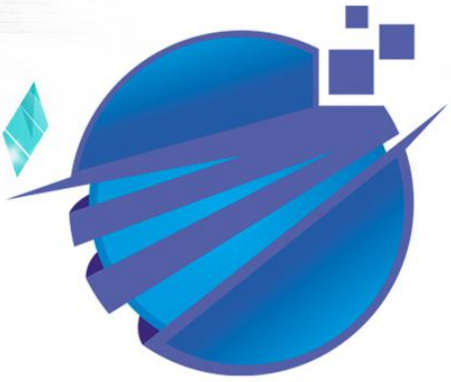


# Paloalto

## Next Generation

## Distributed Computing

## Network Operating System

0.1.0.Version

# Catalog

# Introduction

It is now nearly ten years since the beginning of the block chain with the article of Bitcoin of Satoshi Nakamoto in 2009. Bitcoin, as the "creation currency (CSC)", has already assumed the role of anchoring currency in the encryption currency system. However, in a strict sense, it is difficult to attribute Bitcoin to the underlying public chain.

"De-centralized thinking" of Bitcoin is actually very simple in terms of the underlying code. It consists of only 256 instructions, and these scripts have great fairness and stability. However, because of this, Bitcoin has not realized Turing's completeness and can only support the transmission of simple data, and there are still some problems such as slow speed and high handling fees.

Compared with Bitcoin, the biggest improvement of Ethereum is to support "Turing Complete", which has become a programmable block chain network system.

Ethereum has built a relatively complete set of underlying systems just like the Windows system of computers. Developers can build their own applications on Ethereum.

As a result of this series of improvements, the community calls Bitcoin Block Chain 1.0, which mainly implements security and fairness mechanisms. Ethereum is called Block Chain 2.0, which provides a wider application space for the block chain industry as a platform for intelligent contracts and decentralized applications.

But even so, there are still some problems with Ethereum especially in terms of efficiency. If you transfer money in Ethereum, it will take 15 seconds to dozens of minutes to complete, and if you encounter

Ethereum network congestion, it will even take more than ten hours. A large number of DAPP operations have caused serious congestion in the Ethereum network.

Due to a large number of transaction requests, the transaction cost of ETH (including tokens) rose sharply every day (47 times as much as six months ago in early 2018).

Certainly, the current block chain technology is still at an early stage of development, and existing block chain systems all have one or more of the following problems. It is difficult to try new ideas. The Paloalto project aims to provide an extensible and flexible block chain that supports digital asset transactions, data access, and process control through a hierarchical structure. It creates a framework to allow users to execute smart contracts in an efficient manner. It also provides a developed architecture that uses the underlying infrastructure to quickly and easily create a block chain. It is an efficient and easy-to-use distributed operating system and can provide necessary components for the erection of sub-block chains. At the same time, it provides solutions for the testing of new ideas, the deployment of private chains, the processing of complex tasks and the application of intelligent contracts.

In terms of basic chain technology, Paloalto team has made great innovations in system capacity, horizontal expansion capability, consensus algorithm, transaction speed and security optimization of block chain technology, and has developed and deployed the next generation distributed computing network operating system, which we call Paloalto. The application of Paloalto in the landing industry is not a one-on-one effort, but is integrated with other key technologies such as artificial intelligence, block chain, cloud computing, big data, biometrics and so on, forming the core

capability of value exchange network of Paloalto. Paloalto is a technical transition from design concept and coding implementation, which promotes the "block chain" to be commercialized and enter a new stage of value internet society.

# 1. Current problem of block chain

The block chain technology is still in the stage of continuous improvement. For the Bitcoin of its first application, there are three major problems that are being perfected or attempted to be perfected by other systems.

## 1.1 Block capacity and transaction speed limits

When designing bitcoin, Satoshi Nakamoto set 1MB capacity limit for each block so that each block can only hold 4096 transactions. At the same time, the workload verification mechanism makes it take about 10 minutes to confirm the transaction and record the transaction in the block chain. When the amount of calculation reaches the limit, the calculation time will slow down. At present, Bitcoin network is not mature enough to expand its scale to the main credit card network, and work is underway to raise this limit.

In addition, block expansion has become an urgent need. A Bitcoin core developer proposed a switch from BitcoinCore to BitcoinXT, with the block expanding from 1MB to 8MB and doubling every 2 years thereafter. This change requires 750 out of 1000 consecutive blocks to include miners' change approval. At present, this event is developing.

## 1.2 Mining wastes huge resources

There was a system of engineers who calculated that the mining cost of the entire network was about 800 million US dollars per year, including about 77.712 million US dollars for electricity and 733 million US dollars for mining machines, with Bitcoin's computing

capacity of 110 million GH/S.

In terms of electricity costs, 80,666 kilowatts of electricity are required per day, which is calculated at US $ 100 per megawatt, and the result is about US $ 70.712 million per year. In addition, these mining machines produce 424,725 tons of carbon dioxide per year.

Since the mining work is only to search for random numbers to obtain valid hash values and does not generate other values, Bitcoin's computational resources and power costs have been criticized as a waste of resources. The general idea of other systems to improve this problem is to reduce the number of nodes involved in maintenance and the intensity of mining competition. There are two specific ways, one is to use the private chain or alliance chain to impose "bookkeeping rights" on certain nodes; The second is to introduce ProofofStake mechanism to maintain a reliable database in conjunction with workload certification.

The ProofofStake is a proof of currency ownership. The certifier needs to provide a certain amount of currency ownership. The system determines the "bookkeeping right" according to the proportion of currency occupied by each node and the occupation time; The core of ProofofStake is to allow only those who have economic interests in the block chain to participate in the maintenance of the system, which makes the cost of mining far lower than that under the workload proof mechanism.

## 1.3 Lack of Turing completeness

Since the block chain can guarantee that Bitcoin transaction records will not be deleted, in theory it can also guarantee that any code once written cannot be deleted. However, Bitcoin's script language is

not complete in Turing, that is, it does not support circular statements, meaning Bitcoin can only be used as a digital cash and cannot directly support smart contracts and more complex decentralized applications.

The block chain technology platform script language of Ethereum (EthereumVirtualMachinecode) is Turing completeness; Using EVM code to build applications can theoretically realize any conceivable calculation including infinite loops. Ethereum enables anyone to upload and execute any application program, and the effective execution of the program can be guaranteed, but the block is bloated and unacceptable.

# 2. Key features of Paloalto

## 2.1 DPOS-Pbft consensus

Alto utilizes the only known distributed consistency algorithm, which can meet the performance requirements of applications on block chain and DPOS. Under this algorithm, those who hold tokens on the chain of blocks using Alto can select block producers through the continuous approval voting system. Anyone can choose to participate in block production and have the opportunity to make blocks as long as they can persuade token holders to vote.

Alto can generate blocks exactly every 0.5 seconds, and only one producer can generate blocks at any given point in time. If no block is generated at a predetermined time, the block of the time slot is skipped. When skipping one or more blocks, there is a second gap of 0.5 or more in the block chain.

Using Alto, block producers are chosen according to the preference of token holders. Selected producers are arranged in the order agreed by multiple or more producers.

If producers miss a block and have not produced any blocks in the past 24 hours, they will be excluded from consideration until they inform the block chain that they intend to start producing blocks again. This can ensure smooth operation of the network and minimize the number of missed blocks due to not arranging proven unreliable producers.

Under normal circumstances, the DPOS block chain will not encounter any bifurcation because block producers cooperate to produce blocks rather than compete. If there is a bifurcation,

consensus will automatically switch to the longest chain. This method is effective because the rate at which blocks are added to a block chain branch is directly related to the percentage of block generators sharing the same consensus. In other words, the length of the branch of the block chain with more producers will grow faster than the branch with fewer producers because the branch with more producers will experience fewer legacy blocks.

In addition, no block producer should produce blocks on both bifurcations at the same time. A block producer caught doing so may be voted. This double-produced password evidence can also be used to automatically delete the abuser.

By allowing all producers to sign all blocks, Byzantine fault tolerance is added to the traditional DPOS as long as no producers sign two blocks with the same timestamp or the same block height. Once multiple manufacturers have signed a block, the block is considered irreversible. Any Byzantine producer must generate encrypted evidence of their treason by signing two blocks with the same timestamp or block height. In this mode, the irreversible consensus should be reached within one second.

### 2.1.1 Certificate of entrusted equity DPOS

Certificate of entrusted equity DPOS is the variety of POS by using the typical DPOS such as Bitshares. The basic principle of DPOS is to vote across the network to select a number of nodes to act as bookkeeping authority. The authority of these representative nodes is completely consistent. On behalf of the node, bookkeeping takes place in turn. You can choose whether to create blocks or not. However, they cannot change the details of the transaction, and

malicious or late representative nodes' behavior will also be made public. Therefore, the network may simply and quickly vote them out. The expelled representative nodes will lose their bookkeeping authority and corresponding income.

As a consensus mechanism of weak centralization, DPOS retains some key advantages of centralized system such as transaction speed. However, each coin holder has the ability to decide which nodes can be trusted, and in fact, the representative node will voluntarily reduce its own income to win more votes, and the remaining income will be paid as dividends to all bit share holders. DPOS is somewhat similar to representative democracy and the board of directors system of joint-stock companies. It is an elite system, but its identity is subject to the following people. In DPOS, the holder of the currency has at least the right to decide the identity of the representative node, or miner.

### 2.1.2 Practical Byzantine Fault Tolerance (Pbft)

In 1982, LeslieLamport and other scientists raised the famous Byzantine Failures, which discussed the issue of consistency in the scenario of allowing a few nodes to commit crimes——The Byzantine Empire, with its vast territory, deployed each army across the country for defense purposes. Because the distance is far away, the generals can only rely on messenger to deliver the message. When the war broke out, the generals of the Byzantine imperial army must decide unanimously whether to attack an enemy in order to win the battle. However, the generals were not sure whether there were traitors among them. Traitors may change their attack intention or attack time without authorization in order to destroy the

consistency of the attack and cause the battle to fail. In this state, what kind of long-distance communication method should Byzantine generals agree to achieve unanimity?

The article points out that for Byzantine problems, if the total number of nodes is N and the number of mutinous generals is F, then when N≥3F+1, the problem will be solved and guaranteed by Practical Byzantine Fault Tolerance BFT. Briefly describe the demonstration process: assume that the total number of nodes is N, the total number of evil nodes is F, the number of valid good nodes is L1, and the number of invalid (failed) good nodes is L2; If the system wants to reach an agreement safely, it must satisfy two points: the valid good node must surpass the evil node, and at the same time it must surpass the failed good node. When it is converted into a mathematical formula——L1≥F+1， L1≥L2+1（L2=F），and L1=N-L2-F, i.e. N-F-F≥F+1, resulting in N≥3F+1; Therefore, when the mutineers do not exceed 1/3, there is an effective Practical Byzantine Fault Tolerance. However, BFT has always had the problem of too high complexity and has not really landed in the actual scene.

MiguelCastro and BarbaraLiskov put forward the practical Practical Byzantine Fault Tolerance (PBFT) in 1999 to solve the problem of low efficiency of the original Byzantine fault tolerance algorithm and greatly reduce the complexity of the algorithm, making Byzantine fault tolerance algorithm feasible in practical system applications. PBFT is an algorithm developed for distributed system execution environments where state machine replicas are the main ones. It aims to allow most honest nodes in the system to cover the behavior of malicious or invalid nodes. The operation steps are as

follows:

(1) Randomly take one copy as the master node and other copies as the backup.

(2) The client sends a request to the master node to use the service operation;

(3) The master node sends the request to other replicas through broadcast;

(4) All copies execute the request and send the result back to the client;

(5) The client needs to wait for F+1 different replica nodes to send back the same result as the final result of the entire operation. The PBFT algorithm requires that the number of failed nodes in the system should not exceed one-third of the total network nodes, and the fault tolerance rate is relatively low.

## 2.2 Double layer fragmentation

Fragmentation is a concept widely used in databases, and fragmentation is used to improve the efficiency of databases. One fragmentation is the horizontal part of the database, and each fragmentation is stored in a responsible server, thus the load is spread out to make the database more effective.

After fragmentation in the block chain, each node has only a part of the data in the block chain, not the entire information. Therefore, in fragmentation, the feature of de-centralization is still maintained. Each node does not load information on the entire block chain, thus achieving scalability.

Fragmentation technology increases the throughput of the network by changing the way the network authenticates blocks.

In the public chain, transactions are classified into different fragmentation and processed at the same time. A large number of fragmentations are obtained from the whole network. Each node processes only a small portion of the transactions in the entire network, and the process is simultaneous with other nodes in the entire network. Therefore, the more nodes in the network and the more fragmentation, the more transactions the entire network can handle at the same time. This property of fragmentation is called scale-out.

In any block chain network with high TPS processing capacity, a super node that keeps a whole network account book will be very expensive, which will lead to centralization. In order to avoid this problem, Paloalto supports a cluster of cheap nodes to realize the function of a super node and avoids centralization.

## 2.3 Cross chain

The existing block chain technology has bottlenecks in performance, capacity, privacy, isolation and expansion under a single chain architecture.

Imagine a Visa-like payment application with hundreds of millions of users, with tens of thousands of transaction requests per second, hundreds of millions of transactions per day, and a second response experience for user transactions. Under the existing block chain technology, the use of chained local storage in data storage leads to inability to expand in parallel, the adoption of synchronous state machine model in consensus mechanism leads to inability to process transactions efficiently, and is limited by the performance limit of a single node in the network. Therefore, the single chain architecture

cannot meet the performance, capacity, user experience and other requirements of applications.

In terms of storage capacity, because every node in the single chain in the current block chain technology system has all the data in the whole network, it cannot meet the requirements of high-capacity storage.

At the same time, the interoperability of block chain itself is the basic requirement of some applications. Imagine a wealth management application where users can exchange a certain asset for wealth management products of different institutions. Different assets need to be transferred and exchanged in multiple chains. There are other DAPP that also require cross chain data interaction between multiple chains such as exchange rate quotations, weather, stock prices, specific indicators, etc.

### 2.3.1 Paloalto intersect cross chain

Intersect cross chain of Paloalto mechanism is similar to Sharding mechanism of Ethernet. It can realize the scale-out expansion of storage capacity, processing capacity, fault tolerance mechanism and function combination through multiple blocks of the same type.

Paloalto expands its capacity by means of isomorphizing hierarchical side chains. It provides a multi-chain class routing mechanism by setting up. Through the communication protocol and routing protocol between the multi-chains, Alto will serve as a multi-chain router to maintain the network topology map between the multi-chains. The goal is to solve the problem of connection and distribution between multiple chains. The chain consists of a hierarchical network structure that can be combined at multiple levels.

# 3 Unlimited scalability

Paloalto expands the network "almost indefinitely" by dividing consensus, validation, and storage into different levels of architecture. There are no trading blocks in the consensus layer. The verification layer is responsible for combining the hashes of all fragments into a structure similar to the Merkle Trees so that the global state hashes are stored in the blocks of the top-level chain. The storage layer is divided into multiple chains, each chain is responsible for processing transactions that update the fragmentation status. Although this architecture provides a solid theoretical basis for expansion, there are still some major problems to be solved in practice such as transactions affecting multiple fragmentation states.

Ethernet's expansion plan also includes state fragmentation, but it may take several years to deploy such a working system on the main network. Ethernet will have to move from a fully replicated global state to a fragmented state, while Paloalto implemented fragmentation from the beginning. Paloalto will be able to successfully implement state fragmentation for the first time, which gives it an important expansion advantage.

Paloalto's direct deployment consists of a two-layer block chain structure, with the first layer being a fragmentation layer (which can be understood as a sub-chain layer) for transaction bookkeeping. The second layer is a chain used to confirm transactions in fragments. Without affecting the root chain, the number of fragmentations in the fragmentation layer can be dynamically increased, thereby improving the overall throughput of the system and solving the

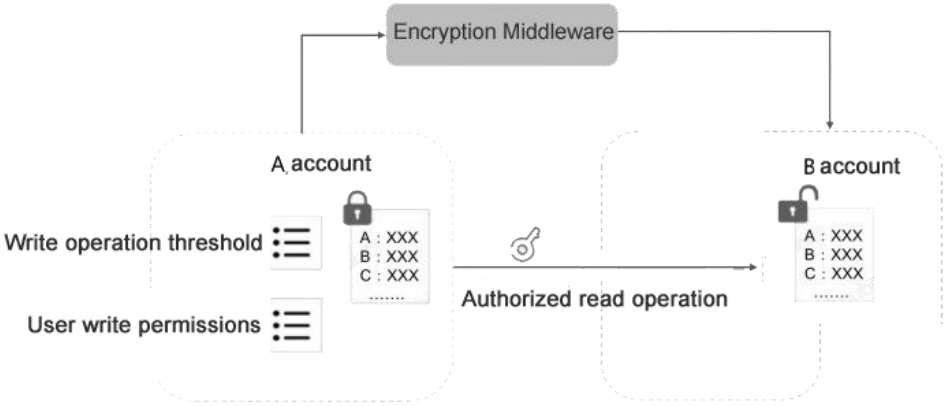
problem that the transfer speed of Tps in the basic system is as high as 10000000/1s per second

## 3.1 Block chain architecture for multi-service fulfillment

Paloalto's block chain structure can meet the capacity and maintenance efficiency of different business areas. It can be used to mark assets and asset transfers, record them, and trace their origin to track the circulation of items.

## 3.2 Rights control policy

Provides two types of permission control strategies for data information writing and reading. Data information write permission, multiple users are set under the same account, and corresponding permissions are set for different operations to meet the use scenario of multi-party signature control. Data information reading permission allows users to grant and revoke the operation permission of a single user or user group on data, and user groups can be flexibly configured by users. The data includes user account information, transaction information, etc. The granularity can be refined to each attribute field of the transaction or account.



# 4 Turing complete state channel

## 4.1 Smart contracts

Turing machine model describes a computing program that can operate like a physical computer and accomplish everything a computer can do. The so-called Turing's complete programming language is a programming language that can realize all the functions that a computer can realize.

Although only the status update of alto transfer occurs in the chain, Paloalto still provides a Turing complete virtual machine that can run "smart contracts". The contract on Paloalto must be an agreement to allocate funds according to certain rules, which is completely contrary to the contract of similar entities in Ethernet. There are two more obvious practical differences: according to the default rules, only relevant parties know a certain contract, and only relevant parties with open channels can create a valid contract. If the parties agree on a contract, they can sign the contract and then back it up for future reference. Only when the result is disputed will the contract be submitted to the chain, at which time the code is stored only as part of the submitted transaction, not in any other case. If this is the case, the block chain will allocate tokens according to the contract and close the channel.

### 4.1.1 Contract interaction

Even if all contracts are stateless and execute independently of each other, contract interaction and statefulness can still be achieved through hash locks.

```
: hashlock
swap
hash
== ;
```

In line 1, we define a function called "hash lock" and expect the stack to contain hash h and secret s. Line 2 interchanges the two to hash the secrets of line 3 before calling the hash (v) and h equality operators on line 4. Returns true if the secret is a pre-mapping of a hash. This function can be used to predict execution of code branches in different contracts based on the presence or absence of the same secret value.

```
macro Commitment a9d7e8023f80ac8928334 ;

Commitment hashlock call
if 0 100 else 0 50 end
1
```

As a simple practical case, hash locks enable users who do not share state channels to send Alto to each other in a trusted manner as long as there is a channel path between them.

They do this by creating contractual backups, one for each channel. The commit on line 1 selects a secret hash. On line 3, we push it into the stack and call the hash lock function. Which branch of the IF statement will be executed depends on the return value of the hash lock. Once these contracts have been signed by all parties concerned, the secret will be released so that it can obtain their respective Alto.

## 4.2 Status channel

The status channel provides the basis for the availability of Dapp, reduces the delay of Dapp and controls the network response time

within the user's tolerable range. Dapp participants send messages and transactions to each other to update the status, but do not publish messages into the chain. If one of the participants leaves or attempts to deceive another participant, the latest transaction can be posted to the block chain at any time to complete the status. The rewards and punishment measures are enough to keep the participants honest.

The status channel is the interaction between the two parties and can be applied to any smart contract. The status channel manages the status of business processes or transactions. It can reduce transaction costs while ensuring interaction performance and privacy among designated people.

Fragmentation technology can achieve scalability to a certain extent, but for applications that rely on a large number of atomic operations (such as streaming payments, Internet of Things devices, games, etc.), fragmentation technology cannot effectively reduce costs, while for a large number of finely divided transactions, spending can be greatly reduced through state channels.

### 4.2.1 Cost of margin locking

At present, the biggest challenge for state channel projects is the need to lock in a large amount of margin. For example, if the average transaction amount is 1 Bitcoin and there are 10,000 nodes on the network, then each node must deposit at least one Bitcoin as a security deposit, and a total of 10,000 Bitcoins will be required. The opportunity cost of such a large margin is very high, so the state channel cannot realize zero transaction cost in the economic model despite realizing zero handling fees in technology.

### 4.2.2 State channel balance

In addition, the state channel balance is also a great technical challenge. For example, when the total amount of transactions from A to B and from B to A are equal between the two nodes A and B, A and B can reach a balanced state, and no amount of one of them will gradually become zero. Once the amount of one party of a channel becomes zero, the channel will become one-way and the reverse transaction cannot continue, thus affecting the connectivity of the network.

### 4.2.3 Status maintenance of node dropping

The maintenance of the status channel requires that the nodes must be online. If a node is attacked or goes offline by itself, the original status will be lost. For example, in the course of the game, the backward party may choose to go offline on its own or take the leading DDoS offline. Therefore, the state channel needs an additional mechanism to ensure that the node can maintain its original state when it goes offline.

## 4.3 Alto embedded predictor DAPP

Whether written as text or code, the most critical function of most contracts is to derive values based on the environment, such as the price of different goods or whether an event occurs. An intelligent contract system lacking this capability is essentially a closed system and can be seen as useless. This is a generally accepted fact, and several projects have attempted to bring external data into the block chain in a decentralized manner. However, to decide whether a given fact is the truth or not, it essentially requires a new consensus

mechanism to be implemented in addition to the consensus mechanism.

It is as expensive to have the two consensus mechanisms run on top of each other as it is to have the two consensus mechanisms run independently. In addition, this does not enhance security, because the less secure one will still be attacked and will generate an "error" value. Therefore, we propose to combine the two consensus mechanisms and essentially reuse the consensus mechanism that we use to determine the state of the system, that is, also use this mechanism to determine the state of the outside world.

All this works as follows: Any Alto holder can start a prophecy machine by submitting an answer to a [Yes/No] question. At this point, they also need to make clear the timetable for this question to be answered, either immediately or at some point in the future. The user who starts the prophecy machine needs to deposit an equal percentage of Alto according to the length of the schedule. As long as the user provides an answer accepted as the truth, these tokens will be returned or destroyed. The block chain will generate a unique identifier for the oracle, which can also be used to retrieve the answer once it is ready.

When the answer to the question is provided, the user who starts the prophecy machine can provide an answer for free. Once the initiator of the prophecy machine has provided the answer, or has exceeded a certain time, any user can submit the opposite answer, of course they all need to deposit the same amount of Alto. If no conflicting answer is submitted after the end of the timetable, the answer provided by

the forecaster will be accepted as the truth and the deposit will be returned. If any contradictory answers are submitted, the consensus mechanism for the new block will be used to answer the prophecy. The latter is more expensive, but we know we can take (destroy) at least one of the two deposits. This is economically feasible.

Governance in the past, the governance of block chain system has always been a difficult problem. Whenever a system upgrade is needed, it must be hard-forked, and then all value holders will have a big discussion. If the economic incentives of users and decision makers are not consistent in a system, or there is no clear upgrade path, even the simple matter of modifying an arbitrarily set variable in the source code will be difficult to achieve, as we have seen in the Bitcoin block size debate. We have also witnessed and experienced more complex governance decisions, such as fixing a loophole in the DAO's smart contract-which requires rapid intervention by system developers.

The first problem of these systems is obvious——the decision-making process for protocol upgrades or changes is not clearly defined and lacks transparency. Paloalto's governance system is part of the consensus mechanism. It will use the forecast market in order to be as efficient and transparent as possible. In addition, the consensus mechanism itself will be defined by a series of variables that determine how the system operates, and will be slightly upgraded with each new block-from trading or inquiring about the geometry of the operation of the predictor to modifying basic parameter values such as block time.

By predicting the market for variables that define the protocol, users can learn how to improve the protocol efficiently. By predicting the market for potential hard splits, we can help the entire community reach a consensus on which version of software to use. Each user can choose which dimension he or she wishes to optimize, but a simple default strategic benchmark is to maximize the value of tokens he or she holds.

## 4.4 Anti-quantum center transverse node

DAPP can use a new data authentication system specifically a lateral data authentication system based on anti-quantization nodes. The horizontal data authentication system based on anti-quantization nodes consists of node servers, networks and verification servers, and is characterized in that the nodes are connected to each other through networks to form anti-quantization distribution clusters, and the verification servers are arranged in parallel between the node servers. The utility model solves the DAPP data access problem, solves the data timeliness inside the system and greatly improves the flexibility of data application by resisting the quantization node cluster internal nodes to conduct horizontal authentication.

## 4.5Zero—knowledge proof

Zero—Knowledge Proof originated from minimum disclosure proof. Set P as the entity that holds certain information and wishes to confirm this fact, and V as the entity that proves this fact. If a protocol proves to V that P does have some information, but V

cannot infer what this information is, we say P has achieved the minimum disclosure proof. Not only that, if V can't get any knowledge other than knowing that P can prove a certain fact, we call P zero knowledge proof, and the corresponding protocol is called zero knowledge protocol.

# 5 Design of communication protocol

According to the characteristics of network simulator and service generation system, a communication protocol based on UDP protocol is proposed. The protocol guarantees the reliable transmission between the two parties through mechanisms such as overtime retransmission and acknowledgement. The protocol designs the types, formats and communication rules of various messages in the communication process.

## 5.1 Implementation principle of communication protocol

The end-to-end transmission service provided by UDP protocol transmits data as much as possible, but it does not guarantee that the data will reach the destination successfully, and the order of arrival of the data does not necessarily coincide with its transmission order. The main function of UDP protocol is to compress data stream into datagram form and provide fast connectionless service, which makes it suitable for signaling messages with sensitive transmission delay. However, UDP protocol does not provide reliable datagram service, so the functions of message sequence, message repetition and retransmission in the data transmission process can only be completed by the upper application.

The communication protocol in this section guarantees the reliability and consistency of data transmission through timeout retransmission mechanism, acknowledgement mechanism, serial number mechanism and heartbeat mechanism.

### 5.1.1 Timeout retransmission mechanism

Both sides of the communication maintain an automatic retransmission timer. When one side of the communication wants to send a message to the other side, it will start its own timer first. If no information is received from the peer within the retransmission waiting time, this timer will be reset, and the packets in the waiting confirmation queue will be retransmitted once, and the retransmission times will be accumulated once. If no information is received from the other party within the retransmission waiting time, the timer will be reset again, retransmission operation will be performed and the retransmission times will be accumulated. If the number of retransmissions reaches the specified number of times, an error message needs to be reported.

### 5.1.2 Confirmation mechanism

Confirmation technology is the basis to ensure the reliability of data transmission. When the receiver receives a message that needs confirmation, it needs to reply a confirmation message to the sender to ensure the reliability of the transmission. Each message in the protocol has a corresponding confirmation message.

### 5.1.3 Serial number mechanism

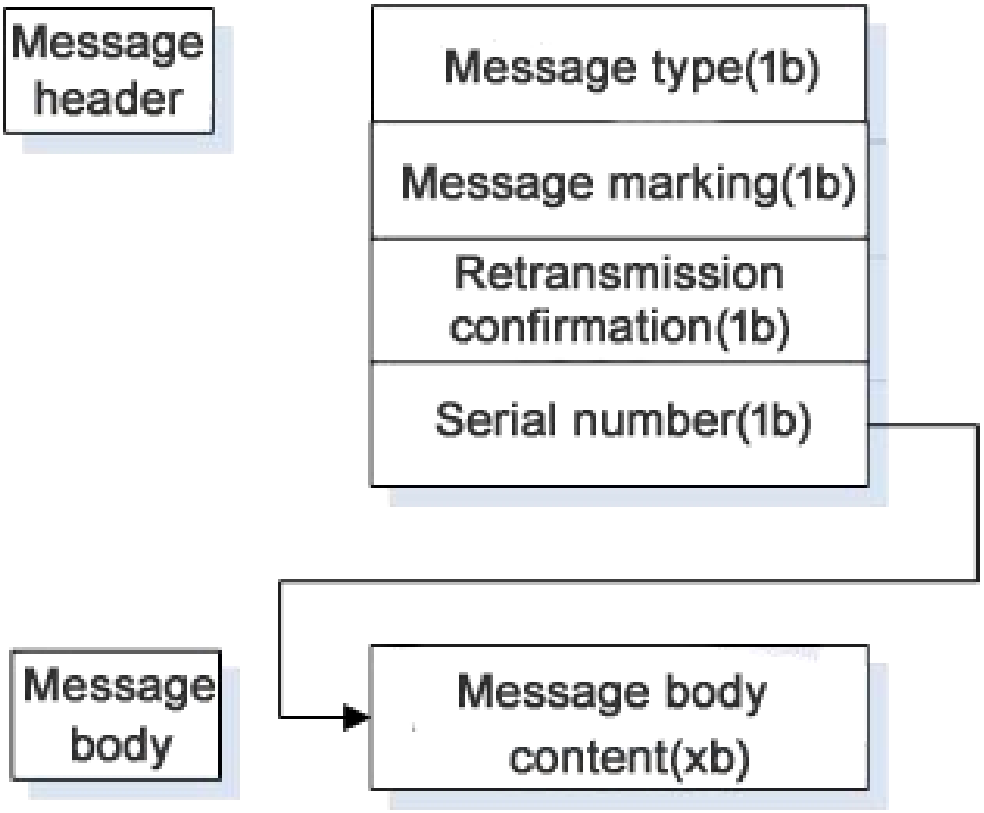
In data interaction, each message sent out is assigned a unique serial

number. The serial number is an integer value, ranging from 0 to 255. The serial number is related to the confirmation mechanism, and the serial number of each confirmation message is consistent with the serial number of the message to be confirmed. The initial sequence number was created when the connection was established. Each message sent will be added with 1 to the sequence number of the previous message. If it is a message that does not need confirmation, the sequence number of the previous message will be used. After the connection is interrupted in any way, the serial numbers of both parties need to be reset.

### 5.1.4 Heartbeat mechanism

Heartbeat mechanism can detect the survival of both parties in an unreliable network. In fact, this mechanism is a custom protocol. The two communicating parties periodically send heartbeat packet to the other party and judge whether the connection is normal by checking whether they can receive a response packet from the other party within the specified time. If one party does not receive the opposite heartbeat packet within the specified time, the other party is considered disconnected and can take corresponding measures, such as disconnecting the network connection.

The communication message between the message format design agent software and the network measurement equipment is divided into two parts: a message header and a message body, the message header is in a fixed format, and the message body is determined according to the specific message identification.




### 5.1.5 Message header definition

The message header includes four fields, which are message type, message identification, retransmission confirmation bit and sequence number.

Message type

Length is 1 byte, unsigned integer, indicating the presence or absence of message body content. A value of 0 indicates no message body; A value of 1 indicates that there is a message body.

Message identification

The length is 1 byte, unsigned integer, indicating the role of sending messages. Each message corresponds to a unique identifier, which is used to distinguish different messages.

Retransmission acknowledgement

Length is 1 byte, unsigned integer, used to indicate whether the message needs to reply to the confirmation message. When the value is 1, confirmation is required; When the value is 0, no confirmation is required; A value of 2 indicates that the message is a retransmitted message.

No.

The length is 1 byte, unsigned integer, representing the continuous feature of the message sequence, and its value is 0-255 cyclically increasing to prevent repeated sending of messages. The initial serial number is created when the connection is established. After the connection is interrupted, the serial numbers of both parties need to be cleared.

Message body definition

The length of the message body content is the number of bytes actually sent and the text type. The content of the message body needs to be determined according to the specific message identification.

## 5.2 Message identification and content design

The main messages that the proxy software interacts with the network measurement equipment are represented by enumeration in the program.

| message name | Message identificati | Message content | Message description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| | on | | |
| CONNECT | 0 | Message header+Message body | The connection request of the proxy software, the data is the port number<br><br>(no reply need to be confirmed) |
| COFIRM | 1 | Message header | Confirmation message (without reply confirmation) |
| CONN_OK | 2 | Message header | Connection success |
| DATA | 3 | Message header+Message body | The data is sent, the data is the content of the file |
| DATA_OK | 4 | Message header | Sending file content confirmation |
| DATA_OVER | 5 | Message header | End or return test results end |
| DATA_OVER_OK | 6 | Message header | Confirmation of the following or back test results |
| ERRO | 7 | Message header+Message body | Exception, data is abnormal encoding |
| FILENAME | 8 | Message header+Message body | The data is sent, the data is the filename |

| | | | |
|---|---|---|---|
| FILENAME_OK | 9 | Message header | Sending file name confirmation |
| JUMP | 10 | Message header | Heartbeat command (no reply need to be confirmed) |
| RESULT | 11 | Message header | Request to report the test results |
| START | 12 | Message header | Start testing |
| START_OK | 13 | Message header | Start the test confirmation |
| STOP | 14 | Message header | Stop testing |
| TEST_OVER | 15 | Message header | Report the end of the test to the agent software |

In the message defined above, the complete message can be formed by filling in the corresponding value according to the message format. The CONNECT message, DATA message, ERROR message and FILENAME message have message body content. CONNECT messages, JUMP messages, and CONFIRM messages do not require a reply confirmation, and all other messages require a reply confirmation. These messages will be described in detail below.

CONNECT (Request Connection Message)

When the agent software requests to connect to the network measurement device, it will send a CONNECT message with the message body containing the agent's local listening port number and a length of 2 bytes. By default, it is sent once per second. If CONN_OK message is not received after three times, it is

considered that there is a problem with the network connection.

DATA (start sending data messages)

The message body content of the DATA message is the issued test parameter file or the reported test result file.

ERROR message

The message body content of the ERROR message is an exception code with a length of 1 byte. Exception encoding is represented by enumeration type in the program.

```
public enum error
{
HeartJumpFailed,// heartbeat connection failure
ComPortsInUse,    // port is occupied
DisConnect,          // network connection disconnected
NumMissMatch,      // ordinal number lost match
ConfigError          // Network Measurement System Configuration
                                Error}
```

FILENAME message

The message body content of FILENAME message is the name of the parameter file actually issued or the name of the test result file reported.

JUMP (heartbeat command message)

After the agent software establishes a connection with the network measurement equipment, it will periodically send JUMP messages to each other, called heartbeat, which is sent once per second by default. If it does not receive the message after waiting for 10 seconds, it will consider the heartbeat of the other party to stop and disconnect the network connection.

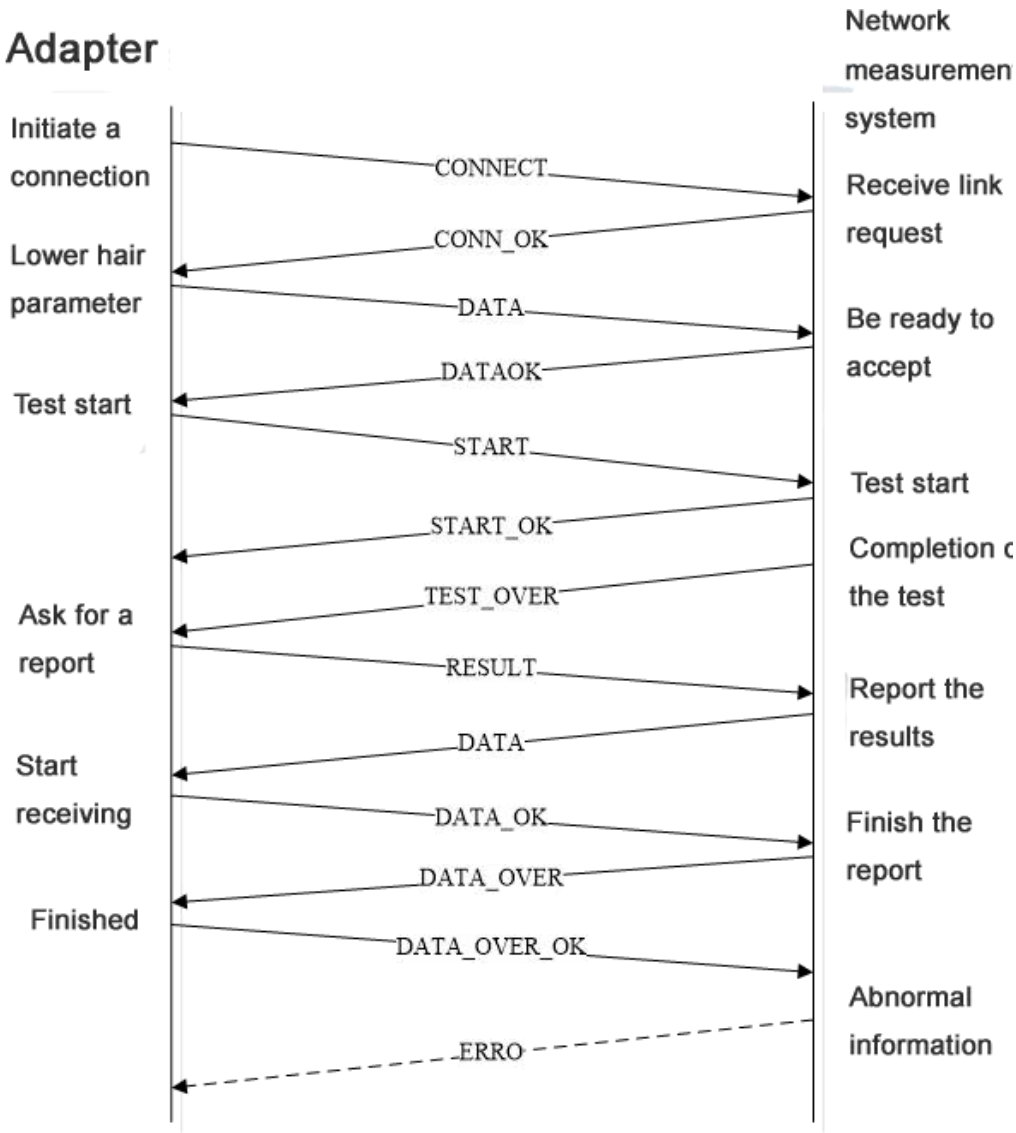
COFIRM message (CO FIRM)

The proxy software or network measurement equipment receives the

message sent by the other party. If the message type is 1, it needs to send an acknowledgement message to indicate that it has been received.

## 5.3 Communication flow between agent software and network measurement equipment

The communication flow between the agent software and the network measurement equipment mainly refers to the whole flow of the test completed by both parties.



## 5.4 Distributed economy

**PaloAlto distributed economy:**

**Total: 1 billion**

**Contract token 30%**

**Community incentive 10 %**

**Ecological maintenance 22 %**

**Cornerstone institutions 21%**

**Foundation union 10 %**

**Development group 6 %**

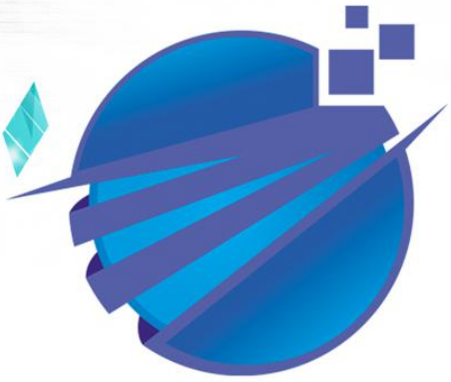
**Consultant 1 %**

# 6 Declaration

This document is a white paper on Paloalto technology, written by the Paloalto development team, and mainly introduces the technical features and application scenarios of Paloalto. In the future, we will continue to upgrade this document to make it consistent with the technical

implementation. To learn about Paloalto's latest information, technical white paper, software release, developer community, etc.

Technology is constantly developing and block chains are also constantly improving. The Paloalto development team will improve the technical plan according to the needs in the future and continuously update the technical white paper, but the circulation and distribution rules of basic tokens will remain unchanged. The Paloalto Foundation hereby informs organizations and individuals participating in digitally encrypted currency investment through any channel, and takes care to guard against risks. The Paloalto Development Group does not assume any consequences arising from participating in investment through any channel.



# Paloalto

## Next Generation

## Distributed Computing

## Network Operating System

0.1.0.Version