactions from contract account to contract account, the initiator will be charged first, and the receiver's contract account can set the proportion of gas assumed.
4. For transactions from contract accounts to ordinary accounts, only the contract account will be charged.
5. The system's native contracts are free of charge whether acting as a sender or a receiver.

3.7.5.1 Free Daily GAS

Be default, each account with a balance of at least 100 TOP is given a free 25000 TGAS every 24 hours. This allows for several free transactions per day. A minimum TOP balance is necessary to avoid mass account creation and mass transaction spamming using the free allotted gas.

3.7.5.2 Gas Acquisition

Ordinary accounts as well as contract accounts can deposit TOP tokens to gain more gas credits. The formula for gas credits is as follows

$$\frac{\textit{TOP Tokens Deposited}}{\textit{Total TOP Tokens Deposited in System}} * \textit{System Total Gas}$$

Where the System Total TGAS = 10,800,000,000 * number of shards. This value can be adjusted via on-chain governance. The number of shards is currently set to 4, and the total gas of the system will increase as the number of shards increases. The total number of TOP tokens deposited in the system is updated every ten minutes, and the value is stored inside a native contract.

The quota for an account is the maximum amount of gas that account can use in a 24-hour period.
The quota does not decrease as the account consumes gas .

3.7.6 DISK Consumption Rules

The rules for DISK consumption based on the types of accounts involved are the same as for gas, only the underlying resource is different. The basic unit of DISK is bytes of storage. Accounts can obtain DISK resources by depositing TOP tokens. The DISK to TOP ratio is tentatively set to 0.00004 TOP / byte, although this can be changed via on-chain governance.

Processing transactions requires the use of DISK resources. The DISK usage of a transaction is calculated by summing the disk space occupied by the original transaction, along with the status change of all Unit blocks related to the transaction. The change in Unit blocks generally refers to the change in property data caused by the transaction.

3.7.6.1 Free DISK Quota

Each account with a balance of at least 100 TOP is given a one-time free 500KB of disk space. Unlike gas, this quota does not refill every 24 hours, as disk space is a long-term resource. Again, a minimum TOP balance is necessary to avoid mass account creation and mass transaction spamming to take up node disk space.

3.7.6.2 Redeeming DISK

Deleting data can release the associated tied up DISK resources. Ordinary accounts can decide how long a transaction record under their account or contract account should remain before being deleted. Accounts can also reduce the amount of account property data stored on-chain, so as to reduce the size of their own accounts or contract accounts to save DISK resources. The addition of DISK as a fundamental blockchain resource forces high-volume users and DApps to use on-chain storage responsibly.

Regular accounts can redeem their TOP tokens used as DISK collateral, and DISK resources used in contract accounts can be redeemed by the contract owner. After a successful redemption application, the account owner will receive the TOP tokens deposited as collateral for DISK resources after a thawing process which lasts 24 hours.

3.7.7 Issuing TEP-20 and TEP-10 Assets

TOP allows developers to issue multiple types of assets. Analogous to ERC-20 tokens on Ethereum, DApps can easily issue TEP-20 contract tokens. These tokens live within the contract and are most useful for things like in-game currencies, points, or credit systems. These token contracts can be deployed on either a service chain or the main chain.

TOP also allows businesses to issue their own TEP-10 native tokens. Unlike TEP-20s, these tokens are globally recognized throughout TOP Chain, and once issued, any account can directly receive TEP-10 native tokens. In contrast with TEP-20 contract tokens, developers must deposit TOP tokens in order to issue TEP-10 tokens. The processes are facilitated by a token issuance contract deployed on the Beacon Chain. After sending an appropriate amount of TOP tokens along with data such as the token symbol, total supply etc. to the issuance smart contract, the information will be broadcast to every node on TOP Chain. The precise details of disbursement can be coded into a service-chain smart contract.

3.8 TOP Chain Governance

TOP Chain employs a decentralized, democratic on-chain governance model. The goal of on-chain governance is to provide a secure way for the community members most involved, along with developers and miners, to make changes and steer the direction of the ecosystem. Those with stake in TOP Network can submit proposals, and subsequently vote on proposals or on-chain parameter changes.

3.8.1 TOP Chain Governance Participants

Only those who have stake in the TOP ecosystem can participate in on-chain governance decisions. The reason being that those with something to lose will make the most informed and beneficial decisions. Any token holder or miner who deposits and locks TOP tokens for on-chain resources, vote staking, or node deposit requirements, is allowed to participate and cast votes for on-chain governance matters.

3.8.2 TOP Community Board

The core of TOP's governance structure is the TOP Community Council (TCC). The TCC's responsibility is to vote on community proposals. Only proposals that have passed a majority vote from the TCC will be submitted to the chain for on-chain voting. Once passed by referendum, the TCC is also responsible for implementing the proposal, which might include modifying the relevant parameters on the chain. The TCC is composed of 7 Council Members, one of whom serves as the convener, and is given the name of Chief Council Member.

3.8.2.1 TCC Member Elections

During the first five years following the launch of TOP's Main chain, a TOP Foundation representative will be designated as Chief Council Member, while the remaining six TCC members will be democratically elected on-chain by all voters in the ecosystem. After five years, the TOP Foundation representative will be removed, and

all of the TCC members will be elected on-chain. The member who gets the most votes will serve as the Chief Council Member. TCC elections occur once a year.

Any qualified voter can participate in the election and vote for a TCC Council Member. The seven candidates (six in the first five years) with the most votes are elected as members. Candidates running for a TCC member position must first deposit the same number of TOP tokens required of an Audit Node. If the candidate fails to be elected, the deposit will be refunded immediately.

3.8.2.2 TCC Member Rewards and Punishments

TCC members will be returned their locked deposit in full after they leave office and receive a reward equal to the deposit. However, if TCC a member processes an insufficient number of proposals (less than 2/3 of the total), they won't receive any reward, and part of their deposit will be slashed, according to

$$\textit{Slashed Amount} = \textit{Deposit} * \frac{\textit{Number of Outstanding Proposals}}{\textit{Total Proposal}}$$

If a member is impeached, there will be no reward and the deposit is fully deducted.

3.8.2.3 Impeachment of TCC Council Members

The community can initiate an impeachment vote of a TCC member at any time to avoid inaction and incompetent or malicious members. If an impeachment vote exceeds 66.7% of the total staked TOP tokens, the TCC member is impeached and immediately removed.

3.8.2.4 Adjusting the Number of TCC Members

During the annual TCC member election process, the community can initiate a referendum to adjust the number of TCC members. If the referendum is supported by more than 66.7% of votes, the adjustment of the numbers of TCC members will take effect. However, the number of TCC members cannot be less than seven, or more than one hundred.

3.8.3 Source Code Control

The TOP Developer Working Group is responsible for maintaining the core source code of TOP Network. The source code submitted by a community member can only be checked into the code base after being reviewed by the TOP Developer Working Group. The TOP Developer Working Group also outlines the technical development roadmap of TOP Network and publishes development tasks to be completed by the wider developer community.

The members of the TOP Developer Working Group are named Core Developers, three of which are called Lead Developers. A representative designated by the TOP Foundation always serves as one Lead Developer, while the remaining two Leaders are elected by members of TOP Developer Working Group and are elected once a year.

Candidates applying to join the TOP Developer Working Group must have contributed to the development of TOP, that is, his/her source code has been submitted and accepted in the past. The applicant must receive at least one Leader's endorsement before being voted in by the TOP Developer Working Group and obtain the consent of more than two-thirds of existing Core Developers to become a Core Developer.

3.9 Security

Security is of paramount importance when designing a blockchain. As such, TOP was built with an abundance of mitigation techniques to ensure security of the system.

3.9.1 Transaction Flooding and DDoS Attacks

In DDoS attacks, attackers may send a large number of transactions from accounts under their control to flood the whole system and thwart legitimate transaction processing. Such attacks seem possible, as TOP Chain does not charge fees for most general transactions, making it nearly costless for attackers to submit a large number of transactions. TOP Chain has implemented several anti-attack strategies to account for this. The goal is to make it very expensive for an attacker to launch attacks by sending high volumes of transactions.

3.9.1.1 Minimum Balance

As mentioned in 3.7.2, TOP accounts must have a minimum balance of 0.1 TOP token before sending transactions. Additionally, an account must maintain a minimum balance of 100 TOP to receive free daily gas resources and receive the one-time allotment of DISK space. If the user goes over these quotas, additional TOP tokens must be deposited before sending anymore transactions. These requirements significantly increase the cost for an attacker to generate a large number of accounts and subsequently send a large volume of transactions in order to flood the system.

3.9.1.2 Locked Assets

New assets received from a transaction will be locked for a certain period, during which the asset cannot be transferred. The lock-up period, a parameter set by the system, depends upon the value of the asset. Assets with higher value have longer lock-up times, which prevents circular transfer attacks and ensures the safety of high-value transactions.

3.9.2 Sybil Attacks

In a Sybil attack, malicious entities attempt to create or control hundreds or thousands of nodes to sabotage the security of the consensus network. TOP uses a special type of Proof-of-Stake (section 4.3), to prevent Sybil attacks from occurring.

3.9.2.1 Minimum Deposit Requirement

Each node needs to have an independent account and make a minimum deposit into an election smart contract on the Beacon Chain before joining the consensus network. Minimum deposit requirements are different for Edge Nodes, Validator Nodes, Archive Nodes and Audit Nodes. The more important the role of a node, the higher the minimum deposit. As a result, it is costly for an attacker to create and own many nodes in the hopes of launching an attack.

3.9.2.2 Admission Requirements and Random Sharding

After a node meets the minimum requirements, its odds of being selected to actually participate in consensus depends upon the Comprehensive Stake (4.3.1). The Comprehensive Stake integrates a form of reputation with the deposit amount, which requires attackers to both accumulate a large sum of assets and stay in the system for

a long period of time with good behavior before launching an attack. These methods significantly increase the cost of an attack. Moreover, TOP Chain uses a VRF-FTS algorithm to randomly select nodes to form Shards and Clusters from the candidate node pool. This reduces the probability that a malicious attacker can direct all of its nodes into a particular Shard or Cluster.

3.9.3 Penny-Spend Attack

In penny-spend attacks, attackers send an enormous number of low-value transactions to a large number of accounts, preventing the system from processing legitimate transactions and wasting the disk space of the consensus nodes. The economics restrictions on high-frequency transactions and the minimum deposit requirement mentioned above can prevent most penny-spend attacks. Additionally, after using the free allotted DISK space, additional DISK space must be acquired through depositing TOP tokens to send and permanently store new transactions

Transactions are stored in the Unit Lattices, where old transaction data in accounts can be pruned in real-time as the system processes new transactions, significantly reducing the storage space for accounts and making it unlikely for penny-spend attacks to exhaust the storage space in nodes.

3.9.4 Attacks on A Single Shard

In attacks targeting a single Shard, attackers try to locate a Shard in which a specific transaction occurs and then attempt to launch DDoS attacks on the Shard, modify transactions, or create fake transactions. TOP Chain prevents such attacks through several measures.

3.9.4.1 Edge Network

The Edge Network separates users and clients from the Core Network so that they cannot directly connect to the Core Network to retrieve account and transaction information from consensus nodes.

3.9.4.2 Random and Dynamic Sharding

The VRF-FTS algorithm randomly selects consensus nodes to form shards based on Comprehensive Stake. In addition, nodes dynamically rotate between different Shards and Clusters to serve different sets of account spaces. When validating transactions, each shard randomly selects its pBFT leader. Such three-layer dynamic randomness makes it difficult for attackers to identify in advance the specific Shard and/or node processing a specific transaction, minimizing the chance of malicious nodes colluding in advance and launching DDoS attacks on a specific node.

3.9.4.3 Double Auditing

In order for a transaction to be confirmed, the transaction is first validated in a Shard, and then further audited by a group of Audit Nodes in the Audit Network. Consensus and Audit node groups are selected randomly at the transaction level, so the attacker cannot anticipate which nodes will be responsible for validating or auditing the transaction. In this way, malicious nodes in a Shard or Cluster, if any, cannot deceive the system into accepting fake transactions.

3.9.4.4 Shard Merge

When a Shard suffers a DDoS attack, its surviving nodes are so few that the Shard cannot work securely. In this case, the Shard which has been attacked will automatically be merged into another shard to form a larger, more secure Shard.

3.9.5 Double Spend Attack

TOP Chain organizes transactions into Unit Lattices, where transactions initiated by an account are stored on a chain dedicated to that account. Therefore, transactions from each account will form their own micro chains. Each valid transaction of an account contains a unique nonce that monotonically increases, as well as the hash value of the previous transaction that the system has confirmed. In addition, when a transaction is verified, the account balance is updated in real-time, allowing each consensus node to discover and reject double-spend transactions immediately.

3.9.6 Smart Contract Security

TOP Chain offers two types of smart contracts, namely Platform Contracts and Application Contracts, which are executed on independent and isolated virtual machines. Platform contracts, written in non-Turing complete scripting languages, handle account assets on the main chain. In comparison, Appli-cation Contracts are written in Turing complete languages and thus are very flexible. However, Appli-cation Contracts are restricted to running on the independent service chains, which protects the main chain from asset loss caused by various possible loopholes in Application contracts.

3.9.7 Single Node Attack

In single node attacks, malicious nodes tamper with transactions or refuse to process transactions. How-ever, even if one-third of the Validator nodes in a Shard are malicious, the pBFT consensus algorithm can still guarantee that the remaining honest nodes execute the consensus process correctly for transactions. Nodes that fail to process transactions in time are replaced by candidate.

Transactions must be verified by Validator nodes, and also audited and signed by a group of Audit Nodes before obtaining final confirmation, which allows Audit Nodes to discover malicious nodes in a Shard. The system selects Audit Nodes randomly at the transaction level, making it impossible for Validator nodes to identify in advance the audit-conducting nodes, thus preventing malicious collusion between Audit Nodes and Validator Nodes.

3.9.8 Replay Transaction Attack

In a replay transaction attack, an attacker intercepts transaction packets on the transport layer in the operating system and saves the transactions for replay later. This attack is a variant of DDoS and Double-Spend attacks. TOP Chain prevents replay transaction attacks with the following strategies.

1. Each account has a monotonically increasing nonce to identify every transaction. Transactions with the same nonce initiated by the same account will be rejected.
2. Each submitted transaction needs to contain the hash value of the latest valid transaction in the account. All transactions in the same account are organized into a Unit Lattice in chronological order of creation, which allows Validator nodes to determine easily whether an incoming transaction is a duplicate.

4 Consensus Mechanism: Parallel pBFT-PoS*

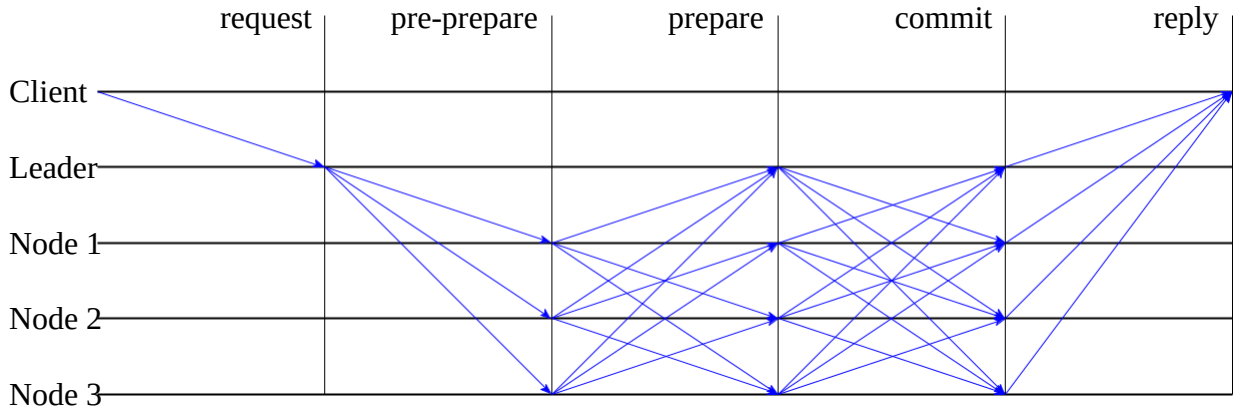
The main directive of a Consensus Mechanism is to allow a network of nodes to reach an agreement regarding the validity of a particular state transition. As it relates to blockchain technology, that would be the addition of a new block, or the execution of a transaction.

Practical Byzantine Fault Tolerance (pBFT) is an alternative consensus mechanism to the more widely used PoW-Longest-Chain algorithm that is implemented in the Bitcoin and Ethereum blockchains. pBFT has actually been around for decades, but in recent years there has been renewed interest in its use surrounding blockchain technology, which has already led to big improvements in the protocol. While pBFT has many advantages over PoW, it has some serious drawbacks in its original form. Before diving into the details of TOP's improved version of pBFT, it is helpful to briefly understand the pitfalls of previous iterations.

4.1 Previous pBFT Iterations

In pBFT consensus algorithms, there is always a special node called the leader, along with many val-idator nodes (also called replicas). Traditionally, each round of the pBFT protocol—which is called a view—consists of three main phases: pre-prepare, prepare, and commit. In the pre-prepare phase, the leader assigns a sequence number to each request and multi-casts the message to each replica. After val-idating the pre-prepare message and casting their vote, each node multi-casts a prepare message to every other node. If a node receives 2 f + 1 prepare messages matching the pre-prepare message stored in their log, the sequence number for the request is agreed upon. In the commit phase, each node multi-casts to every other node a commit message. If 2 f + 1 matching commit messages are received signifying that 2 f + 1 nodes agree that at least 2 f + 1 consistent votes were received, the request is executed.

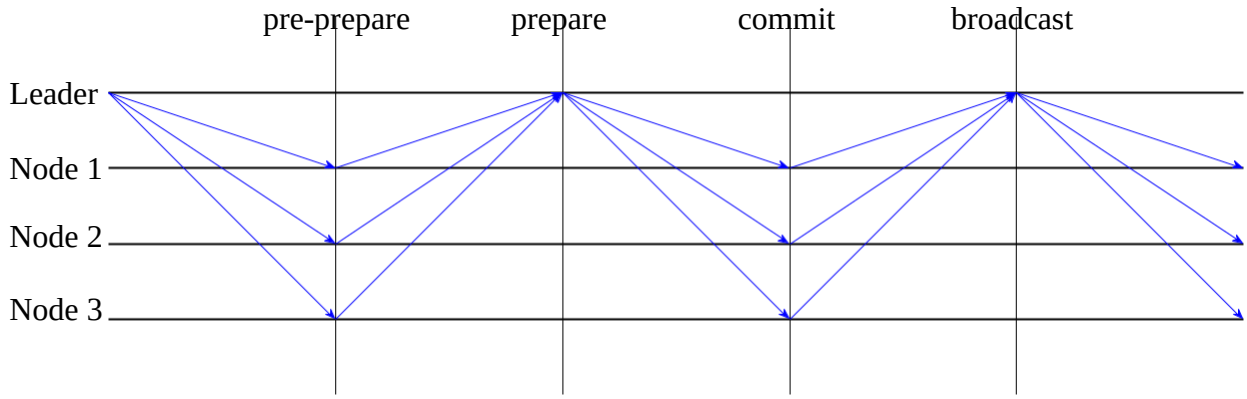
As a consensus mechanism in the BFT family, with a total of 3 f + 1 nodes, the protocol can tolerate up to f faults. The process is as follows:



This is the traditional version of pBFT. Like the many BFT protocols which have followed, it uses a two-phase commit paradigm. While this form of pBFT is very secure, it has a major drawback in terms of communication overhead. Since each node must send every other node a message multiple times per round, the communication complexity is O(n2). Worse yet, leader replacement in the case of failure is O(n3). As a result, the number of nodes in a pBFT group must remain small, which is not suitable for a decentralized blockchain system.

4.1.1 Leader-Based BFT Protocols

Subsequent BFT protocols have accounted for the high communication complexity of traditional pBFT by significantly simplifying the protocol. Instead of each node multi-casting to every other node, the leader receives a partial signature from each node, aggregates them into a single multi-signature, and then broadcasts the result to every node. This reduces the communications complexity to O(n), which is a vast improvement. However, a new issue is introduced. In this type of scheme, the leader becomes pivotal, which greatly reduces security, and can lead to liveness problems. If the leader becomes malicious or drops, the protocol can halt. With a few colluding nodes, the leader could even potentially pass an invalid state transition.



The remedy to this flaw is what's called a view-change protocol. If the leader's network drops, or the leader attempts to act maliciously, a view-change is initiated, and the protocol restarts with a new leader. While this is a partial solution, view-change protocols are very complex. When the network needs a view-change, the replicas must run a separate process to come to an agreement on the last valid state, reorder transactions, and then restart consensus. The complexity of this process can be as high as O(n3) in traditional pBFT, to O(n2) in more recent optimized versions using multi-signatures. Furthermore, there are typically numerous associated edge cases, and failure to account for every scenario could be catastrophic.

4.1.2 VRF-Time Based View-Changes

In some of the latest versions of pBFT, complex view-change protocols are simplified to VRF time-based view changes. Instead of initiating a view-change only when something goes wrong, a leader is replaced automatically after a certain timeout period, and the next leader is elected randomly in a verifiable manner via VRF. Still, the pBFT leader has a great deal of power during its tenure and can perform certain attacks during the time it is elected. Because of this, pBFT has a problematic trade-off. Because of issues surrounding malicious leaders, pBFT committees must be relatively large to reduce chances of collusion. However, due to the high levels of communication required between validators, scalability decreases sharply as the pBFT groups become large.

4.2 Highly Parallel pBFT (hpPBFT)

TOP Network's pBFT inspired algorithm, dubbed hpPBFT, adopts a three-phase commit stratagem which permits O(n) communication complexity with O(n) view-change complexity. The algorithm uses VRF time-based view changes, but adds additional security through a secondary audit network, and increased scalability through parallelization.

4.2.1 Secondary Audit

With small consensus groups, the version of pBFT described in 4.1.2 is very fast under normal operation conditions. However, the view-change complexity is either O(n2) or O(n3) depending on the implementation, which means using very large groups can cause the overhead of running a node to increase, while also decreasing performance. The problem is that using small groups in such a protocol can be dangerous, as the leader can perform several attacks with relatively few colluding members. In short, there is a trade-off between security and scalability.

TOP's hpPBFT avoids this issue by making a change to the protocol which lowers the view-change complexity to O(n), while also increasing security. The view-change process is simplified by using a 3-phase commit paradigm instead of the usual 2-phase commit. While inserting an extra phase adds a small amount of latency, it is well worth it to achieve linear leader replacement complexity.

Security of the protocol is increased by introducing an audit committee and leader from the audit network. During each view, the audit committee performs their owns checks along with the original pBFT group. As a result, consensus does not rely too heavily on the original pBFT leader. Even if a shard is compromised or the leader is malicious, the audit network will prevent any invalid state transitions.

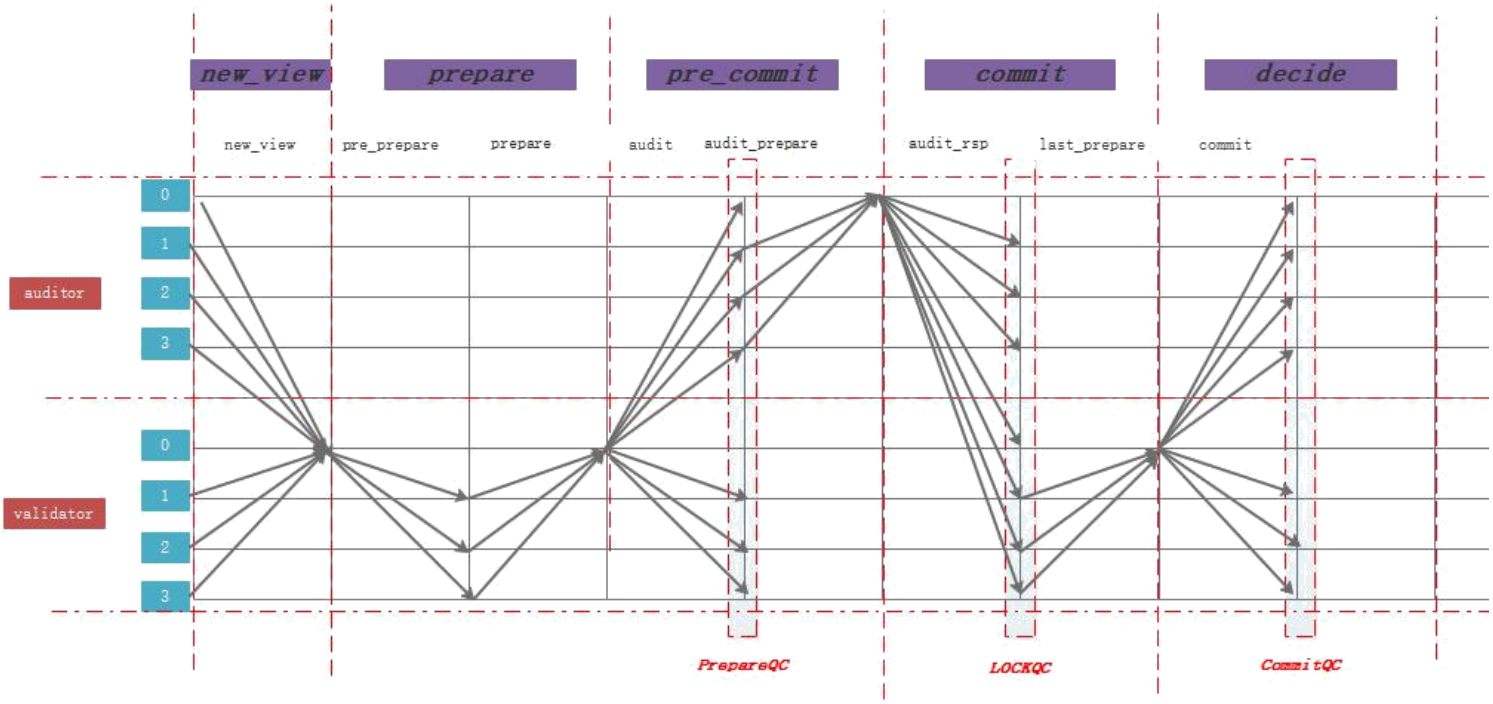


Fig. 8. hpPBFT 3-Stage Consensus

4.2.2 Completely Parallel Consensus

TOP applies several techniques to produce a high degree of parallelization. Besides the obvious parallel nature of sharding in general, TOP implements several other forms of parallelization.

4.2.2.1 Node Virtualization Over Multiple CPU Cores

In our design, each node can take on multiple roles simultaneously. For instance, an Advanced node can perform the duties of a Validator node concurrently. The necessary code for each type of node is assigned to run on separate CPU cores. This expands the total useful work a node can do, and essentially increases node count, thus increasing overall scalability. It also helps to make shards more secure, as Advanced nodes which have higher staking requirements can act as Validators, making it harder for attackers to gain the majority of stake within a shard. To be eligible for this functionality, a node needs to meet the minimum deposit and voting requirements set for Advanced nodes.

4.2.2.2 Parallel Subset Committee Consensus

As with all sharding designs, each shard can perform transaction validation in parallel with other shards. However, TOP goes one step further and allows for further parallelization within a shard. For each trans-action sent from each account, a subset of nodes within the destination shard is randomly selected via VRF-FTS algorithm to form a consensus committee. If multiple transactions from accounts within the same account subspace are sent at the same time, multiple sub-committees within the associated shard will be selected simultaneously. In some cases, the same Validator node will be selected to participate in multiple sub-committees at the same time. This is accomplished through multi-threading techniques. Each node can use multiple threads to execute the consensus algorithm in multiple sub-groups at the same time (fig 6). This extra level of parallelization is made possible due to the account chain model discussed in section 3.2.4.

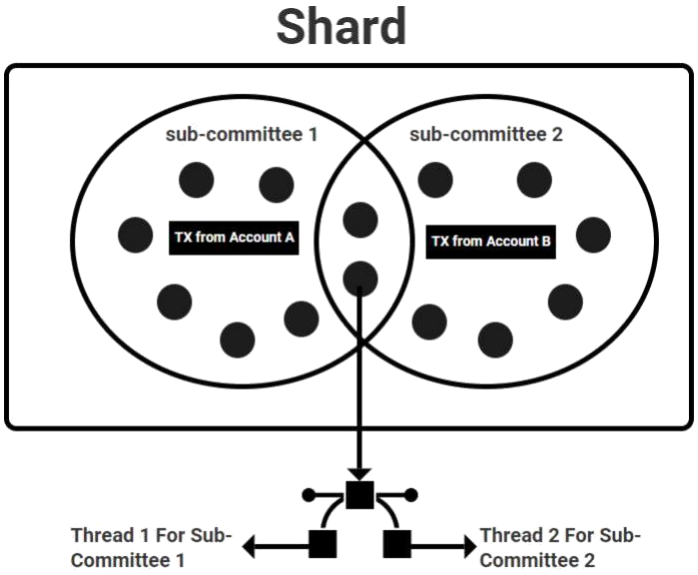


Fig. 9. Consensus Sub-Committees

4.3 Staking (PoS*) and Node Elections

pBFT algorithms on their own are not secure in a permissionless setting. Instead, these algorithms need to be paired with some form of Sybil resistance. For blockchains like Bitcoin and Ethereum, PoW is used as the Sybil resistance mechanism. However, PoW is very energy inefficient, and requires nodes to use expensive mining equipment. TOP Chain instead uses Proof-of-Stake (PoS) to prevent Sybil attacks. PoS dictates that nodes must deposit stake in the form of the native blockchain token in order to join the network. This makes it expensive for malicious entities to gain control of enough nodes to launch attacks.

4.3.1 Comprehensive Staking

In most forms of PoS, the only factor determining a node's eligibility to join the network is a minimum deposit of the blockchain's native token. TOP expands this concept into what we call Comprehensive Stake—which is represented by the the * in hpPBFT-PoS*. Comprehensive stake takes into account multiple factors to determine how likely a node is to participate in consensus. These are:

Deposit: The main component of the Comprehensive Stake is the number of TOP tokens deposited. Each types of node on TOP has differing requirements in terms of the minimum deposit to be eligible.
Credit: A node's contribution history is used as a form of credit or reputation. The more jobs successfully completed by a node, the higher its credit, and thus Comprehensive Stake. If a node's bandwidth or compute power is too low compared to the rest of the nodes, its reputation score will drop.
Voting: Any token holder can use their tokens to vote for a particular node. The more votes a node receives, the higher its chances of election. TOP allows for proxy voting via smart contract. Token holders can pledge their tokens to another individual or organization to vote on their behalf.

4.3.1.1 Node Reputation Scores

Auditors and Validators each have their own reputation scores. Since a node can serve as both an Auditor and a Validator, a node can have both an Auditor reputation score and a Validator reputation score. After a new node is registered, the reputation score defaults to 0.1.

Shards report things like the number of blocks produced by each Validator and each Auditor during a given period. The Beacon Chain will use this information to update nodes' reputation scores. Validator and Auditor reputation scores increase with each new block produced up to a maximum score of 1. When the reputation score reaches the maximum score of 1 point, it will not increase any further. However, if a node's bandwidth or compute power falls below that of other nodes, or if a node performs some malicious act, its reputation score will decrease down to the lowest mark of .1. All these parameters are configurable via on-chain governance.

4.3.2 Node Elections

The TOP node election system was developed with the ideals of permissionless-ness and decentral-ization in mind. Any node has a chance at becoming an active node given they meet some minimum requirements. For nodes registering to participate in consensus, the chances of being selected as an active node are proportional to a node's stake.

There are several types of nodes on TOP Network. Each node has minimum staking requirement which must be met in order to be eligible for active selection. Registration, sorting and election are handled through several smart-contracts deployed on the Beacon Chain, along with a VRF-FTS algorithm.

Processes of TOP Node Election:
-The node account calls the system contract to complete the payment of node deposit and the registration of node information (including the node type).
-When the node process is started, the network entry request is triggered, and when the request is agreed upon, the node is understood to be placed in an alternative pool of Root network.
- According to the node type selected during registration, the smart contract of the main chain election will be triggered periodically, and the nodes will be randomly selected from the alternative pool of Root network to be recommended to the alternative pool of the main chain or the service side-chain.
- The main chain will periodically elect the nodes of the main chain alternative pool to be the active nodes of the main chain.

4.3.2.1 Node Types and Requirements

There are 4 types of nodes on TOP Network: Validator Nodes, Archival Nodes, Edge Nodes, and Auditor Nodes. Each type of node has differing minimum eligibility requirements. Validator Nodes and Edge Nodes only require a minimum deposit of TOP tokens. Auditor Nodes and Archive Nodes require both a minimum TOP token deposit, as well votes from token holders. These minimum values can be adjusted through on-chain governance protocols.

It is also important to note that each miner account can fill multiple node roles at the same time. For instance, a miner could simultaneously act as an Audit Node, Validator Node, and Archive Node. In the case of multiple roles, the miner must meet the requirements of the node type with the highest minimums.

Edge Nodes

Edge Nodes act as the access points for clients. All transactions are first sent to the Edge Network before being relayed to the Routing and Core Networks. This protect the consensus nodes from DDOS and other similar attacks. To register as an Edge Node, a miner must deposit a small minimum number of TOP tokens, tentatively set to 100,000 TOP.

Validator Nodes

Validator Nodes make up the Shards of TOP Chain. Validator Nodes are responsible for verifying trans-actions via TOP's hpPBFT consensus mechanism. To register as a Validator Node, a miner must make a minimum deposit of 500,000 TOP tokens to have the chance at becoming an active node. However, a Validator Node's chances to become active and be selected as a pBFT leader are proportional to the node's Comprehensive Stake.

Audit Nodes

Audit Nodes have several duties in the TOP Chain system. These include: cross-shard synchronization, cross-shard transaction routing, and participating as secondary auditors in TOP's hpPBFT consensus mechanism.
To register as an Auditor, a minimum deposit of 1 million TOP is required. In addition, the node must garner at least as many votes from token holders, who vote for nodes they support by locking TOP tokens. Note that if an account is only registered as a Validator, it will not be able to receive votes even if its security deposit later reaches 1 million TOP.

When the number of votes an Auditor has received is less than the amount of TOP tokens deposited, the status of the Auditor is inactive. When the number of votes is the amount of TOP tokens deposited, the roles of the Auditor can be activated. At this point, the node can be upgraded to the status of "Advanced Node."

Archival Nodes

Archive Nodes store the entire state of TOP Chain. This is their primary responsibility. Among other things, Archive nodes help new nodes sync to the current blockchain state and ensure data availability as nodes are shuffled between Shards and Clusters. To register as an Archive Node, a miner must make a minimum deposit of TOP tokens and receive votes from tokens holders.

5 Service-Layer

While a scalable infrastructure is necessary for a DApp platform, it is not enough. Blockchain development is complex, and there are very few blockchain developers on the market. To make a bustling ecosystem, developers need the necessary resources to easily build secure and scalable DApps. TOP provides an array of tooling and frameworks for DApp development, which allows even non blockchain developers to quickly build on the platform.

5.1 Smart Contracts

To ensure both security and flexibility, TOP Network provides two types of smart contracts: platform smart contracts and application level smart contracts.

5.1.1 Platform Smart Contracts

Platform smart contacts are deployed on the main-chain, and are used for scenarios involving asset transfer, elections, and on-chain governance. These contracts are written in a stack-based non-Turing complete scripting language to ensure security and enforceability. Although platform contracts are not Turing complete, they can call built-in Turing complete core functions executed by the Core VM which provides flexibility.

5.1.2 Application Level Smart Contracts

Application level smart contacts are deployed on service chains and allow for the execution of complex business logic. Each service chain can utilize a different execution environment. As a result, application developers will ultimately be able to choose which language suites their needs. As of now, only the Lua language is supported, although in the future we will be adding support for Web Assembly-based frameworks that support many more programming languages, including Rust, JavaScript, and more.

5.2 Industry Specific Development Frameworks

The service layer also includes development frameworks for various industries. This is similar to traditional App development, where application developers often use frameworks which abstract most of the lower stack levels. By providing easy to use frameworks, the pool of developers who are able to build on TOP expands greatly.

5.2.1 Decentralized Cloud Communications

As an area of expertise, TOP provides robust development frameworks for various cloud communications functions, including VPN, Messaging, CDN, VoIP, Video, and Streaming. The underlying communications stack, from deploying networking infrastructure to encoding/decoding voice and video data, is handled for the developer. Using our SDK/APIs, DApp developers can focus on building the interface and choosing which functionalities to provide instead of worrying about the underlying communications stacks.

TOP Network offers an array of network communication and telecommunication services, which can support any functions that a communication application requires.

5.2.1.1 TOP RCS Service

TOP RCS (Rich Communication Services) offers the following services:

-One-to-one Chat
-Group Chat
-Internet Call
-Video Call
-Walkie Talkie
-Content Sharing
-File Transfer
-Presence
-Others

Developers can easily utilize TOP RCS to build powerful messaging apps similar to WeChat or WhatsApp. Since TOP RCS is based on the international standard protocol, all communication apps using TOP RCS are interoperable. On the contrary, WeChat, WhatsApp and Telegram cannot interact with each other as they are confined to their own private, incompatible protocols.

TOP RCS implements end-to-end encryption with the Double Ratchet algorithm. TOP uses the X3DH algorithm to safely manage and exchange secret keys and encrypt data with the AES 256 algorithm. Messaging apps supported by TOP RCS are highly secure and able to protect user privacy. No centralized third party can censor content or block users on TOP Network.

TOP RCS Service adopts a Proof-of-Delivery consensus algorithm to verify the workload of nodes that provide messaging services and uses Proof-of-Calltime consensus mechanism regarding the calling ser-vice.

5.2.1.2 TOP Proxy Service

TOP Proxy Service provides a proxy overlay network for users to connect to the Internet, which is also known as Virtual Private Network (VPN) service. The proxy overlay network on TOP Network consists of a proxy

server group and a relay server group. An individual server can act as both a proxy and a relay server. Below is the process of users accessing the Internet with TOP Proxy Service:

1. A user requests VPN service from TOP.
2. TOP Proxy Service randomly selects three relay servers and one proxy server.
3. The user connects to one of the relay servers, which is further connected to the proxy server through the other two relay servers.
4. The proxy server is connected to the Internet.

TOP Proxy Service protects users' online security and anonymity and offers strong anti-blocking capability by adopting a series of techniques listed below.

1. The selection of three relay servers and one proxy server for each session is random.
2. Each server knows only the next server it is connected to instead of the entire transmission path.
3. Data is encrypted by each server and decrypted only by the next server.
4. Data is obfuscated by the client SDK and each relay server so that attackers cannot identify traffic patterns.
5. Traffic is split and passed on to the two intermediary relay servers so that neither of them has access to the entire traffic.

TOP Proxy Service utilizes Proof-of-Bandwidth (PoB) consensus mechanism to verify the workload of each relay server and proxy server.

5.2.1.3 TOP Streaming Service

TOP Steaming Service offers streaming media service and live streaming service. A P2P content distribution network (CDN) is built within TOP Network, via which TOP Network delivers videos to the audience. The CDN service developed by TOP costs much less compared with conventional CDN ser-vices, reducing bandwidth expenses significantly for developers.

TOP Streaming Service supports standard HLS and HTTPS protocols. It is very easy and simple for apps to access TOP Streaming Service.

TOP Streaming Service adopts Poof-of-Bandwidth (PoB) consensus mechanism to verify the workload of each streaming server.

5.2.1.4 TOP IoT Service

TOP IoT Service provides standard communication protocols for IoT devices to interconnect and interact with each other. TOP IoT also offers SDK for IoT devices to convert their private protocols to standard protocols. TOP Network provides a secure public infrastructure network for a massive number of IoT devices to exchange and store data.

TOP, an ecosystem of IoT, supports a variety of IoT device-related services including registration, discovery, remote call, settlement and payment.

5.2.1.5 TOP VoIP Service

TOP VoIP Service provides PSTN telephony service, which enables users to make calls over the Inter-net to cell phones and landline phones. For VoIP calls, voice signals are carried on the Internet before reaching the PSTN network, allowing VoIP call providers to offer lower rates compared to traditional phone calls. The rate gap between VoIP calls and traditional calls is especially huge for International calls.

TOP Network is an open communication network and it is entirely permissionless. Any voice termination providers from small voice aggregators to large carriers can join TOP Network to become a voice service provider. Without middlemen taking a share of on the service fees, TOP VoIP Service offers much lower price and better voice quality.

Apps connect to TOP VoIP Service with TOP SDK, which encapsulates Audio Encoding/Decoding, QoS, protocol implementation, etc.

TOP VoIP Service adopts Proof-of-Calltime consensus mechanism to verify the workload of VoIP ser-vice providers.

The decentralized communications infrastructure consists of an array of service node providers. These service providers could be anything from company data centers, to individuals contributing bandwidth as a VPN relay node. DApp developers can employ service nodes through a decentralized smart contract-based marketplace, where service fees are paid in TOP tokens.

5.2.2 Blockchain Gaming

TOP also provides development frameworks for blockchain games. Many blockchain games have com-mon development workflows. For instance, a smart-contract which issues some sort of in-game asset. These common development components are provided pre-built by TOP for developers to utilize. In this way, blockchain game developers can use optimized and secure components out of the box, greatly reducing development time and potential bugs.

6 Application-Layer

The Infrastructure and Service layers act as the building blocks for the Application Layer. This top level consists of a diverse set of DApps fueled by a large userbase of over 80 million users (at time of writing). The applications in the ecosystem can be broken down into three categories: utility, in-house, and third-party developer DApps.

6.1 Utility DApps

Utility DApps are services which facilitate the operation and growth of the ecosystem. These could be cryptocurrency wallets, decentralized exchanges etc. TOP Network will provide some Utility DApps, although these types of DApps will not necessarily be exclusively developed by TOP. As an example, we have already developed HiWallet, a cryptocurrency storage and exchange service which will also act as a common entry point for other DApps in the ecosystem.

6.2 In-House DApps

Currently, there are far too many blockchain projects compared to the number of DApp developers. Most projects have taken the strategy of heavily marketing and vying for the attention of a small pool of DApp developers. We take a different approach. In the the early stages, TOP Network will be a closed-loop ecosystem. We have formed several in-house development teams to build the first large-scale DApps on TOP Network. Additionally, we will be porting our pre-existing traditional communications applications onto TOP Network which already have millions of DAUs. This will help jumpstart the ecosystem, which will make it easier for third-party DApp developers to launch their DApps in the future.

6.3 Third-Party DApps

Once the ecosystem is more mature, and we have released our own DApps along with porting millions of users onto the platform, we will begin the process of actively seeking out and inviting third-party applications developers to build their applications on TOP Network. With the provided development frameworks, developers will be able to quickly and effortlessly build all kinds of DApps, from gaming and communications, to social networking or DeFi.

7 Conclusion

While blockchain technology has come a long way in the past decade, it still needs vast improvements before it can truly support real world applications. TOP's mission is to fill in the holes and remove the remaining bottlenecks holding back DApp development. By building a comprehensive full-stack blockchain platform, we hope to finally bring blockchain to the masses, and accelerate the development and adoption of decentralized applications.